

# **Best Practices for Secure Web Development**

**Revision 3.0**

**Revision Date**

## Revision History


## Acknowledgments

*Secure Programming for Linux and Unix HOWTO*

---

*The World Wide Web Security FAQ*

---

# Contents

---

<b>1</b>	<b>WHY? .....</b>	<b>1</b>
<b>1.1</b>	<b>Frequently Asked Questions.....</b>	<b>2</b>
<b>2</b>	<b>FUNDAMENTALS .....</b>	<b>4</b>
<b>2.1</b>	<b>Security as part of the business picture .....</b>	<b>4</b>
<b>2.2</b>	<b>Security as part of the requirements gathering.....</b>	<b>4</b>
<b>2.3</b>	<b>Security as part of the architecture.....</b>	<b>4</b>
<b>2.4</b>	<b>Watch what you use.....</b>	<b>5</b>
<b>2.5</b>	<b>Never trust incoming data. Never.....</b>	<b>5</b>
<b>2.6</b>	<b>When possible, help the user .....</b>	<b>6</b>
<b>2.7</b>	<b>Code reviews are your friends.....</b>	<b>6</b>
<b>2.8</b>	<b>Privacy &amp; law .....</b>	<b>6</b>
<b>2.9</b>	<b>Stay up-to-date!.....</b>	<b>7</b>
<b>2.10</b>	<b>Document, document, document!.....</b>	<b>7</b>
<b>3</b>	<b>TECH DETAILS .....</b>	<b>8</b>
<b>3.1</b>	<b>Don't be anonymous when you won't end up so.....</b>	<b>8</b>
<b>3.2</b>	<b>Do not use more power than you actually need.....</b>	<b>8</b>
<b>3.3</b>	<b>Don't use GET to send sensitive data! .....</b>	<b>9</b>
<b>3.4</b>	<b>Never trust incoming data – details .....</b>	<b>10</b>
<b>3.5</b>	<b>Don't rely on the client to keep important data .....</b>	<b>10</b>
<b>3.6</b>	<b>Don't store sensitive stuff in the *SP page itself.....</b>	<b>11</b>
<b>3.7</b>	<b>Beware of extensions.....</b>	<b>11</b>
<b>3.8</b>	<b>Keep an eye on HTML Comments left in production code .....</b>	<b>12</b>
<b>3.9</b>	<b>Error messages .....</b>	<b>12</b>
<b>3.10</b>	<b>Cross-site scripting .....</b>	<b>12</b>
<b>3.11</b>	<b>Check the wizard-generated or sample code.....</b>	<b>16</b>
<b>3.12</b>	<b>Language &amp; technology specifics .....</b>	<b>16</b>
<b>3.13</b>	<b>Middleware security .....</b>	<b>19</b>
<b>3.14</b>	<b>Declarative vs programmatic.....</b>	<b>20</b>
<b>3.15</b>	<b>Distributed systems and firewalls .....</b>	<b>21</b>
	<b>PKI is not a silver bullet.....</b>	<b>23</b>
<b>3.17</b>	<b>Snake oil .....</b>	<b>23</b>
<b>3.18</b>	<b>When randomness matters.....</b>	<b>23</b>
<b>3.19</b>	<b>Use the logs. Create useful logs. ....</b>	<b>24</b>
<b>3.20</b>	<b>SSL .....</b>	<b>24</b>
<b>3.21</b>	<b>Other pointers .....</b>	<b>25</b>

**Legal Notice.**

**About the author:**

---

# **1 WHY?**

---

**Target Audience**

## What exactly do you mean by 'information security'?

*protect*

*technologies*

*processes  
enable*

- *authentication*
- *authorization*
- *privacy*
- *accountability & non-repudiation*
  
- *integrity*
- *detection & monitoring*
- *legal aspects*

**I thought the firewall would take care of this. Or file permissions. Or SSL.**

*around*

*in*

**I'm an experienced web developer and don't think I need this.**

*secure*

**Can't someone do this after I finish my dev work?**

**Note**

## 2 FUNDAMENTALS

---

*managing risk*

*assets*

*risks*

- *assets*
- *use cases*
- *the users, their roles and rights*
- *legal and business issues*



- 

- 

- 

*reported*

*of private data*

*collection and handling*

---

*Crypto Law Survey*

---



## 3 TECH DETAILS

---

...





- 

- 

---

**relying on  
hidden form fields to maintain sensitive data between requests**

**Form Tampering Vulnerabilities in Several  
Web-Based Shopping Cart Applications**

---

---

*source*

**::\$DATA**

*Translate: f*

.

---

---

---

**.inc**

- *physical paths*

*c:\inetpub\wwwroot\common.asp*

- *platform architecture.*

*client*

*Send detailed ASP error message to*



*not*

```
<html>
<%
    if request.form ("yourname") <>" then
        Response.Write("Hello " + request.form ("yourname"))
    else
%>
    <form method="POST">
        <input type="text" name= yourname>
        <input type="submit" value="submit">
    </form>
<%
    end if
%>
</html>
```

```
<script language='javascript' >alert ('gotcha!');</script>
```

*the visitor will get the script as if it were part of the legitimate site and*

*the script will get executed on the browser.*

*cannot be identified by the browser*

*within an HTML tag*

```
<a href=" [event]='bad script here' "> click me </a>
```

```
<% Response.Write("<BODY BGCOLOR=\"\" +  
Request.Cookies("UserColor") + "\">"); %>
```

```
Cookie: %22+onload%3D%27window%2Elocation%3D  
%22http%3A%2F%2Fwww%2Eevilsite%2Ecom%22%3B%27
```

```
<body BGCOLOR="" onload=  
'window.location="http://www.evilsite.com";' ">
```

*What to do?*

*displayed*

**&gt;**

**&lt;**

Server

HTMLEncode

**<script>**

**&lt;script&gt;**

see



Server.URLEncode

---

---

---

---

---

---

---

---

**buffer overflows**

---

---

**attacks**

**format string**

---

---

---

---

*relying*

*correctly.*

*Java Authentication and Authorization*

*Service*

---

*Java Security*

---

*Inside Java 2 Platform Security*

*Security Code Guidelines*

---

*Secure*

*Programming for Linux and Unix HOWTO*

`-Djava.security.debug`

`java -Djava.security.debug=help`

---

*World Wide Web Security FAQ*

---

---

---

*whisker*

---

*CGI/Perl Taint Mode FAQ"*

---

---

---

---

*Secure Programming for Linux and Unix HOWTO* 

---

---

*Secure UNIX Programming FAQ* 

---

*Writing Safe Setuid Programs* \_\_\_\_\_

*How to find security holes* \_\_\_\_\_

*How to Write Secure Code* \_\_\_\_\_

*xmlsig*

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

*authentication authorization impersonation delegation of credentials*

*the*

\_\_\_\_\_

*Programming Windows Security*

\_\_\_\_\_

\_\_\_\_\_

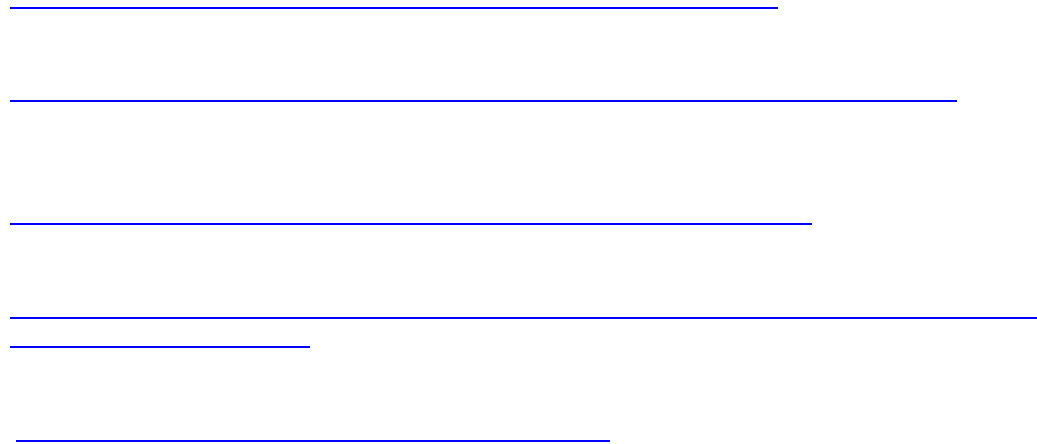
\_\_\_\_\_

\_\_\_\_\_

*Designing Enterprise Applications with J2EE*

---

*Building Java Enterprise Systems with J2EE*



*Declarative*

*Programmatic*



*IsCallerInRole*

*and*

*COM Internet Services      DCOM over HTTP)*

*RMI ,Servlets and Object Serialization FAQ*

---

*Joint Firewall Revised Submission*

---

*CORBA Firewalls*

---

SOAPMethodName

text/xml      text/xml-SOAP

M-POST

*Simple Object Access Protocol (SOAP) and Firewalls*

---

---

---

*Public-Key Infrastructure*

*Understanding the*

*Ten Risks of PKI*



*Cryptogram*



*pseudo*      *appear*

*SecureRandom*

---

*not*

...

...

...

---

---