Lição 11

Segurança na WEB

Objetivos

Ao final desta lição, o estudante será capaz de:

- Obter segurança na comunicação entre servidor e cliente usando *Session Socket Layer*
- Aprender medidas para evitar as maiores falhas em aplicações WEB

SSL

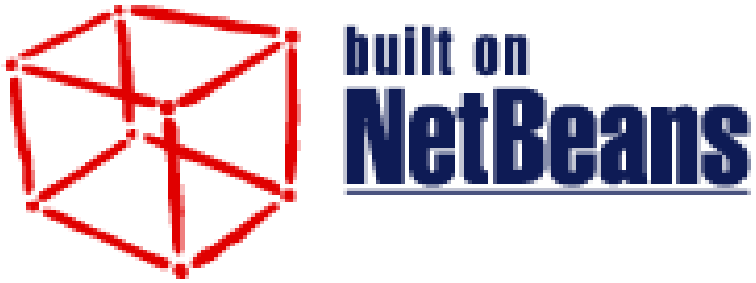
- *Secure Socket Layer*
- Camada de protocolo que se interpõem entre a camada padrão TCP/IP e a camada mais alta
- Protocolo no nível de aplicação tais como HTTP
- Permite que o servidor se autentique para o cliente e depois disso encripta o canal de comunicação
- O uso desta tecnologia garante um canal seguro entre o servidor e o cliente

Certificados

- É como um passaporte
- Identifica uma organização ou pessoa para outros e contém informações importantes sobre esses
- Normalmente é emitido por Certificados de Autoridades (CAs)
  - Um CA é como um escritório de passaportes
  - Existem diversos certificados de autoridades conhecidos, o mais popular é o *Verisign*
  - A decisão a respeito de que o CA fornecerá ao usuário com um certificado, é do administrador
- Na ausência de um certificado autenticado por um CA, um certificado temporário pode ser criado pela ferramenta incluída no Java SDK

Certificados

- Passaremos agora para o NetBeans

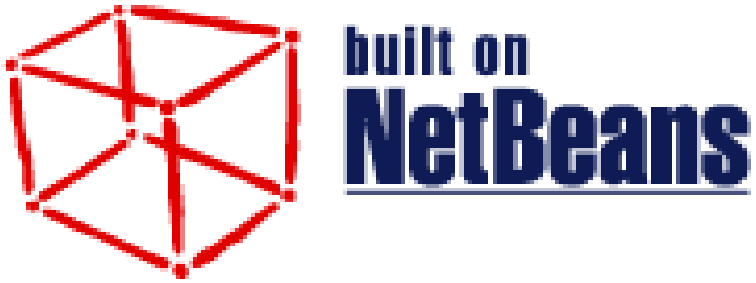


Criando um Listener HTTP Seguro

- Criar com sucesso um certificado e registrá-lo para ser utilizado
- Criar um *listener HTTP*, que possa fazer uso do certificado
- O *listener* reserva uma porta na maquina servidora para comunicações seguras

Criando um Listener HTTP Seguro

- Passaremos agora para o NetBeans



Falhas de Segurança em Aplicações WEB

- *Open Web Application Security Project* (Projeto aberto de Segurança em Aplicações Web – OWASP)
- OWASP é um projeto de fonte aberto fundada por uma entidade sem fins lucrativos, e que aponta causas de insegurança em softwares, e desenvolve medidas para encontrar essas falhas
- Liberou uma lista das maiores falhas de segurança de aplicações WEB

Entrada Inválida

- Todas aplicações WEB recebem dados de requisições HTTP feitas por usuários, e através do uso desses dados executam suas operações
- *Hackers* podem manipular uma parte dessas requisições
- Existem diferentes tipos de ataque:
  - Cross site scripting
  - Buffer overflows
  - Falhas de Injeção

Controle de Acesso Quebrado

- Há muita aplicação que categoriza seus usuários dentro de diferentes regras
- Provê diferentes níveis de iteração com diferentes níveis de conteúdo para cada categoria
  - A maioria das aplicações definem regras de usuário e regras de administração
  - Somente o administrador esta “autorizado” a acessar certas páginas ou realizar certas ações de integridade na administração do site
- O problema é que muitas aplicações não reforçam eficazmente essa autorização

Autenticação Quebrada e Gerenciamento de Sessão

- Autenticação e gerenciamento de sessão se refere a todos os aspectos de tratamento de autenticação de usuário e o gerenciamento da sessão ativa
- Há diversas áreas que podem ser negligenciadas:
  - Senha forçada
  - Uso da senha
  - Armazenamento de senhas
  - Proteção do ID da sessão

Cross Site Scripting

- Acontece quando se faz uso de outra aplicação WEB para enviar *scripts* maliciosos a outro usuário
- Isto pode ser feito encaixando conteúdos ativos dentro da requisição que gerará a saída HTML vista por outro usuário
- Uma vez que outros usuários acessarem este conteúdo, os navegadores não saberão que este não é confiável

Buffer Overflows

- Pessoas más intencionadas podem usar buffer overflows para corromper a pilha de execução de uma aplicação web
- Conseguem fazer isso enviando astutamente pedidos que fazem com que usuários executem códigos arbitrários
- Problemas com Buffer overflow são tipicamente difíceis de se identificar e são de difícil exploração por hackers
- Aplicações rodando em um servidor J2EE é imune a esse tipo de ataque

Falhas de Injeção

- Uma brecha onde *hackers* podem enviar ou "injetar" chamadas para o sistema operacional ou recursos externos como banco de dados
- Um exemplo comum é a de SQL
- Algumas precauções:
  - Validação rigorosa da informação da requisição do usuário
  - Em vez de usar comandos simples SELECT, INSERT, UPDATE, e DELETE, criar funções que execute comandos equivalentes
  - Providenciar que as aplicações terão o mínimo de privilégios, apenas o necessário para executar sua funcionalidade

Armazenamento Inseguro

- Aplicações WEB geralmente precisam armazenar informações como senhas, informações de cartões de crédito, entre outras
- Devido a natureza sensível dessas informações, estes geralmente são encriptados para prevenir acesso ocasional
- No entanto, a implementação de encriptar são geralmente fracas e ainda vulnerável às tentativas persistentes
- Há algumas áreas onde ocorrência de erros são comuns:
  - Falha ao encriptar dados críticos
  - Armazenamento inseguros de chaves, certificados e senhas
  - Armazenamento impróprio desses itens secretos na memória
  - Código de randomização pobre
  - Algorítimos de escolhas pobres
  - Tentar inventar um novo algorítimo de encriptar

Negação de Serviço

- Refere-se a ataques maliciosos feitos por *hackers*
- Fazem uso de requisição de concorrência múltiplas para o servidor
- Devido elevado número de tais requisições, o servidor começa a ser inundado e se torna incapaz de atender requisições de outros usuários
- Estes ataques fazem mais do que simplesmente consumir a banda toda do servidor
- Podem consumir recursos limitados importantes

Gerenciamento Inseguro de Configurações

- Determina quão segura sua aplicação pode ser
- Configurações impróprias no lado servidor podem desviar todos os esforços dos desenvolvedores em salvaguardar a aplicação
- Exemplos de erros de configuração de servidores:
  - Falhas de segurança não corrigidas no software do servidor
  - Falhas de segurança do servidor que permite a listagem de diretórios e/ou ataques a estes
  - *Backup* desnecessários ou arquivos de exemplos
  - Permissões impróprias de arquivos e diretórios
  - Serviços desnecessários

Gerenciamento Inseguro de Configurações

- Exemplos de erros de configuração de servidores (continuação)
  - Contas padrões com senhas padrões
  - Acesso administrativo ou funções de depuração
  - Mensagens de erro que passam informações excessivas
  - Má configuração de certificados SSL e ajustes errados na forma de encriptar
  - Uso de certificados auto-assinados conseguir autenticação
  - Uso de certificados padrões
  - Autenticação imprópria com sistemas externos

Sumário

- SSL
- Certificados de Segurança
- Criando um *Listener* HTTP Seguro
- Falhas de Segurança em Aplicações WEB

Parceiros

- **Os seguintes parceiros tornaram JEDI™ possível em Língua Portuguesa:**