# Cidades Inteligentes: Segurança em aplicações WEB em Prefeituras

## Aplicações PHP e Cobit 5

Edson Lüders
Mestrando - Mestrado Profissional em Gestão de Redes e Serviços
Pontifícia Universidade Católica de Campinas
Campinas, SP Brasil
edson.l@puccampinas.edu.br

David Bianchini
Prof. Dr. - Mestrado Profissional em Gestão de Redes e Serviços
Pontifícia Universidade Católica de Campinas
Campinas, SP Brasil
davidb@puc-campinas.edu.br

***Resumo* – Nos últimos anos o desenvolvimento de aplicações para Cidades Inteligentes vem aumentando e se salienta na diretiva do governo na adoção de softwares livres. Nesse contexto se observa que a linguagem PHP é utilizada atualmente em 82,9% dos websites mundialmente. A gestão pública ainda dá seus primeiros passos na informatização de seus serviços e, quando existente, ainda é por meio de desenvolvimentos internos. Contudo já existe a iniciativa do Ministério do Planejamento, desde 2005, no desenvolvimento em parceria com a iniciativa privada e disponibilização do sistema e-cidade e seus fontes de forma gratuita para as prefeituras. Porém o aumento dos ataques cibernéticos e o aumento da sofisticação dos ataques elege a segurança da informação como um dos elementos fundamentais no sucesso do desenvolvimento, implantação e sustentação de websites em linguagem PHP. O presente artigo apresenta três vulnerabilidades mais exploradas em PHP com soluções simples para corrigi-las, efetua um teste experimental no website e-cidade, descrevendo os resultados obtidos com o uso da ferramenta *Netsparker Desktop* e, por fim, apresenta a importância da adoção do framework COBIT 5 para processos de Tecnologia da Informação, com foco nos processos DSS05 - Gerenciar Serviços de Segurança.**

***Palavras-Chave – Cidades Inteligentes; aplicações PHP; E-CIDADE; Vulnerabilidades; COBIT***

### I. INTRODUÇÃO

Nos últimos anos o desenvolvimento e implantação das Cidades Inteligentes [1] vem ao encontro de uma sociedade cada dia mais conectada e ansiosa pelas facilidades advindas dos avanços tecnológicos e comunicação (TICs) dos últimos anos. Indiscutivelmente essas tecnologias trouxeram mudanças profundas em vários aspectos do estilo de vida, social, econômica e política em toda a sociedade, onde surgiram novas demandas como por exemplo, crescimento exponencial do comércio eletrônico de produtos e serviços, rápida adoção de transações eletrônicas bancárias e financeiras, uso excessivo das redes sociais, aplicativos de mensagens instantâneas e muitos outras mudanças na forma e na maneira de como as pessoas estão interagindo e isto reflete diretamente em suas relações com empresas e setor público.

Estes avanços tecnológicos para o setor público trarão inúmeros benefícios, como a maior proximidade dos munícipes, conhecendo melhor suas demandas e anseios. Outro fator importante refere-se ao acesso às informações que, de uma forma mais rápida do que se consegue atualmente, se torna decisiva para uma melhor tomada de decisões. Aqui se observa ganhos efetivos para a aprovação de projetos de investimentos nas áreas mais importantes, tais como educação, saúde, segurança, transporte e mobilidade urbana, uso dos recursos naturais e meio ambiente.

Por outro lado a gestão pública, na maiorida das prefeituras do Brasil, esta dando seus primeiros passos em busca de um modelo de governança e eficiência. Tanto para seus processos internos quanto para seus sistemas, onde em sua maioria se tem sistemas desenvolvidos internamente e, em poucos casos, sistemas adquiridos de empresas terceiras e por processo de licitação. Desde 2005, o Ministério do Planejamento vem formentando o desenvolvimento de softwares livres em parceria com empresas privadas, dentre eles, o sistema E-CIDADE [2], que é um ERP (*Enterprise Resource Planning*) voltado para a gestão de um município, o qual é composto pelos módulos de Educação, Saúde, Financeiro, Patrimonial, Cidadão, Gestor, Recursos Humanos, BI (*Business Inteligence*) e Geoprocessamento. Outra iniciativa do Ministério da Saúde é a adoção do Sistema e-SUS Atenção Básica (e-SUS AB) [3] por todas as prefeituras, a partir de 2018, de forma garantirá o acesso as informações dos atendimentos realizados pelo SUS - Sistema Único de Saúde e desta forma priorizar ações, projetos e investimentos que visam a melhoria da qualidade do sistema de saúde público.

Evidencia-se a adoção de softwares livres, por todas as esferas do governo, e está se consolidando como um padrão, dentre eles a utilização do sistema operacional de servidores em Linux, de servidores WEB em Apache, bancos de dados como PostgreSQL e MySQL, linguagem PHP que atualmente é utilizada em 82,9% dos websites mundialmente [5] e outras ferramentas e tecnologias livres [4].

A crecentente necessidade da informatização dos municípios e a implantação de cidades inteligentes, irão

aumentar exponencialmente o desenvolvimento de soluções e aplicações voltadas para o serviço á população.

Porém a estes atuais e futuros desenvolvimentos deve se dar uma atenção especial á segurança em todas as fases, desde o desenvolvimento até a implantação e atualizações pois o crscente número de ataques cibernéticos, sofisticação e sofisticação desses ataques vem crescendo dia a dia.

A literatura da área nos aponta que nos desenvolvimentos em PHP as 3 vulnerabilidades mais exploradas por hackers são: *SQL Injection* [6][7], *Cross-Site Scripting* [8] e *Any File Inclusion* [9].

Pode-se observar na Tabela 1 uma lista do OWASP do ano de 2013 sobre as 10 maiores vulnerabilidades de segurança em aplicações WEB [10,15] , uma lista atualizada esta sendo preparada para o final de Novembro de 2017.

| **OWASP Top 10 - 2013** | | | |
|---|---|---|---|
| A1 | Inserção de Código | A6 | Exposição de Dados Sensíveis |
| A2 | Quebra de Autenticação e Gerenciamento de Sessão | A7 | Falta de Função para Controle do Nível de Acesso |
| A3 | Cross-Site Scripting (XSS) | A8 | Cross-Site Request Forgery (CSRF) |
| A4 | Referência Insegura e Direta a Objetos | A9 | Utilização de Componentes Vulneráveis Conhecidos |
| A5 | Consiguração Incorreta de Segurança | A10 | Redirecionamentos e Encaminhamentos Inválidos |

Tabela 1 – Lista Top 10 OWASP – 2013 – Fonte: Fundação OWASP [10]

Em adição á solução ou mitigação dos riscos inerentes ao desenvolvimento de websites em PHP recomenda-se a instalação em ambientes computacionais seguros, pois são de vital importancia para garantir a disponibilidade, confidencialidade e interidade das informações, portanto adotamos o uso do COBIT 5 Framework [11] na construção do ambiente.

Este artigo tem como contribuição:

- Demosntrar de forma clara os riscos e fontes de estudos pela comunidade científica para desenvolvimento de websites em liguagem de programação PHP.
- Aplicação da ferramenta Netsparker Desktop Web Application Security Scanner [12] para identificar vulnerabilidades na aplicação E-CIDADE versão 2017-1 em ambiente de experimental.
- Processos de segurança com base no framework COBIT 5.

## II – Principais vulnerabilidades exploradas em aplicações WEB - PHP

A - *SQL Injection* [7]: É um dos mais comuns tipos de ataques, onde consiste em adicionar caracteres ou comandos SQL em campos de formulários do website ou através de parametros passados por *URLs*, por exemplo na página de acesso.


Figura 1 – Exemplo de inclusão de SQL Injection – Fonte elaborado pelo autor

Na Fig. 1 é representado como um ataque de *SQL Injection* é basicamente realizado, que consite na inclusão de de sentenças válidas dentro de campos, como por exemplo os campos usuário e senha, estas sentenças permitem diblar a lógica existente no código da sentença SQL da aplicação, permitindo o acesso ao sistema e suas funcionalidades , na Fig. 2 temos um exemplo de código em PHP totalmente vulneravel a este tipo de ataque, onde é inserido códigos maliciosos, como por exemplo a expressão **'or 1=1;#,** no campo $str recebido dos dos campos GET[‘usuário’] e GET[‘senha’], resultando em uma senteça SQL válida “**select * from usuarios where usuario=’’ or 1=1;# senha=’’**” e permitindo o acesso ao sistema.

```
<?php
$user=$_GET['usuario'];
$pass=$_GET['senha'];
$sql="SELECT * FROM usuarios WHERE usuario ='".$user."'
     and senha='".$pass."'";
$result = mysql_query($sql);
?>
```

Figura 2 – Exemplo de código em PHP com risco de SQL Injection

Para a correção dessa vulnerabilidade basta que seja tratado os campos de entrada de forma a impedir o uso de caracteres especiais, por exemplo como na Fig. 3, onde incluimos duas funções, *preg_replace* e

```
<?php
$user=$_GET['usuario'];
$pass=$_GET['senha'];

// Tratamento dos dados de entrada
$user=preg_replace('/[^[:alnum:]_.-]',"",$user);
$pass=addslashes($pass)

$sql="SELECT * FROM usuarios WHERE usuario ='".$user."'
     and senha='".$pass."'";
$result = mysql_query($sql);
?>
```

Figura 3 – Exemplo tratamento campos evitando o SQL Injection

Com este pequeno tratamento no código fonte é possível evitar o sucesso no uso dessa vulnerabilidade.

B- *Cross-Site Scripting* [8]: Também conhecida por CSS ou XSS é uma vulnerabilidada bastante explorada onde permite inserir códigos *JavaScript, VBScript, JavaScript, ActiveX* e *Flash* maliciosos no navegador do usuários, parecendo parte do site e imperceptível para a vítima e permitindo o roubo de informações confidencias contigos em

cookies, realizar *phishing* e entre outras possibilidades.É classificado em 3 categorias: a) *Stored*, b) *Reflected* e c) *DOM based XSS*.

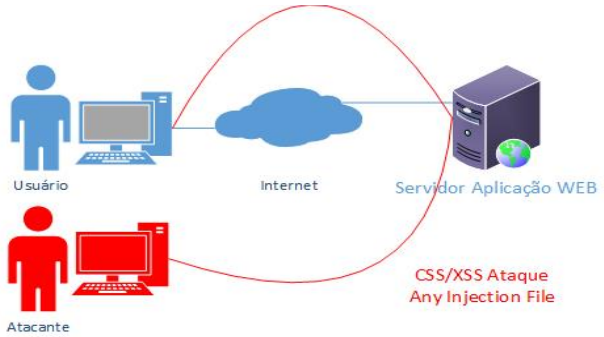


Figura 4 – Exemplo de *Cross-Site Scripting* (CSS/XSS) ataque Injection – Fonte elaborado pelo autor

Como demonstrado na Fig. 4 o ataque CSS/XSS ocorre do lado do usuário com a inclusão de um website com códigos maliciosos em *JavaScript* conforme exemplo na figura. 5.

```
// Site com passagem de parametros/valores
http://www.exemplo.com.br/?messangem=Você+Esta+Agora+No+Site
// É incluido outro site com códigos JavaScript Maliciosos
// e sendo executados pelo browse do usuário
http://www.exemplo.com.br/?mensagems=<script src=
"http://css.exemplo.com.br/malicioso-script.js"></script>
```

Figura 5 – Exemplo de inclusão de um código JavaScript por parametro

Nesta nova situação, para se corrigir é necessário usar funções para validar os dados de entrada, limitar o tamanho dos campos a serem inseridos, definir um conjunto de caracteres válidos para os campos de entrada. Em PHP existem funções que realizam a codificação automativamente tais como as funções: **htmlspecialchars()** e **htmlentities()**.

C - *Any File Inclusion* [14]: Vulnerabilidade muito comum onde permite que um atacante insira um arquivo em uma URL de website acessado pelo usuário ou em um servidor remoto, estes arquivos podem carregar e executar comandos maliciosos que permitem ter acesso a senhas de serviços ou de usuários, roubo de dados confidênciais, ataques DoS e entre outros tipos de ataque. Um simples exemplo de *Any File Inclusion* é demonstrado na figura 6.

```
1) Um exemplo de Any File Injection, considere o website onde
   o valor de pagina_exemplo é invisivel para o usuário

www.website_vitima.com.br/abc.php?proxima_pagina=pagina_exemplo

2) A vulnerabilidade em PHP consite no código:

$teste = $_REQUEST["proxima_pagina"];
Include($pagina_exemplo.".php");

3) O parametro 'proxima_pagina' direciona autimaticamente para
   outra página PHP
4) Um possível ataque pode ser da seguinte forma:
www.website_vitima.com.br/abc.php?proxima_pagina==
http://www.ataque_website.com.br/pagina_ataque

O arquivo pagina_ataque pode conter códigos maliciosos que serão
acessados e executados pela página 'abc.php'
```

Figura 6 - Exemplo de Any File Injection em PHP

Já neste caso para a correção dessa vulnerabilidade recomenda-se o tratamento dos parametros informados nas URLs, definindo os caracteres aceitos e o tamanho máximo de caracteres nos parametros.

## III – Teste Experimental

Para a realização do teste experimental foi realizado o download do sistema e-cidade pelo Portal Software Livre (https://softwarepublico.gov.br/social/e-cidade) versão 2017-1. Procedeu-se a sua instalação em um ambiente físico em Microsoft Windows 10 e ambiente virtual VMWare, realizado a instalação do website E-CIDADE e executado o teste de vulnerabilidade com o aplicativo Netsparker Desktop. Para o teste experimental foram utilizados os recursos: Notebook Dell Latitude E 7440, processador Intel I7, 256 HD SSD, 8GB RAM, sistema operacional Microsoft Windows 10 64 Bits, VMWare Workstation 12 Pro – Ambiente Virtual com 20GB Disco, 1GB RAM e 1 processador, Ubuntu Server 16.04.2 LTS, Apache 2, PHP 5.6 e PHP-FPM, Postgresql 9.5 e o software Netsparker Desktop Application Security Scanner 4.9.

Dentre os softwares de testes existentes atualmente no mercado foi eleito o NetSparker Desktop porque ser um dos softwares lideres em execução testes de segurança para códigos fontes WEB.

O teste experimental resultou nas vulnerabilidades apresentadas na Tabela 2 que foram apontadas com base na classificação do OWASP Top Ten 2013.

| Classificação OWASP | Qtde |
|---|---|
| A3 - Cross-Site Scripting (XSS) | 17 |
| A5 - Configuração Incorreta de Segurança | 1 |
| A6 - Exposição de Dados Sensíveis | 1 |
| A8 - Cross-Site Request Forgery (CSRF) | 1 |
| A9 - Utilização de Componentes Vulneráveis Conhecidos | 1 |

Tabela 2 – Resultado do teste de vulnerabilidades identificadas pelo software Netspaker – Fonte elaborada pelo autor

Outras 9 vulnerabilidades de baixo risco e informações também foram identificadas pela ferramenta, desta forma totalizando 30 vulnerabilidades, é fortemente recomendado o correto tratamento e correção das vulnerabilidades de alto e médio grau de risco (OWASP A3, A5 e A6).

O fato de se ter testado em um ambiente governamental, foi o de mostrar, pelo teste experimental, que a despeito dos esforços já realizados, ainda há que se fazer efetivos e constantes investimentos em segurança nos desenvolvimentos PHP de sistemas para o setor governamental.

Visando garantir um ambiente seguro, robusto e eficiente e desta forma atender os objetivos de todos os *stakeholders*, torna-se importante apresentar, ainda que de maneira sucinta, as contribuições do Framework COBIT 5.

## IV –Framework COBIT 5 [11]

O framework COBIT, da ISACA[13], é aplicável a gestão de toda a área de tecnologia, desde o nível estratégico, ao tático e operacional, avaliando o nível de maturidade de todos

os processos de TI, que visam garantir um ambiente seguro, robusto e eficiente e desta forma atender os objetivos de todos os *stakeholders*.

O COBIT 5 é a versão mais recente do framework possui 37 processos que são distribuidos em 5 domínios, sendo: I) Avaliar, dirigir e monitorar, II) Alinhar, planejar e organizar, III) Costruir, adquirir e Implementar, IV) Entregar suporte e serviço e V) Monitorar, o framework. O framework esta fundamentado em 5 princípios, que são: Satisfazer as expectativas dos stakeholders, separar governança de gestão, habilitar uma visão holística, ser um Framework integrador e cobrir o negócio como um todo. Para aferir e avaliar o nível de maturidade dos procesos são utilizados 6 níveis que foram definidos por: 0 - Processo Incompleto, 1 - Processo Executado, 2 - Processo Gerenciado, 3 - Processo estabelecido, 4 - Processo Previsível e 5 - Processo Otimizado.

Para um ambiente seguro de TI recomenda-se a implementação dos seguintes processos de segurança do framework apresentadas na Tabela 3.

| | |
|---|---|
| Avaliar, Dirigir e Monitorar | |
| EDM03 Assegurar a Otimização de Riscos | Assegura que a tolerância a riscos da organização sejam compreendidos, articulados e comunicados. |
| Alinhar, Planejar e Organizar | |
| APO12 Gerenciar os Riscos | Identificar, avaliar e reduzir os riscos relacionados a TI dentro dos níveis de tolerância estabelecidos pela diretoria executiva da organização. |
| APO13 Gerenciar a Segurança | Define, opera e monitora um sistema para a gestão de segurança da informação. |
| Construir, Adquirir e Implementar | |
| BAI06 Gerenciar Mudanças | Gerencia todas as mudanças de uma maneira controlada, incluindo mudanças de padrão e de manutenção de emergência relacionadas com os processos de negócio, aplicações e infraestrutura. |
| BAI07 - Gerenciar Aceite e Transição de Mudança | Aceita e autoriza formalmente novas soluções operacionais, isto inclui desde o planejamento de implementação do sistema, conversão de dados, testes de aceitação, comunicação, preparação de liberação, transferência para o ambiente de produção. |
| Entregar, Serviços e Suporte (DSS) | |
| DSS05 Gerenciar Serviços de Segurança | DS5 Garantir Segurança dos Sistemas |
| | DS11 Gerenciar Dados |
| | DS12 Gerenciar os Ambientes Físicos |
| | DS13 Gerenciar Operações |

Tabela 3 – Processos de Segurança Framework COBIT – Fonte elaborada pelo autor.

## V - Conclusão

Atualmente o aumento da exposição aos riscos cibernéticos e a sofisticação dos ataques vem aumentando dia a dia, principalmente em websites, portanto ter aplicações desenvolvidas de forma segura e em ambientes seguros.

O ambiente experimental aqui aplicado foi utilizado neste momento, apenas em caráter de se aplicar os conceitos aqui abordados mas que serão futuramente serão aplicados em uma ambiente real de TI em uma prefeitura e será assunto para futuras discussões, estudo e trabalho.

Conclui-se que na construção de sistemas deve-se pautar pelo uso não apenas de uma ferramenta ou metodologia mais sim, na utilização de forma conjunta e complementar de várias tecnologias de defesa, como ferramentas de avaliação dos códigos fontes e frameworks, para assegurar a integridade, confidencialidade e continuidade dos serviços disponibilisados através de sistemas web a população.

## Referências


[1] Elsa NEGRE, Camile ROSENTHAL-SABROUX, Mila GASCÓ, “A KNOWLEDGE-BASED CONCEPTUAL VISION OF THE SMART CITY”, 48th Hawaii International Conference on System Sciences, 2015, pp 2317-2325.

[2] Portal do Software Público Brasileiro, disponível no endereçco eletrônico: https://softwarepublico.gov.br/social/e-cidade consulta realizada em Outubro de 2017.

[3] Portal da Saúde, disponível no endereço eletrônico: http://dab.saude.gov.br/portaldab/esus.php, consulta realizada em Outubro de 2017.

[4] Levantamento do desenvolvimento de ferramentas e soluções em software livre, http://www.softwarelivre.gov.br/levantamento/levantamento/levantamento, consulta em Outubro de 2017.

[5] W3Techs - World Wide Web Technology Surveys, site que fornece informações do uso de várias tecnologias para WEB, https://w3techs.com/technologies/overview/programming_language/all, consulta realizada em Outubro de 2017.

[6] Vamshi Krishna Gudipati, Trinadh Venna, Soundarya Subburaj, Omar Abuzaghleh, “Advanced Automated SQL Injection Attacks and Defensive Mechanisms”, IEEE 2016.

[7] Voitovych O.P., Yuvkovetskyi O.S., Kupershtein L.M. , “SQL Injection Prevention System”, 2016 International Conference "Radio Electronics & InfoCommunications" (UkrMiCo), September 11-16 de 2016, Kiev, Ukraine

[8] Dr. Anil Kumar, Krishna Reddy, “Constructing Secure Web Applications With Proper Data Validations”, IEEE International Conference on Recent Advances and Innovations in Engeneering (ICCRAIE-2014), May 09-11, 2014, Jaipur, India

[9] Md Fazlul Haque, “Enhancement of Web Security Against External Attack”, European Scientific Journal May 2017 Edition Vol. 13, No. 15 ISSN: 1857 – 7881.

[10] A fundação OWASP é uma entidade aberta e entidade sem fins lucrativos que estuda e promove soluções para segurança de aplicações na internet. Periodicamente publica a lista das 10 maiores riscos de segurança em desenvolvimento de websites em todo o mundo, https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project#OWASP_Top_10_for_2013, consulta realizada em 10/10/2017.

[11] ISACA (2013). COBIT 5 for Risk. United States of America: ISACA

[12] https://www.netsparker.com/web-vulnerability-scanner/ , consultado em Outubro de 2017

[13] ISACA - Information Systems Audit and Control Association - Sociedade sem fins lucrativos que desenvolveu e promove o uso do Framework COBIT- http://www.isaca.org/COBIT/Pages/default.aspx

[14] Jingling Zhao, Rulin Gong, “A New Framework of Security Vulnerabilities Detection in PHP Web Application”, 2015 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing

[15] OWASP TESTING GUIDE V3 – 2008 Copyright 2002-2008 OWASP Foundation