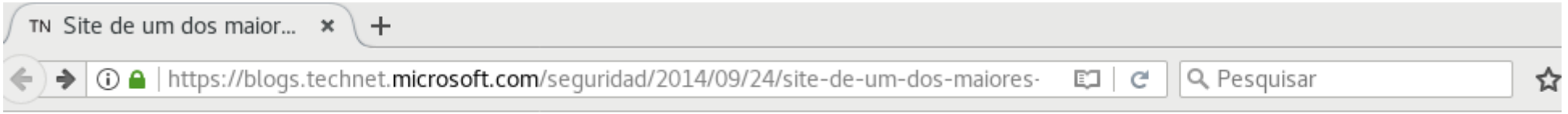
Estratégias de Segurança para Desenvolvimento de Software

Italo Valcy e Kaio Rodrigo
CoSIC / STI-UFBA

Aplicações como alvo nos ataques



Site de um dos maiores jornais do Brasil foi comprometido com malware que tentou alterar as configurações de DNS nos roteadores das vítimas.

septiembre 24, 2014 by Teresa Zancanelli Ghiorzoe // 0 Comments

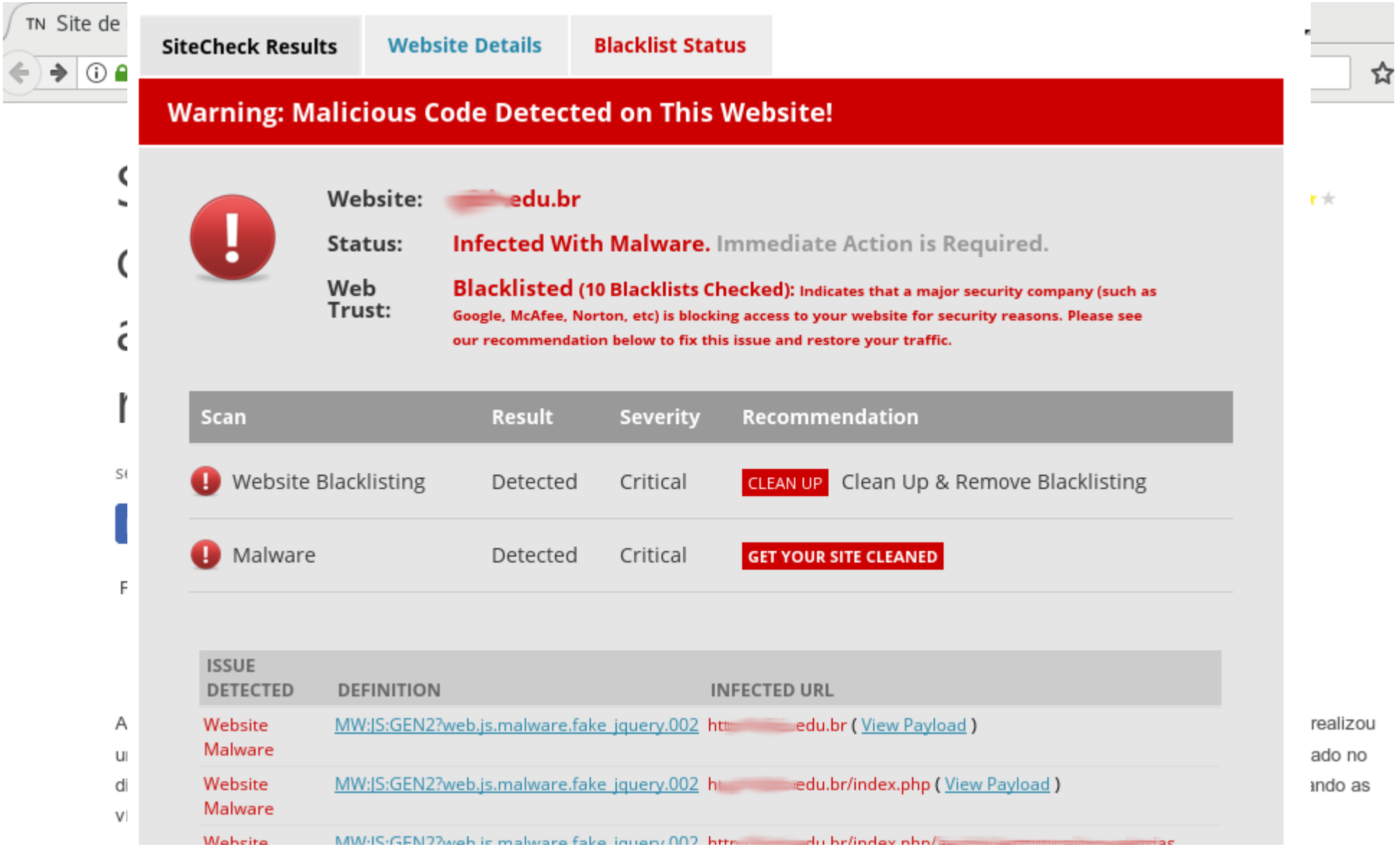
Share 0 | 0 | 0

Por Michael Mimoso – em 12 de setembro de 2014

A Empresa de segurança WebPubli publicou um relatório ontem indicando que o site do jornal *"O Estadão"*, estava carregando iframes que realizou um ataque de **força bruta** contra roteadores residenciais das vítimas, utilizando credenciais de administrador. Um ataque semelhante foi relatado no dia 2 de Setembro pela Kaspersky Lab e pelo investigador de segurança Fabio Assolini, quem disse que viu semelhante redirecionamento levando as vítimas a sites de *phishing* que se faziam passar por bancos no Brasil.

https://blogs.technet.microsoft.com/seguridad/2014/09/24/site-de-um-dos-maiores-jornais-do-brasil-foi-comprometido-com-malware-que-tentou-alterar-as-configuraes-de-dns-nos-roteadores-das-vtimas/

Aplicações como alvo nos ataques



Aplicações como alvo nos ataques

Aplicações como alvo nos ataques

- Grande impacto à imagem da organização

SEGURANÇA

Verizon vai comprar Yahoo mais barato por conta de ciberataques

Convergência Digital ... 21/02/2017 ... Convergência Digital

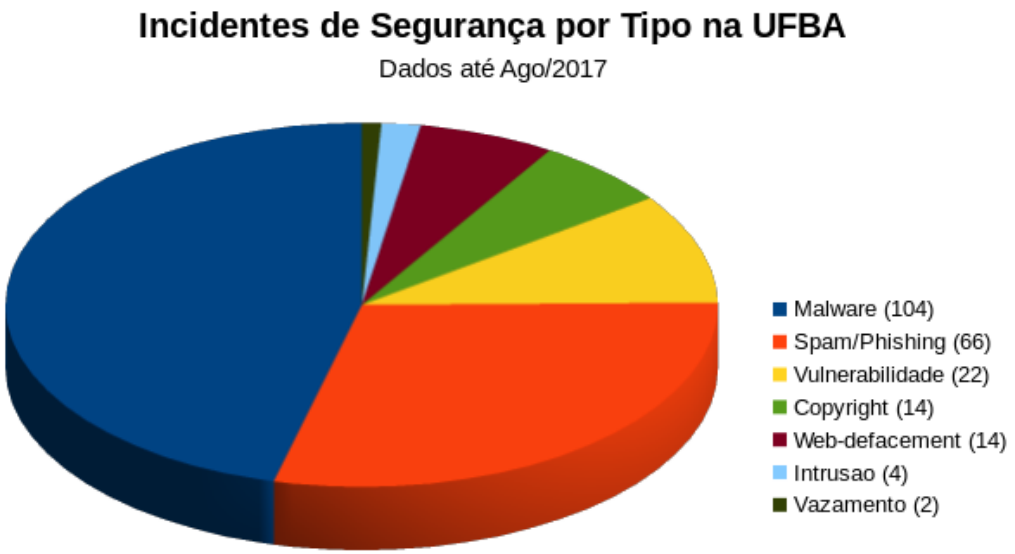
A operadora americana Verizon informou ao mercado nesta terça, 21/2, que vai pagar menos pela Yahoo. O negócio, anunciado em meados do ano passado por US$ 4,83 bilhões – quase R$ 16 bilhões – sairá por US$ 4,48 bilhões. O 'desconto' de quase R$ 1 bilhão (US$ 350 milhões) se deve aos dois grandes ciberataques admitidos pela Yahoo.

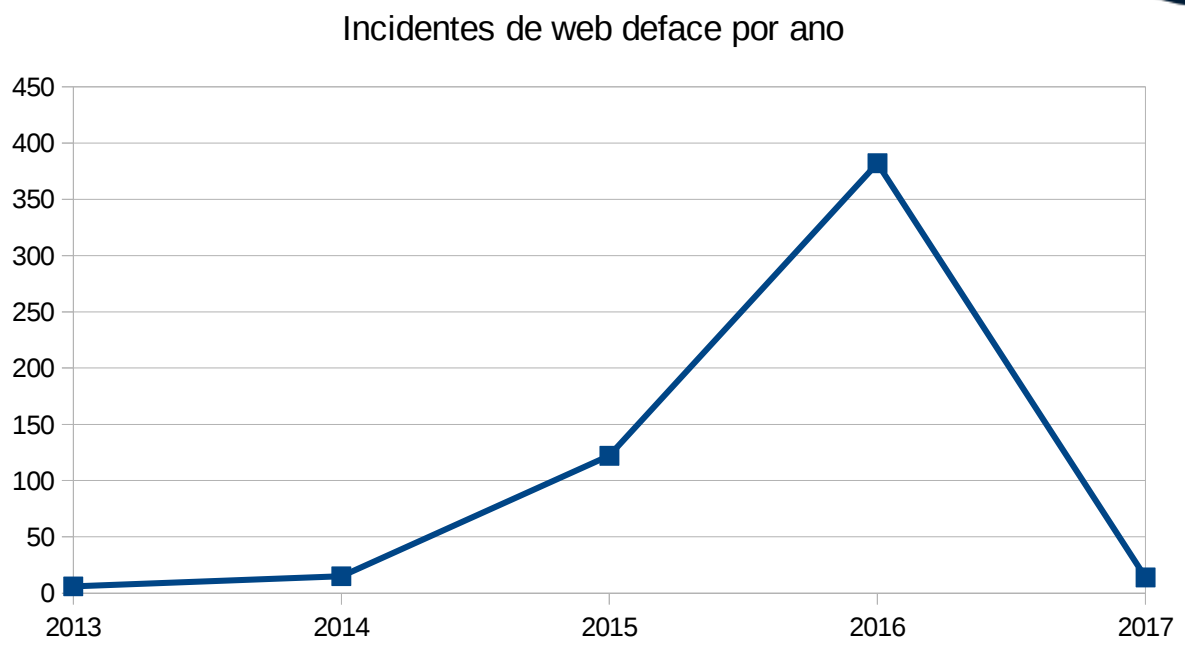
Em setembro de 2016, a Yahoo admitiu que em 2014 hackers roubaram dados pessoais de mais de 500 milhões de usuários. Dois meses depois, em dezembro, admitiu outro ataque, em 2013, que afetou aproximadamente 1 bilhão de usuários.

Naquele mesmo mês, a comissão reguladora financeira dos Estados Unidos (SEC) pediu documentos sobre os ataques avisou que abriu uma investigação para verificar se o Yahoo informou aos investidores previamente sobre os dois grandes vazamento de informação.

Aplicações como alvo nos ataques

- Diversas origens de problemas
- Alto impacto ao negócio
- Contenção de recursos

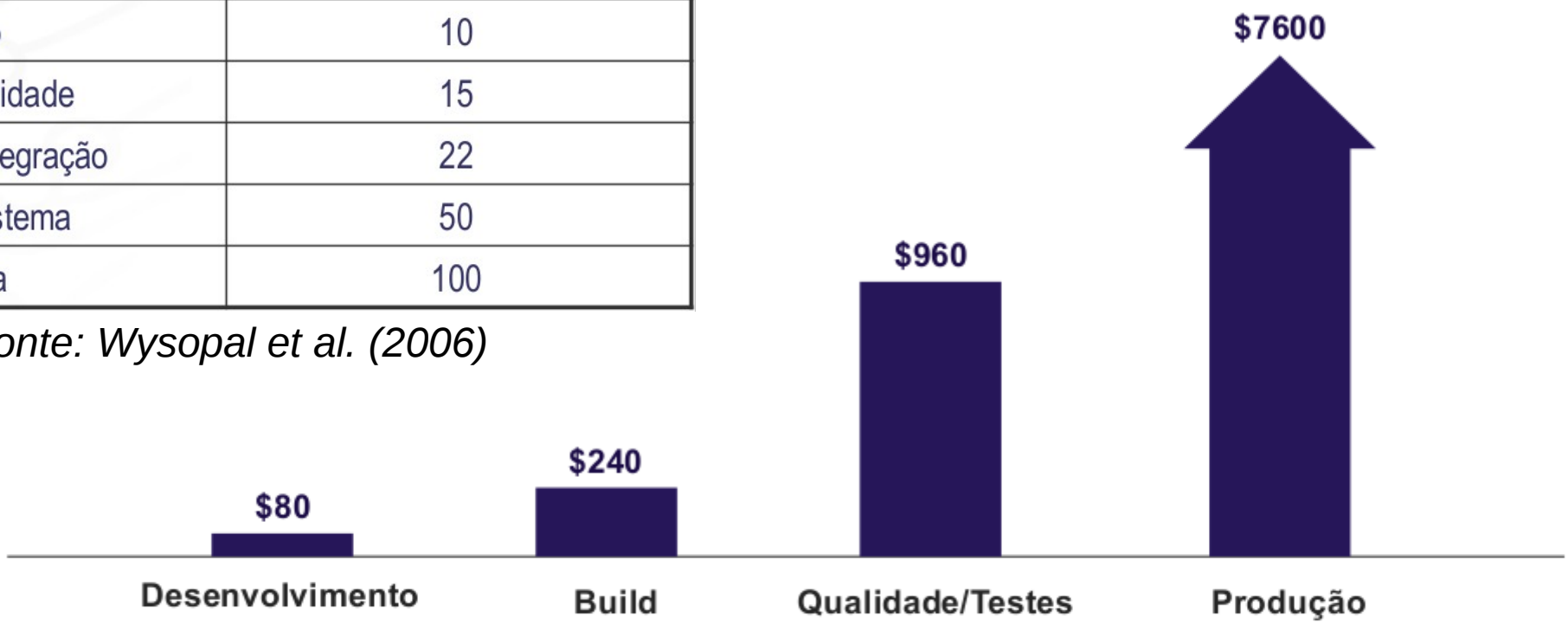




Custos de correção

Fase	Custo relativo para correção
Definição	1
Projeto alto nível	2
Projeto detalhado	5
Codificação	10
Teste de unidade	15
Teste de integração	22
Teste do sistema	50
Pós-entrega	100

Fonte: Wysopal et al. (2006)



CUSTO DE UM BUG DE SEGURANÇA EM CADA ESTÁGIO DE DESENVOLVIMENTO

Fonte: Ponemon Institute: National Institute of Standards and Technology

Estratégias aplicáveis

- Tratamento de Incidentes de Segurança
- Hardening de infraestrutura
- Pentest de aplicações web
- Análise estática de código
- Desenvolvimento Seguro de Software

Tratamento de Incidentes

- Mesmo com todas as ações de prevenção, incidentes inevitavelmente ocorrem
  - Necessidade de um Plano de Gestão de Incidentes

- Análise forense, correlação de eventos, cyber threat intelligence, recuperação e remediação

Tratamento de Incidentes

- Artigo “Estabelecimento de CSIRTs e Processo de Tratamento de Incidentes de Segurança em Instituições Acadêmicas Brasileiras: estudo de caso da parceria CAIS/RNP e UFBA” em TICAL 2017
  - https://goo.gl/nRi9MJ
- Artigo “Tratamento Automatizado de Incidentes de Segurança da Informação em Redes de Campus” em SBSEG 2011
  - https://goo.gl/mAhcxg

Hardening da Infraestrutura

- Consiste em modificar as configurações do sistema para torná-lo mais robusto em termos de segurança
  - Templates de configuração segura
  - Detecção e prevenção de ataques
  - Falha fechada / falha segura
  - Virtual Patching
- Projetos de repositório de hardening
- Diversas ferramentas de apoio:
  - Modsecurity, MS URL Scan, Docker / Cloudlinux, IIS application pools, OSSEC, fail2ban

Hardening da Infraestrutura

- Minicurso “Implementando uma Infraestrutura de Rede Segura” no 20º SCI/RNP (2014)
  - https://esr.rnp.br/sci/2014
- Palestra “Apache Mod_Security” no EnSI 2015
  - https://goo.gl/mm3SHt
- Palestra “Segurança de aplicações web - como aumentar a segurança dos sites da sua organização” no EnCSIRTs 2015
  - https://goo.gl/qBTyGu
- Artigo “Estratégias para Segurança em Aplicações Web: Auditoria, Infraestrutura e Filtragem de Ataques” no WTICIFES 2016
  - https://goo.gl/o1XCHf

Pentest de aplicações web

Teste de invasão é um método utilizado para verificar a segurança de uma aplicação por meio da simulação de ataques reais explorando as vulnerabilidades encontradas.

- Navegadores web
- Proxies de interceptação
- Web spiders
- Fuzzers
- Varredores de portas e serviços
- Varredores de vulnerabilidades
- Outras ferramentas

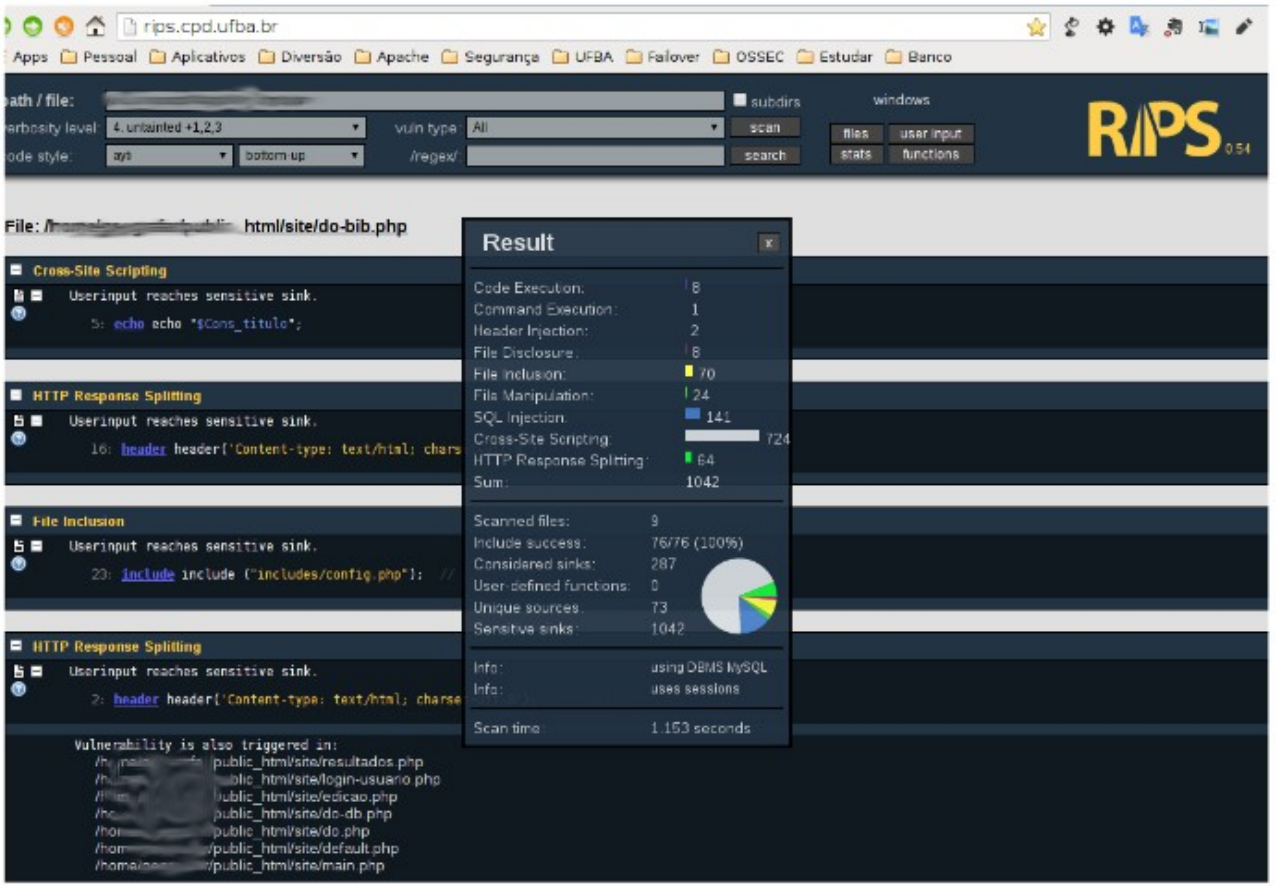
Pentest de aplicações web

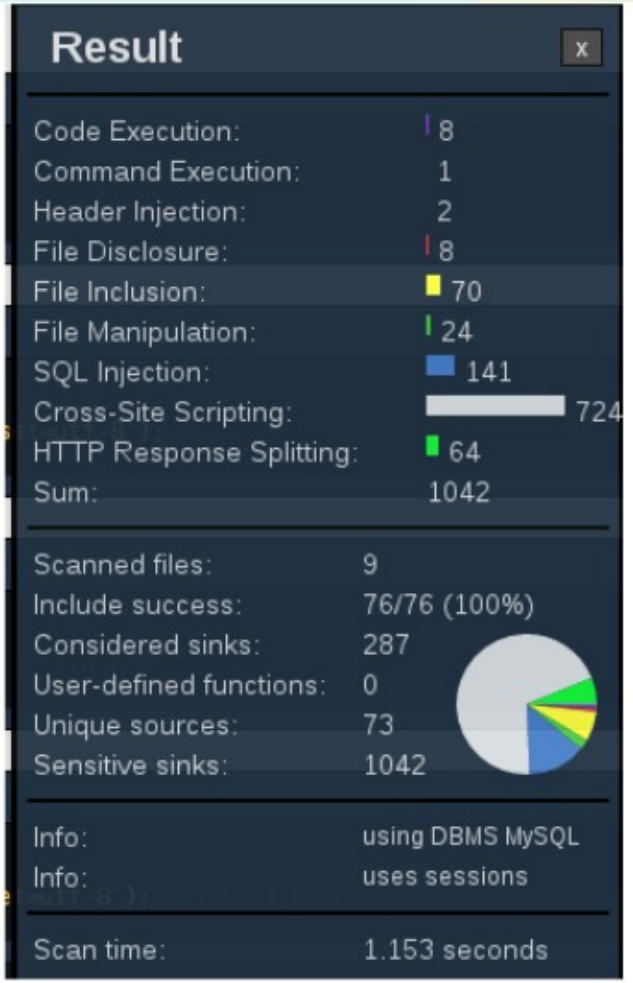
- Existem diversas ferramentas que apoiam o processo de auditoria
  - w3af, sqlmap, arachni, openvas, nmap, nikto, burp suite, etc.
- Diferencial é o conhecimento do auditor
- Sugestão de treinamento:
  - SEG9 / ESR-RNP
  - https://esr.rnp.br/seg9

Análise estática de código

- Busca padrões de código fonte sujeito a falhas (bugs de segurança)
- Teste white-box
- Complexo e custoso
- Pode trazer resultados bem interessantes, mas também falsos positivos

Análise estática de código





Análise estática de código

sonar	Sonar PDF Report	Multi Chapter Simple Parent Project

1.2. Issues Analysis

Most violated rules	
Security - Weak TrustManager implementation [find-sec-bugs]	2
Malicious code vulnerability - May expose internal representation by returning reference to mutable object	12
Malicious code vulnerability - May expose internal representation by incorporating reference to mutable object	12

Análise estática de código

Brakeman Report

Application Path	Rails Version	Brakeman Version	Report Time	Checks Performed
/tmp/[redacted] /var/www [redacted]	5.0.1	3.6.1	2017-03-28 11:45:39 -0300 3.460352043 seconds	BasicAuth, BasicAuthTimingAttack, ContentTag, CreateWith, CrossSiteScripting, DefaultRoutes, Deserialize, DetailedExceptions, DigestDoS, DynamicFinders, EscapeFunction, Evaluation, Execute, FileAccess, FileDisclosure, FilterSkipping, ForgerySetting, HeaderDoS, I18nXSS, JRubyXML, JSONEncoding, JSONParsing, LinkTo, LinkToHref, MailTo, MassAssignment, MimeTypeDoS, ModelAttrAccessible, ModelAttributes, ModelSerialize, NestedAttributes, NestedAttributesBypass, NumberToCurrency, QuoteTableName, Redirect, RegexDoS, Render, RenderDoS, RenderInline, ResponseSplitting, RouteDoS, SQL, SQLCVEs, SSLVerify, SafeBufferManipulation, SanitizeMethods, SelectTag, SelectVulnerability, Send, SendFile, SessionManipulation, SessionSettings, SimpleFormat, SingleQuotes, SkipBeforeFilter, StripTags, SymbolDoSCVE, TranslateBug, UnsafeReflection, ValidationRegex, WithoutProtection, XMLDoS, YAMLParsing

Summary

Scanned/Reported	Total
Controllers	25
Errors	0
Ignored Warnings	0
Models	31
Security Warnings	10 (7)
Templates	122

Warning Type	Total
Cross Site Scripting	3
Dynamic Render Path	1
File Access	5
Session Setting	1

Security Warnings

Confidence	Class	Method	Warning Type	Message
High	GlossaryController	[redacted]xml	File Access	Model attribute used in file name near line 69: File.open(("app/views/glossary/
High	GlossaryController	[redacted]	File Access	Model attribute used in file name near line 72: File.open(("app/views/glossary/
High			Session Setting	Session secret should not be included in version control near line 22
Medium	PagesController	[redacted]	Dynamic Render Path	Render path contains parameter value near line 3: render(template => "pages/#{p
Weak	GlossaryController	[redacted]	File Access	Parameter value used in file name near line 119: File.open(File.join("app/views

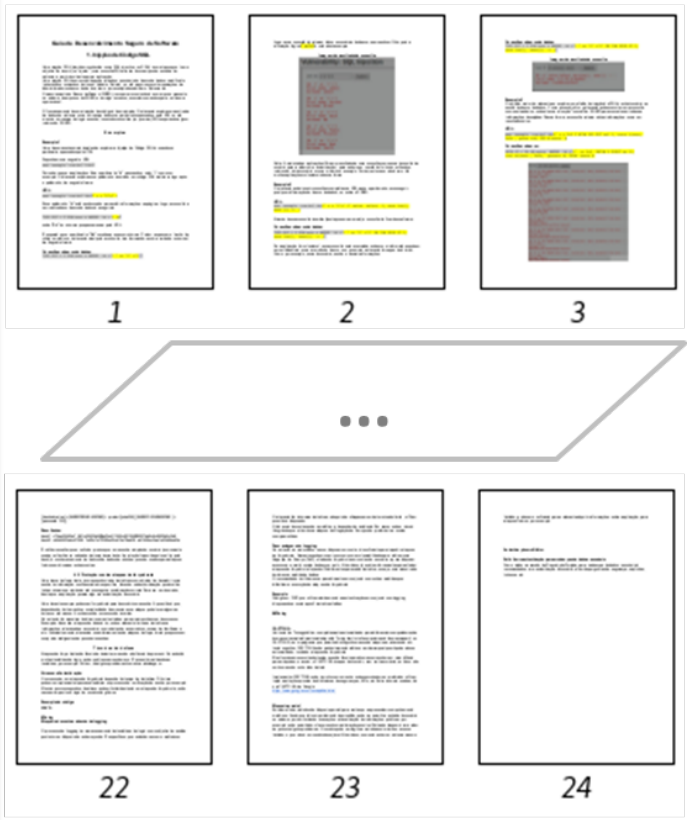
Desenvolvimento Seguro de Software

SDL (Security Development Lifecycle) é uma disciplina que utiliza metodologias, normas e ferramentas para desenvolver software seguro, evitando falhas de segurança desde sua conecpção.

- Gerenciar os riscos de segurança em todo ciclo de vida do software
- Entender as falhas de segurança mais comuns e implantar controles de segurança
- Implantar testes de segurança específicos

Desenvolvimento Seguro de Software

- Guia para desenvolvimento seguro de software, customizado e especializado
  - SQL Injection
  - XSS
  - CSRF
  - Autenticação Segura



- Projeto em andamento

Cross-site Scripting

XSS é uma injeção de código onde o atacante insere um script malicioso na aplicação alvo, através de campos de busca, mensagens de fórum, etc.

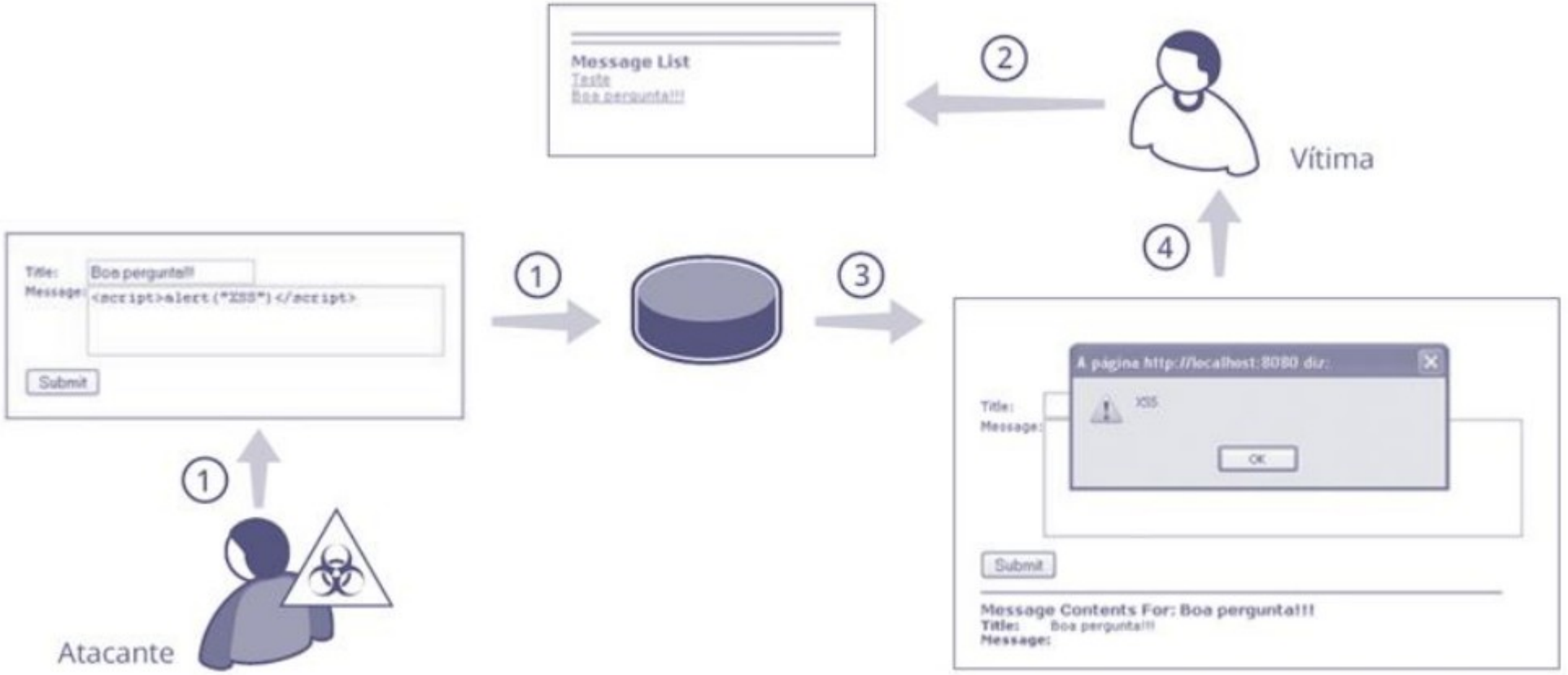
O navegador executa automaticamente o código inserido pelo invasor, alterando o comportamento da aplicação. Muitas vezes de forma silenciosa.

Roubo de informações do usuário

Alterar o conteúdo da página

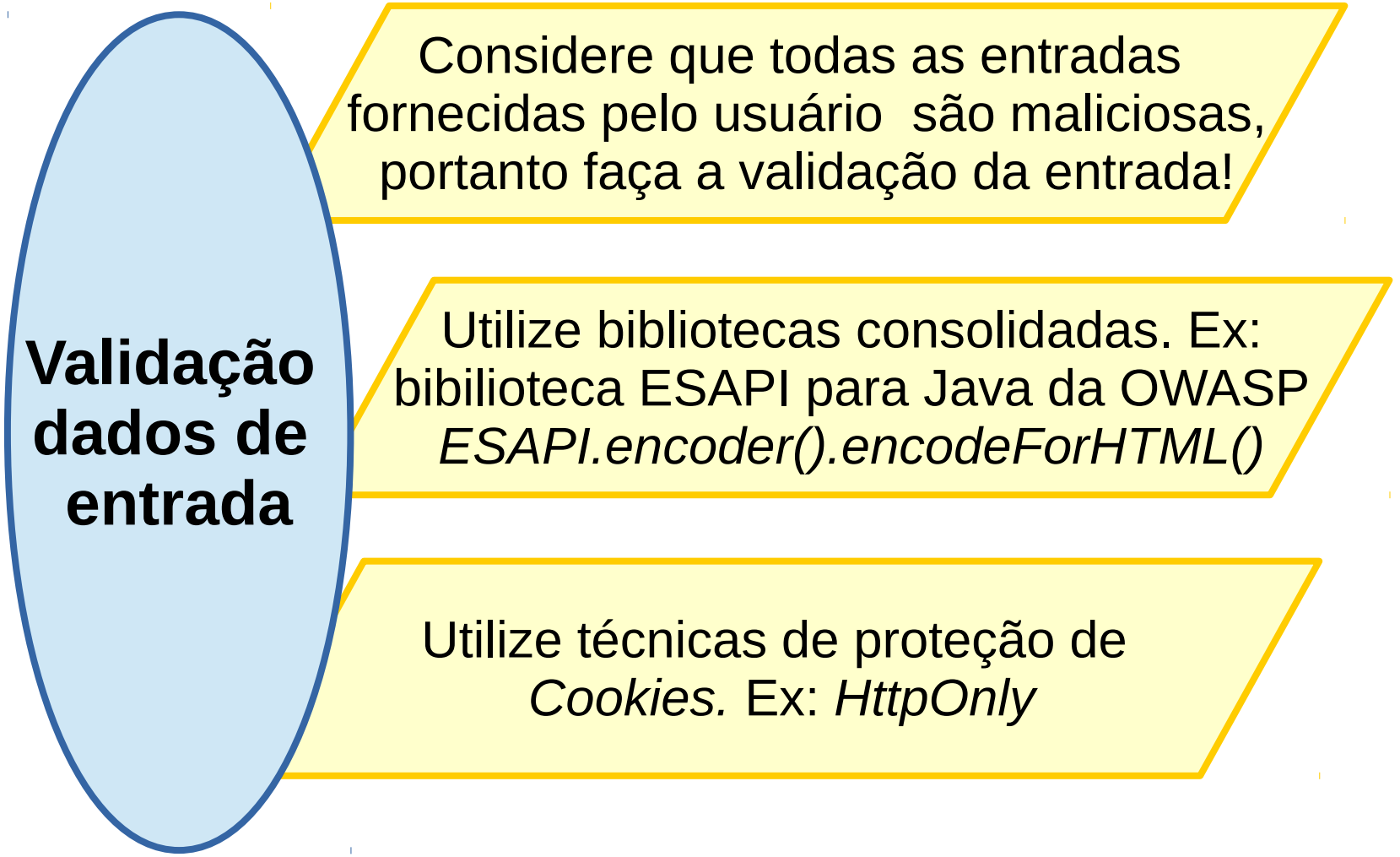
Executar involuntariamente funções da aplicação.

Cross-site Scripting



Fonte: Curso pentest de aplicações web, ESR-RNP

Cross-site Scripting



Injeção de código (SQL)

Atacante insere código malicioso na aplicação para alterar seu comportamento original. Pode causar diversos Impactos:

- Confidencialidade
- Disponibilidade
- Integridade

Injeção de código (SQL)

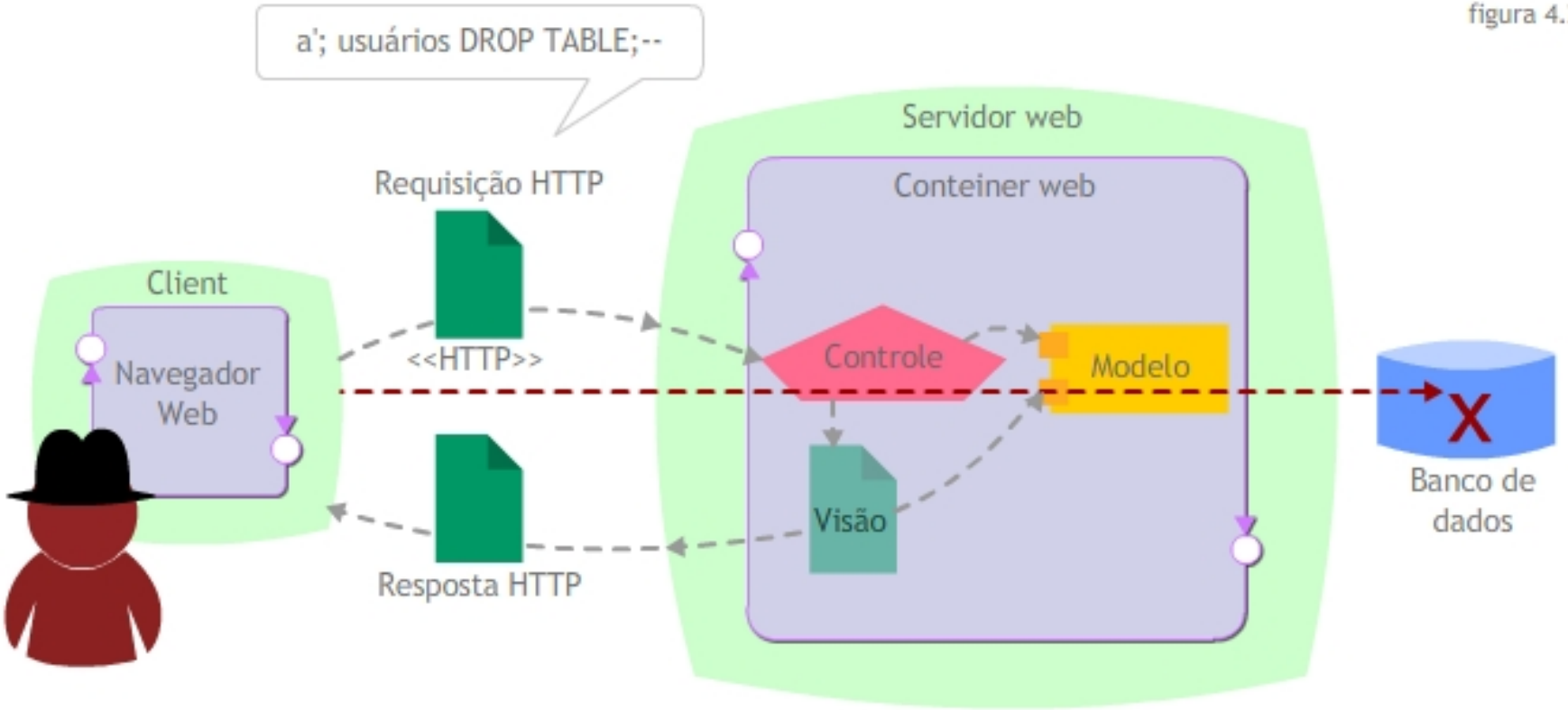
```
select * from user_data
        where last_name = '' or 1=1--'
```

Enter your last name: ' or 1=1; – Go!

USERID	FIRST_NAME	LAST_NAME	CC_NUMBER	CC_TYPE	COOKIE	LOGIN_COUNT
101	Joe	Snow	987654321	VISA		0
101	Joe	Snow	2234200065411	MC		0
102	John	Smith	2435600002222	MC		0
102	John	Smith	4352209902222	AMEX		0
103	Jane	Plane	123456789	MC		0
103	Jane	Plane	333498703333	AMEX		0
10312	Jolly	Hershey	176896789	MC		0
10312	Jolly	Hershey	333300003333	AMEX		0
10323	Grumpy	White	673834489	MC		0
10323	Grumpy	White	33413003333	AMEX		0
15603	Peter	Sand	123609789	MC		0
15603	Peter	Sand	338893453333	AMEX		0
15613	Joesph	Something	33843453533	AMEX		0

Fonte: Curso SEG9 / ESR-RNP

Injeção de código (SQL)



Fonte: Escola de Inclusão Sociodigital, Serpro

Injeção de código (SQL)

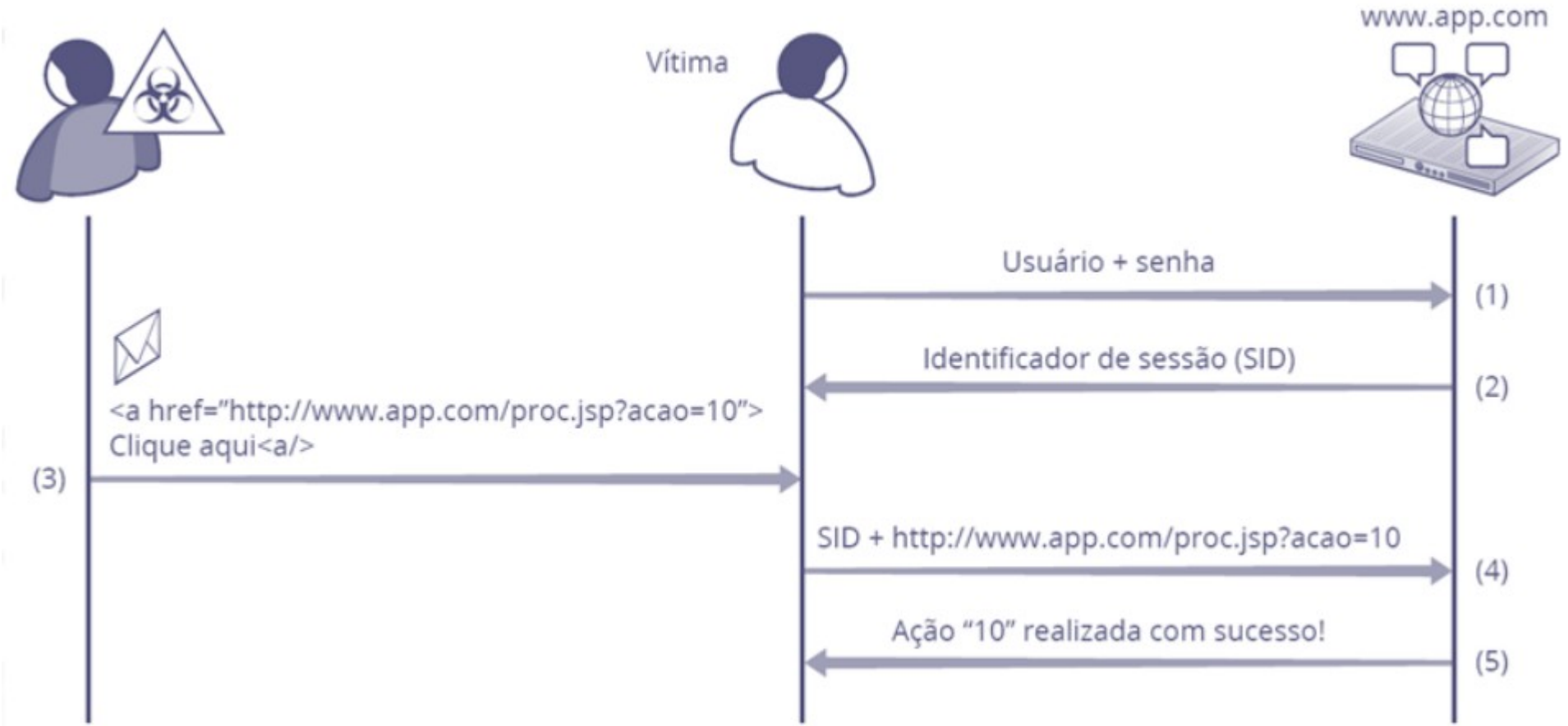
- A maioria das linguagens e bibliotecas já possui métodos de execução de **consultas parametrizadas**, filtro de dados da entrada do usuário, etc.

- Apesar de SQLi ser um ataque simples e antigo, ainda há muitas aplicações vulneráveis e uma atenção especial deve ser dada a essa vulnerabilidade.

Cross-site Request Forgery

Cross Site Request Forgery (CSRF) é um ataque que se aproveita de uma sessão de usuário já estabelecida com a aplicação vulnerável, para realizar operações de maneira automática, sem o conhecimento e consentimento da vítima.

Cross-site Request Forgery




```
<img src="http://192.168.0.1:80/adv_filters_mac.cgi?
 editRow=-1&delrow=-1&filters=1&macFilter=0&name=&
 mac1=&mac2=&mac3=&mac4=&mac5=&mac6=" />
```

Fonte: SEG9 / ESR-RNP

Cross-site Request Forgery

- Contra-medidas
  - Token Anti-CSRF
  - Uso de APIs e *Frameworks* padrões
  - Proteção contra ataques de *Clickjacking*
  - Proteção contra sobreposição de *iframes*

Cross-site Request Forgery

- Token Anti-CSRF (Ex: python-flask_wtf)

=> Código do template HTML (view)

```
<div class="login-box-body">
  <form method="post">
    {{ form.hidden_tag() }}
    <div class="form-group has-feedback">
      {{ form.email(required=true, class="form-c
      <span class="glyphicon glyphicon-envelope
    </div>
```

=> Código HTML resultante

```
<form method="post">
  <input id="csrf_token" name="csrf_token"
  value="ImRkMTlkMjYxMjNlZDZhMTFmMmJhMDczNTJjOWU1MTQwMzAwYzNiNTgi.DK13Qg.IPB8eNaT2AwvHkMKJOGmmGnlQSQ" type="hidden">
  <div class="form-group has-feedback">
    <input id="email" class="form-control" name="email" placeholder="E-mail" required="" value="" type="email"> ev
```

=> Código de validação do controller

```
@app.route('/login/', methods=['GET', 'POST'])
def login():
    ignore_captcha = login_attempts_count() <= 2
    form = LoginForm(ignore_captcha)

    if form.validate_on_submit():
```

Ataques ao mecanismo de autenticação

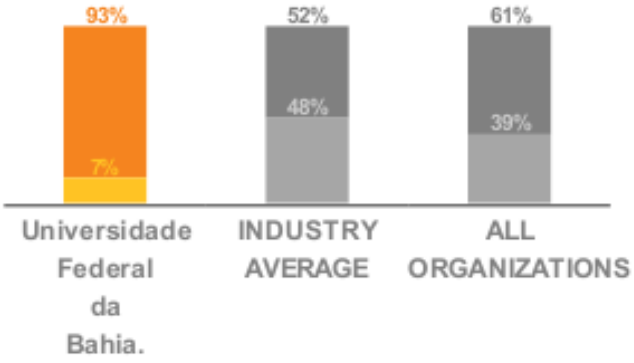
Um dos principais requisitos das aplicações web é a autenticação entre as Entidades.

- Algo que se sabe.
- Algo que se tem.
- Algo que se é.

Ataques ao mecanismo de autenticação

Um dos principais requisitos das aplicações web é a autenticação entre as Entidades.

- Algo que se sabe.
- Algo que se tem.
- Algo que se é.



Ataques ao mecanismo de autenticação

- Itens cobertos no guia:
  - Armazenamento seguro de senhas
    - Uso de SALT
  - Proteções contra ataques de força-bruta
    - Registro de eventos (Logging) + Bloqueio de IP
    - Captcha
  - Autenticação de múltiplos fatores

Ataques ao mecanismo de autenticação

- SALT para proteção de senhas:

Username	Salt Value	String to be Hashed	Hashed Value = SHA256(Password + Salt Value)
user1	E1F53135	password123 + E1F53135	72ae25495a7981c40622d49f9a52e4f1565c90f048f59027bd9c8c8900d5c3d8
user2	84B03D03	password123 + 84B03D03	b4b6603abc670967e99c7e7f1389e40cd16e78ad38eb1468ec2aa1e62b8bed3a

E depois disso tudo, estamos seguros?

Conscientização das pessoas

Desenvolvimento Seguro de Software

- Incorporar os cuidados com segurança em todas as fases do desenvolvimento de software
- Criar cultura de dev seguro entre a equipe
- Aumentar casos de teste relacionados com segurança
- Selecionar o conjunto de ferramentas de apoio no processo de auditoria

Obrigado!!!
;-)

Perguntas?

Italo Valcy, Kaio Rodrigo
{italovalcy, kaio.rodrigo}@ufba.br