Defaced!

<table>
<thead>
<tr>
<th>Mirror saved on:</th>
<th>2010-05-11 10:45:22</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notified by:</td>
<td>AhliSyurgaCrew</td>
</tr>
<tr>
<td>Domain:</td>
<td><a href="http://www.ppdhuluperak.gov.my">http://www.ppdhuluperak.gov.my</a></td>
</tr>
<tr>
<td>System:</td>
<td>Linux</td>
</tr>
<tr>
<td>Web server:</td>
<td>Apache</td>
</tr>
<tr>
<td>IP address:</td>
<td>119.110.111.23</td>
</tr>
</tbody>
</table>

Sorry PPD, Silap Deface :P

- AhliSyurgaCrew
... and the next day...
phpnuke.org has been compromised

Posted: 07 May 2010 07:25 AM

Websense® Security Labs™ ThreatSeeker™ Network has discovered that the popular Web site, phpnuke.org, has been compromised.

PHP-Nuke is a popular Web content management system (CMS), based on PHP and a database such as MySQL, PostgreSQL, Sybase, or Adabas. Earlier versions were open source and free software protected by GNU Public License, but since then it has become commercial software. As it is still very popular in the Internet community, it is not surprising that it has become a target of blackhat attacks.
Eleonore Exploit Pack.

Eleonore exploits pack license version 1.3.2

Fast statistic:
Traffic: 44838 / Loads: 3562 / Percent: 7.94%
What I cannot create, I do not understand.”

- Richard Feynman
Overview

“Layers of security.”
--Unknown

“Complexity is the worst enemy of security.”
--Bruce Schneier
ManageEngine Firewall Analyzer 5

- CSRF
  - Execute SQL queries (Not injection)
  - Create a new administrative account

- XSS
  - The results from a sql query
  - SELECT “<script>alert(/xss/)</script>”
Profense Web Application Firewall

- "Defenses against all OWASP Top Ten vulnerabilities"
- CSRF - CVE-2009-0468
  - Proxy for MITM
  - Configuration changes
  - Shutdown the machine (DoS)
- Reflective XSS - CVE-2009-0467
The PHP-Nuke Exploit

(Nov 2004)

- PHP-Nuke 7.0
  - SQL Injection
    - Get admin
  - Broken admin
    - Enable phpBB
  - Filter Bypass
  - Eval()

(Aug 2009)

- PHP-Nuke 8.1.35
  - (User account Required!)
  - SQL Injection
    - Get admin
  - Broken admin
  - Path disclosure
  - Filter Bypass
  - Another sql injection
  - Local File Include
The PHP-Nuke Exploit

- (OWASP A1) SQL Injection
- (OWASP A3) Broken Authentication
- (CWE-200) Information Exposure
- (CWE-436) Filter Bypass
- (OWASP A1) SQL Injection (Again!)
- (CWE-98) Local File Include (LFI)
The PHP-Nuke Exploit

- (OWASP A1) SQL Injection
- (OWASP A3) Broken Authentication
- (CWE-200) Information Exposure
- (CWE-436) Filter Bypass
- (OWASP A1) SQL Injection (Again!)
- (CWE-98) Local File Include (LFI)
DEMO!
Fruit Analogy
- OWASP A1: Injection

<table>
<thead>
<tr>
<th>Attack Vectors</th>
<th>Security Weakness</th>
<th>Technical Impacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploitability</td>
<td>Prevalence</td>
<td>Detectability</td>
</tr>
<tr>
<td>EASY</td>
<td>COMMON</td>
<td>AVERAGE</td>
</tr>
</tbody>
</table>

- SQL Injection in the Journal module to get administrative credentials
PHP-Nuke Exploit

- OWASP A3: Broken Authentication and Session Management

<table>
<thead>
<tr>
<th>Attack Vectors</th>
<th>Security Weakness</th>
<th>Technical Impacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploitability</td>
<td>Prevalence</td>
<td>Detectability</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>COMMON</td>
<td>AVERAGE</td>
</tr>
</tbody>
</table>

• Shortcut to admin privileges
PHP-Nuke Exploit

- **PHP-Nuke Login:**

```php
$sql = "SELECT pwd FROM ".prefix."_authors 
    WHERE aid='$aid';
$result = $db->sql_query($sql);
$pass = $db->sql_fetchrow($result);
$db->sql_freeresult($result);
if ($pass[0] == $pwd && !empty($pass[0])) {
    return $adminSave = 1;
}
```

- **Secure Login:**

```php
$adodb->Execute("SELECT * from ".prefix."_authors 
    WHERE aid=?
    and pwd= ?", array($aid,sha256($salt.$pwd)));
```
Overview

- PoC Exploit:

```php
<?php
$admin_cookie=base64_encode($_GET['aid']."":"$_GET[pwd']");
print("javascript:document.cookie='admin='.$admin_cookie."'");
?>
```
PHPSecInfo Rocks!

Security Information About PHP

Session

<table>
<thead>
<tr>
<th>Test</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>save_path</td>
<td>Notice: save_path is disabled, or is set to a common world-writable directory. This typically allows other users on this server to access session files. You should set save_path to a non-world-readable directory</td>
</tr>
<tr>
<td></td>
<td><strong>Current Value:</strong> /var/lib/php4 (1733)</td>
</tr>
<tr>
<td></td>
<td><strong>Recommended Value:</strong> A non-world readable/writable directory</td>
</tr>
<tr>
<td>use_trans_sid</td>
<td>Notice: use_trans_sid is enabled. This makes session hijacking easier. Consider disabling this feature</td>
</tr>
<tr>
<td></td>
<td><strong>Current Value:</strong> 1</td>
</tr>
<tr>
<td></td>
<td><strong>Recommended Value:</strong> 0</td>
</tr>
</tbody>
</table>
Demo!

- Why 2 sql injection exploits?
(CWE-436) Filter Bypass

- '%20union%20'
- '*/union/*'
- ' union '

• (CWE-436) Filter Bypass

- '%20union%20'
- '*/union/*'
- ' union '

```sql
UNION/**/ SELECT "text" into outfile("/tmp/test.txt")
```
PHPMyAdmin CSRF+SQLi=RCE (CVE-2008-5621)

```php
$allow_list = array(
    /* needed for direct access, see FAQ 1.34 */
    /* also, server needed for cookie login screen (multi-server) */
    'server', 'db', 'table', 'target',
    /* Session ID */
    'phpMyAdmin',
    /* Cookie preferences */
    'pma_lang', 'pma_charset', 'pma_collation_connection',
    /* Possible login form */
    'pma_servername', 'pma_username', 'pma_password',
    /* rajk - for playing blobstreamable media */
    'media_type', 'custom_type', 'bs_reference',
    /* rajk - for changing BLOB repository file MIME type */
    'bs_db', 'bs_table', 'bs_ref', 'bs_new_mime_type'
);

<html>
    <img src="http://10.1.1.10/phpmyadmin/tbl_structure.php?db=information_schema&table=TABLES%60+where+0+union+select+char%2860%27+from"/>
</html>
```
PHP-Nuke Exploit

- PHP Local File Include->Remote Code Execution.

- AppArmor will not allow MySQL write to /var/www/ (Even if its chomd 777!)
PHP-Nuke Exploit

- SQL Injection To create a file in /tmp/
- AppArmor allows this.
- Local File Include to execute the file in /tmp/
SELinux doesn't allow this!

<table>
<thead>
<tr>
<th>Date</th>
<th>Host</th>
<th>Count</th>
<th>Category</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fri 02 Jul 2010</td>
<td>fedora</td>
<td>1</td>
<td>File Label</td>
<td>SELinux is preventing the httpd from using potentially mislabeled files (/tmp/backdoor.php).</td>
</tr>
<tr>
<td>Tue 30 Mar 2010</td>
<td>fedora</td>
<td>2</td>
<td>&lt;Unknown&gt;</td>
<td>SELinux is preventing mysql to access potentially mislabeled file(s) (/tmp/backdoor.php).</td>
</tr>
<tr>
<td>Tue 30 Mar 2010</td>
<td>fedora</td>
<td>3</td>
<td>&lt;Unknown&gt;</td>
<td>SELinux is preventing rm (rm) to access potentially mislabeled file(s) (/tmp/backdoor.php).</td>
</tr>
</tbody>
</table>

**Summary**

SELinux is preventing the httpd from using potentially mislabeled files (/tmp/backdoor.php).

**Detailed Description**

SELinux has denied httpd access to potentially mislabeled file(s) (/tmp/backdoor.php). This means that SELinux will not allow httpd to use these files. It is common for users to edit files in their home directory or tmp directories and then move (mv) them to system directories. The problem is that the files end up with the wrong file context which confined applications are not allowed to access.

**Allowing Access**

If you want httpd to access these files, you need to relabel them using restorecon -v '/tmp/backdoor.php'. You might want to relabel the entire directory using restorecon -R -v '/tmp'.

**Additional Information**

- **Source Context:** unconfined_u:system_r:httpd_t:s0
- **Target Context:** unconfined_u:object_r:mysqld_tmp_t:s0
- **Target Objects:** /tmp/backdoor.php [ file ]
- **Source:** httpd
- **Source Path:** /usr/sbin/httpd
PHP-Nuke Exploit

- Eval() and preg_replace /e
  - SELinux does not stop this.

```php
<string1>="phpinfo()";
$string2=preg_replace('/\e', $string1, ");
```

Study in Scarlet
PHP-Nuke Exploit

END