Segurança da Informação

David Valentim Dias
Desenvolvedor, Impulso

Desenvolvimento e Segurança

2

Open Web Application Security Project (OWASP)

Fundação que reune recursos sobre as falhas de segurança mais comuns

- Tutoriais
- Ferramentas (pentest, scripts, ...)
- Treinamento
- Fomento científico

owasp.org (https://owasp.org)

3

OWASP TOP 10

- Injection (https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A1-Injection)
- Broken Authentication (https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A2-Broken_Authentication)
- Sensitive Data Exposure (https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A3-Sensitive_Data_Exposure)
- XML External Entities (XXE) (https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A4-XML_External_Entities_(XXE))
- Broken Access Control (https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A5-Broken_Access_Control)
- Security Misconfiguration (https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A6-Security_Misconfiguration)
- Cross-Site Scripting (XSS) (https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A7-Cross-Site_Scripting_(XSS))
- Insecure Deserialization (https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A8-Insecure_Deserialization)
- Using Components with Known Vulnerabilities (https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A9-Using_Components_with_Known_Vulnerabilities)
- Insufficient Logging & Monitoring (https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A10-Insufficient_Logging%252526Monitoring)

Injection

Geralmente ocorre com a entrada de dados inesperada

```
func (db *DB) GetUserByName(name string) (int, error) {

    var id int

    query := "select oid from users where name = '" + name + "'"

    err := db.QueryRow(query).Scan(&id)
    if err != nil {
        return id, err
    }

    return id, nil

}
```

Run

Injection possível com:

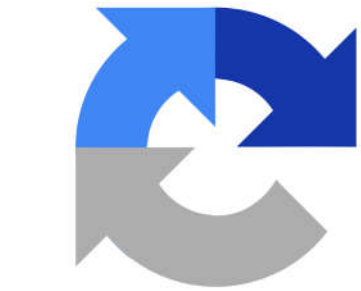
```
"' or '1'='1"
```

5

Broken Authentication

Falhas no modelo de autenticação do usuário:

- Senhas fracas (admin/admin)
- Ataques de força bruta
- Recuperação fraca de senha (Qual o nome do seu cachorro?)
- Senhas não criptografadas ou cripto. fraca
- Sessões infinitas



reCAPTCHA

Sensitive Data Exposure

Os dados armazenados e em trânsito precisam de alguma proteção?

- Usar criptografia sempre que possível. HTTP(**S**) = 60%
- Não armazenar dados desnecessários (PCI DSS)
- Desativar cache em respostas com dados sensíveis

Problema

Quando estamos cadastrando um novo funcionário e caso ele exista em outro Org. exibimos o número do cartão que ele possui nas outras Orgs.

Solução

Ocultar parte do número do cartão com *

XML External Entities (XXE)

- Aplicações que aceitam XML como entrada devem desabilitar/validar DTD (Document Type Definitions)
- Atualizar os processadores de XML
- Atualizar SOAP >= 1.2

```
<!ENTITY xxe SYSTEM "file:///dev/random" >]>
```

8

Broken Access Control

Politicas que forçam o usuário a conseguir agir apenas dentro do seu contexto.

- Modificar a URL usando chaves primárias para ter acesso/edição de outra conta
- CORS mal configurado
- Elevação de privilégios

```
/api/boleto/:ordem
```

Sempre verificar se o CNPJ atual pode acessar pedidos que não é dono.

Security Misconfiguration

- Servidor exibindo informações para desenvolvimento em ambiente de produção
- Recursos/serviços desnecessários ativos
- Contas padrões
- Software desatualizado

Problema

Servidor da intranet exibia erros em modo de depuração.

Solução

Exibir "Oops, tives um problema!" e logar o erro específico internamente.

Cross-Site Scripting (XSS)

Inclusão de dados (scripts) não validados no html

localhost:8080/boleto?numerodoc=123456 (http://localhost:8080/boleto?numerodoc=123456)

localhost:8080/boleto?numerodoc=<script>document.location='http://www.google.com'</script>

(http://localhost:8080/boleto?numerodoc=<script>document.location='http://www.google.com'</script>)

```
func boleto(w http.ResponseWriter, r *http.Request) {
    doc := r.URL.Query().Get("numerodoc")
    content := `<p>Número do documento: `
    content += doc
    html := header + content + footer
    w.Write([]byte(html))
}
```

Run

Correção

```
t, _ := template.New("").Parse(header + `<p>Número do documento: {{.}}`
t.Execute(w, doc)
```

11

Insecure Deserialization

Situação

O cookie de autenticação possui uma estrutura com dados do usuário

```
{
    user: "david.valentim@bol.com.br",
    pass: "12345678"
}
```

O cookie não foi criptografado e o campo user é verificado em algumas rotinas para dar permissão.
O atacante modifica o cookie:

```
{
    user: "admin@bol.com.br",
    pass: "12345678"
}
```

Atacante possui as permissões de um usuário diferente 12

Using Components with Known Vulnerabilities

Ocorre principalmente com software que não possuem manutenção ou que não foram atualizados

github.com/satori/go.uuid (github.com/satori/go.uuid)

```
Affected versions of this package are vulnerable to Insecure Randomness
```

Caso do Drupal e IPT... 13

Insufficient Logging & Monitoring

- Falta de log e monitoramento
- Mensagens de log que não contribuem
- Alertas em tempo real
- In 2016, identifying a breach took an average of 191 days – plenty of time for damage to be inflicted.

Thank you

David Valentim Dias
Desenvolvedor, Impulso
david.valentim@bol.com.br (mailto:david.valentim@bol.com.br)
https://dvdscripter.wordpress.com (https://dvdscripter.wordpress.com)