Autenticação em aplicações *web*

Notas para a UC de “Segurança Informática”

Inverno de 11/12

Pedro Félix (pedrofelix *em* cc.isel.ipl.pt)

Instituto Superior de Engenharia de Lisboa

Limitações práticas

- *Deployability*
  - Protocolo HTTP
  - “User-agent”
- Aceitação por parte do utilizador
- Desempenho

Objectivos do atacante

- Falsificação existencial
  - Obter um autenticador válido
- Falsificação selectiva
  - Dado um utilizador, obter um autenticador válido para esse utilizador
- Obtenção da chave usada na criação de autenticadores

Classificação do atacante

- Atacante interrogativo
  - É utilizador do sistema
  - Interroga o sistema sobre a validade de autenticadores e observa as respostas
- Atacante observador
  - Capacidade de observar todas as mensagens de e para o servidor
- Atacante activo
  - Capacidade de observar e alterar todas as mensagens de e para o servidor
  - Atacante do modelo de Dolev-Yao

Fases

- O protocolo HTTP não possui estado – a autenticação tem de ser realizada em todos os pedidos
- Duas fases na autenticação
- Fase 1
  - Apresentação das credenciais pelo utilizador (ex.:”username+password”)
  - Obtenção dum autenticador
- Fase 2
  - Apresentação do autenticador automaticamente pelo “user-agent”

Autenticadores: “cookies” vs. URLs

- “Cookies”
  - Dependente da aceitação por parte do “user-agent”
  - Persistência entre sessões
  - Acesso em código de cliente

- URL
  - Acesso em código de cliente
  - Está presente no “header” “Referer”

Autenticadores: implementação

- Identificador de sessão
  - Informação sobre a sessão presente no servidor
  - “Cookie” contém o identificador (chave) para aceder a essa informação
  - Deve ser computacionalmente infazível criar um identificador válido
    - Geração criptográfica de números aleatórios
- “Message Authentication Code”
  - Informação sobre a sessão presente no “cookie”
  - “Cookie” protegido por um MAC
  - Se a confidencialidade for requisito, cifrar o conteúdo do “cookie”
  - Maior escalabilidade – chaves partilhadas por todos os servidores do “web farm”

Autenticadores: implementação

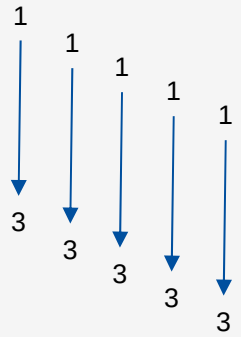
- Associar autenticador a atributos do cliente
  - Endereço, “user-agent”
- Mecanismo de validade temporal próprio
  - Não utilizar o mecanismo dos “cookies”
  - Limite da validade temporal
    - Presente na informação de sessão
    - Presente no “cookie” (protegido pelo MAC)
- “Logout”/Revogação
  - Invalidar a sessão
  - Colocar o “cookie” numa lista de revogação (até à expiração da validade)
- Protecção dos “cookies”
  - Protecção do transporte através do uso de SSL
  - Protecção no cliente – *flag* “HttpOnly” (IExplorer)

Ataques de dicionário “online”

- Atacante interrogativo
- Técnicas clássicas de protecção:
  - Atraso na resposta/”backoff”
  - Bloqueamento
- Problemas das técnicas clássicas
  - Dificuldade de implementação no cenário web
  - Disponibilidade do serviço
  - Ataques globais
- Ataques de dicionário globais
  - Ataque a *qualquer* utilizador do sistema

Bloqueamento: problemas

- Implementação típica
  1. Obter a informação de verificação do utilizador (v) ou se a conta está bloqueada
  2. Verificar a informação de autenticação - g(v)(a)
  3. Se o resultado é falso, actualizar o número de tentativas de autenticação do utilizador
- Problemas:
  - Número de tentativas realizadas em simultâneo entre os passos 1 e 3 maior do que o número de tentativas permitidas
    - Seriar o processamento da autenticação coloca problemas de implementação
    - Disponibilidade



Atraso: problemas

- Implementação típica
  1. Obter a informação de verificação do utilizador (v) ou se a conta está bloqueada
  2. Verificar a informação de autenticação - g(v)(a)
  3. Esperar tempo constante ou dependente do número de tentativas erradas
  4. Apresentar o resultado ao utilizador
- Problemas:
  - Número de tentativas realizadas em simultâneo entre os passos 1 e 3
    - Seriar o processamento da autenticação coloca problemas de implementação e de disponibilidade
  - Alguns cálculos
    - Seja N o número de pedidos aceites por unidade de tempo
    - Seja T o atraso inserido pelo pedido
    - Tempo para o teste de M candidatos = T + M/N

Solução: aumentar o custo dos pedidos

- Diminuir o número de pedidos realizados através do aumento do seu custo para o cliente
- Desafio computacional que tem de ser calculado pelo "user-agent" cliente antes da realização do pedido
  - Necessita de computação do lado do cliente (ex. Javascript)
- CAPTCHA - "Completely Automated Public Turing Test to Tell Computers and Humans Apart" - http://www.captcha.net/
  - Fácil para humanos
  - Difícil para computadores

Utilização de CAPTCHA (1)

- Técnica 1
  - Requerer a resposta a um CAPTCHA como condição à realização da autenticação
- Problemas
  - Aceitação por parte do utilizador
  - Custo computacional para o servidor
    - Geração
    - Armazenamento

Utilização de CAPTCHA (2)

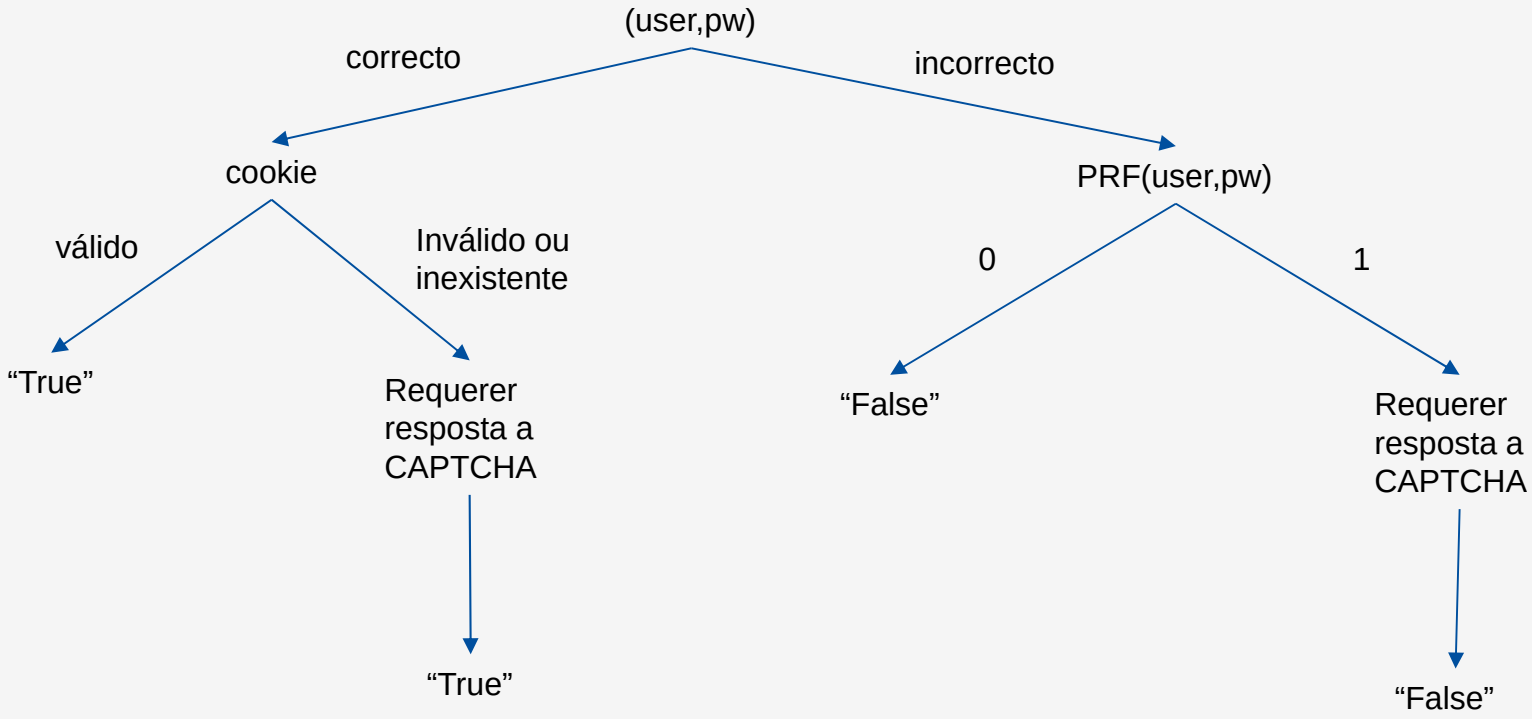
- Técnica 2
  - Com probabilidade P, requerer a resposta a um CAPTCHA como condição à realização da autenticação
- Se P for pequeno
  - Diminui o custo computacional para o servidor
  - Aumenta a aceitabilidade do utilizador
- Problema
  - Atacante ignora os casos em que é requerido CAPTCHA

Utilização de CAPTCHA (3)

- Técnica 3
  - Requerer CAPTCHA apenas depois duma autenticação errada
- Se P for pequeno
  - Aumenta a aceitabilidade do utilizador
- Problema
  - Não protege contra ataques globais

Algoritmo de Pinkas e Sander (1)

- Requisito: Colocação de “cookies” persistentes nos “user-agents” onde foi realizada uma autenticação com sucesso



Algoritmo de Pinkas e Sander (2)

- PRF(user,pw):
  - Se (user,pw) existem em M, retornar M[(user,pw)]
  - Gerar b $\in$ {0,1} com probabilidade Pr{b=1} = p
  - M[(user,pw)] = b
  - retornar b

- Aceitabilidade
  - Utilizador com “cookie”
    - (user,pw) correcto – nunca responde a CAPTCHAs
    - (user,pw) incorrecto – responde a CAPTCHAs com probabilidade p
  - Utilizador sem “cookie”
    - responde sempre a CAPTCHAs (tipicamente só uma vez, depois fica com “cookie”)

Algoritmo de Pinkas e Sander (3)

- Custo computacional
  - Se os utilizadores tiverem “cookies”, a geração de CAPTCHAS é realizada apenas nas tentativas erradas e com probabilidade p
  - O CAPTCHA associado a cada par (user,pw) pode ser sempre o mesmo (armazenar os CAPTCHAs associados a pares válidos)

- Segurança
  - Considerando que o conjunto de “passwords” prováveis tem dimensão N
  - A “password” correcta pertence a um conjunto de dimensão pN, em que o teste obriga à resposta do CAPTCHA
  - Número médio de tentativas com CAPTCHA = pN/2
  - Custo médio = pN/2 x Custo de resolver o CAPTCHA

- É fundamental que o atacante não obtenha informação sobre a correcção da password através de outras formas (ex. tempo de resposta)

Referências

- Securing Passwords Against Dictionary Attacks, Benny Pinkas and Tomas Sander, 2002