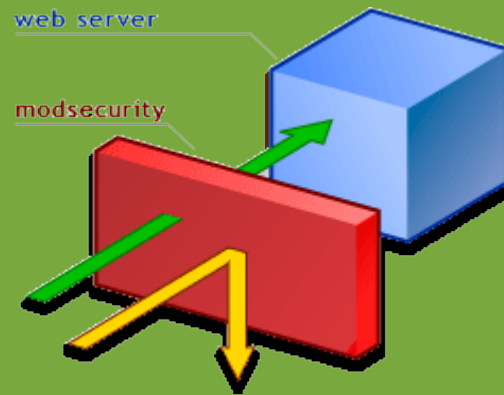


Advanced Web Application Defense with ModSecurity



Daniel Fernández Bleda
&
Christian Martorella



Tempde (The Netherlands)
July 28-31, 2005

What The Hack!

Who we are? (I)

Christian Martorella:

+6 years experience on the security field, mostly doing audits and Pen-testing.

Open Information System Security Group,
Barcelona chapter President.

First Improvised Security Testing Conference
Organizer

Security Forest, ExploitTree maintainer.

CISSP, OPST



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Who we are? (II)

Daniel Fernández Bleda:

Security Engineer +5 years experience on the field.

Institute for Security and Open Methodologies (ISECOM), Member, and OSSTMM promoter.

Open Information System Security Group, Barcelona chapter Member.

CISSP, CISA, OPST/OPSA Trainer



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Aim of the Presentation

Security of web applications

Introduce Modsecurity

Show how can you protect your web applications

Show our work in extending the features of this module.



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Actual Scenario (I)

Web Applications are insecure

Today are the most vulnerable part of company infrastructure.

Everyday the attacks at the application layer are growing:

SQL Injection

XSS

Command Injection

Buffer Overflows

Session manipulation

Etc..



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Actual Scenario (II)

The quantity of vulnerabilities is growing
80% of the last 30 exploits posted to Milw0rm target Web Applications: Wordpress, Phpbb, Xml-rpc, etc.

We can find a lot of Firewalls to analyse and filter traffic at the network level, but at the Application level, what are our options? Open and Free, very little..

There are secure networks with insecure applications that jeopardize all security of the company.



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

What are the Problems? (I)

The web development is chaotic

Lack of web security awareness.

First users requirement, last security (if time allow)

If it works don't touch it (sad but true)

False sense of security:

We have a firewall, we are safe

We use SSL, we are safe



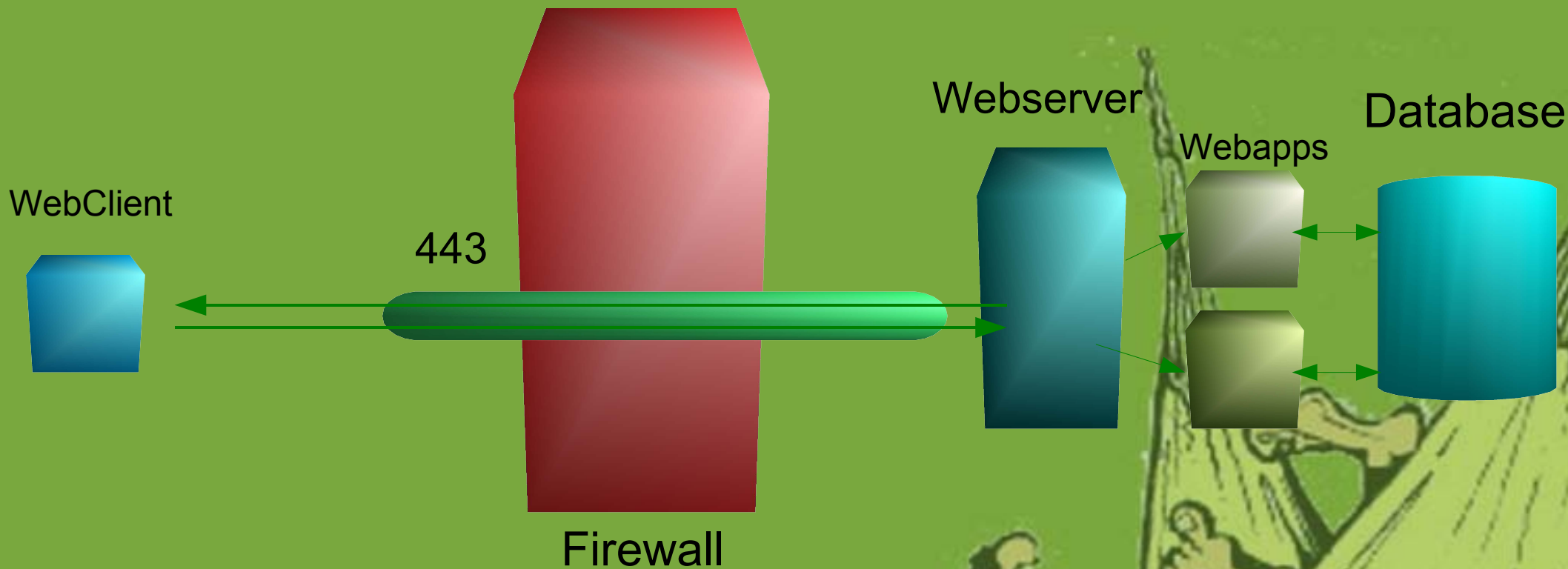
Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

What are the Problems? (II)

Network firewalls doesn't do anything at this level:



What The Hack!

Tempde (The Netherlands)
July 28-31, 2005

What is ModSecurity?

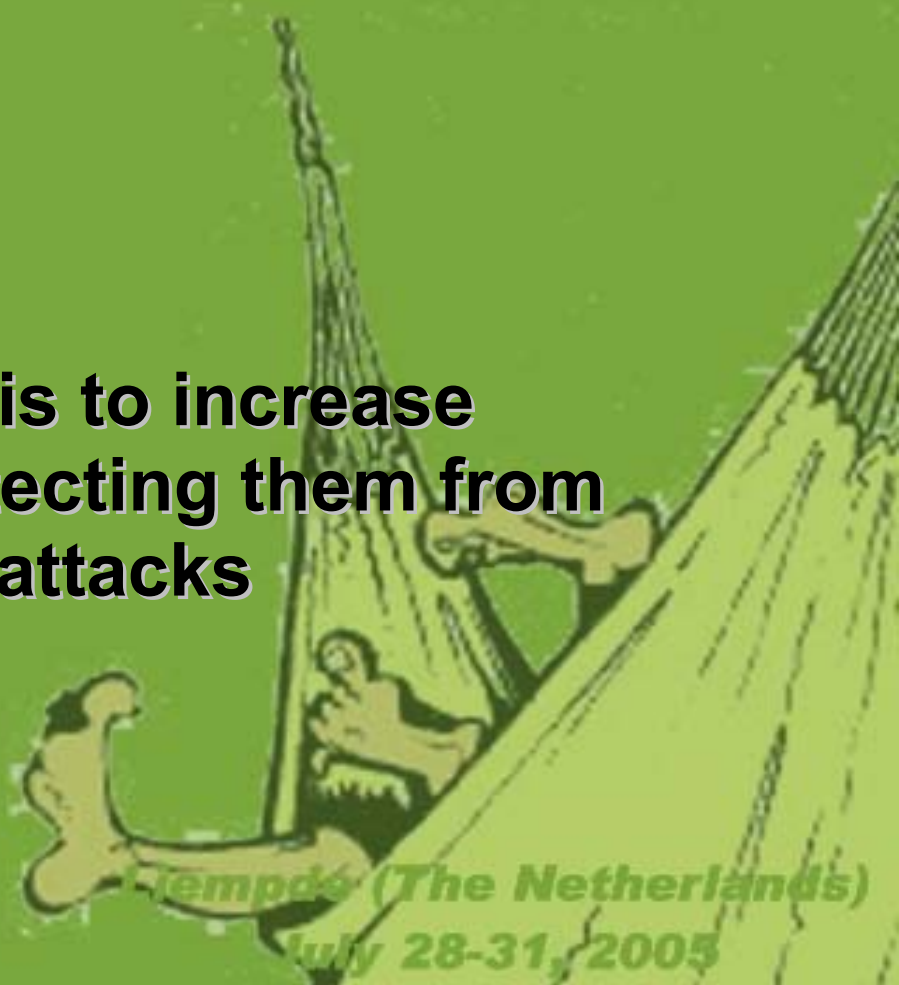
Intrusion Detection / Prevention for Web Applications

Operate as an Apache Module.

Open Source and GPL

Development by Ivan Ristic

The purpose of ModSecurity is to increase web application security by protecting them from known and unknown attacks



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

When can we use it?

Applications you can't modify: legacy applications, protected code like Zend encoder, Phpshield, etc.

New vulnerability discovered, temporal protection until patch is released.

Intrusion Detection .

Extra layer of security.

To protect Web services.

Web applications operated by people other than original software developers.



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Features (I)

Request filtering: incoming requests are analysed as they come in, and before they get handled by the web server or other modules.

Output Filtering: It could analyse the server response.
(Error, Critical data, Ex. PHP Errors)

Understanding of the HTTP protocol: since the engine understands HTTP, it performs very specific and fine granulated filtering.



Features (II)

Anti-evasion techniques: paths and parameters are normalized before analysis takes place in order to fight evasion techniques.

Remove multiple forward slash characters

Treat backslash and forward slash characters equally

Remove directory self-references

Detect and remove null-bytes (%00)

Decode URL encoded characters



Tempde (The Netherlands)

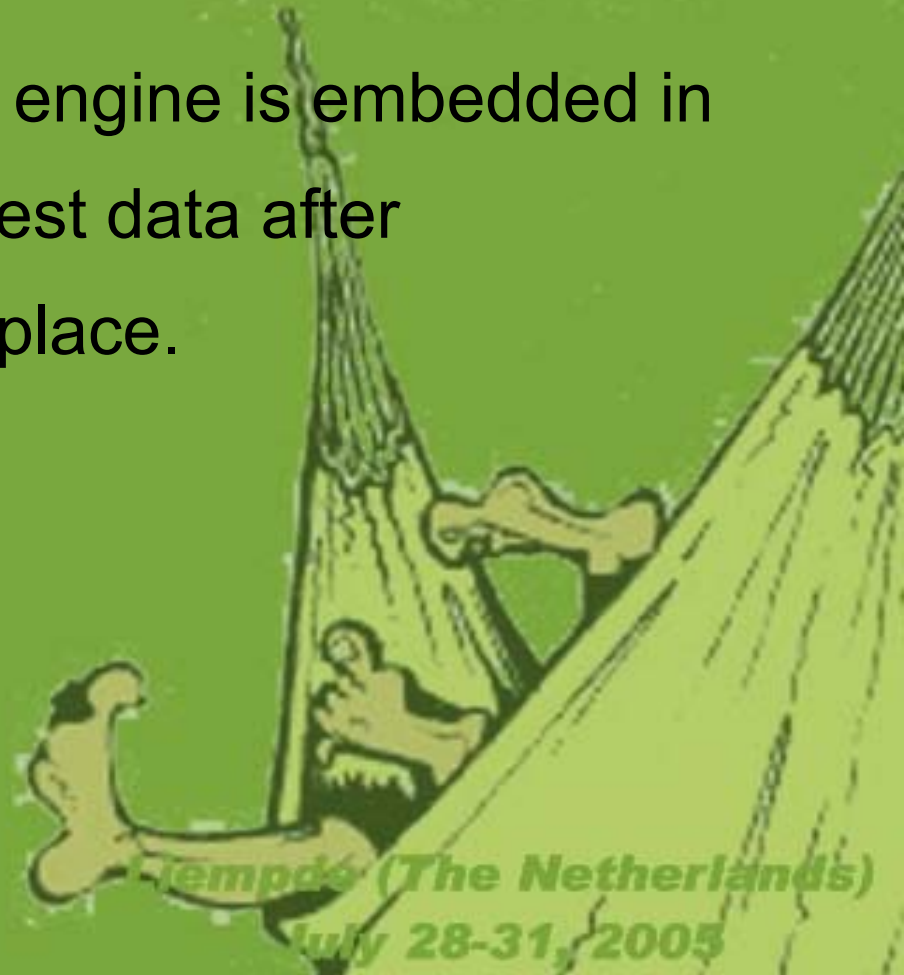
July 28-31, 2005

What The Hack!

Features (III)

POST payload analysis: the engine will intercept the contents transmitted using the POST method.

HTTPS and Compression: since the engine is embedded in the web server, it gets access to request data after decryption and decompression takes place.



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Features (IV)

Audit logging: full details of every request (including POST) can be logged for later analysis.

For doing an effective forensic investigation in a web attack, we need at least:

- Source IP
- Time stamp
- HTTP method
- URI requested
- Full Http Data



*Tempde (The Netherlands)
July 28-31, 2005*

What The Hack!

Special built-in checks

URL encoding validation

Unicode encoding validation

Byte range verification [0-255] detect and reject :

- Shellcode
- Cross Request Form Forgerie (CR,LF)

If the language used in our application is English we can limit the byte range to [32-126].



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Rules

What is a rule?

Rules are formed using regular expressions

Any number of custom rules supported

Also negated rules supported

Analyses:

Headers

Environment variables

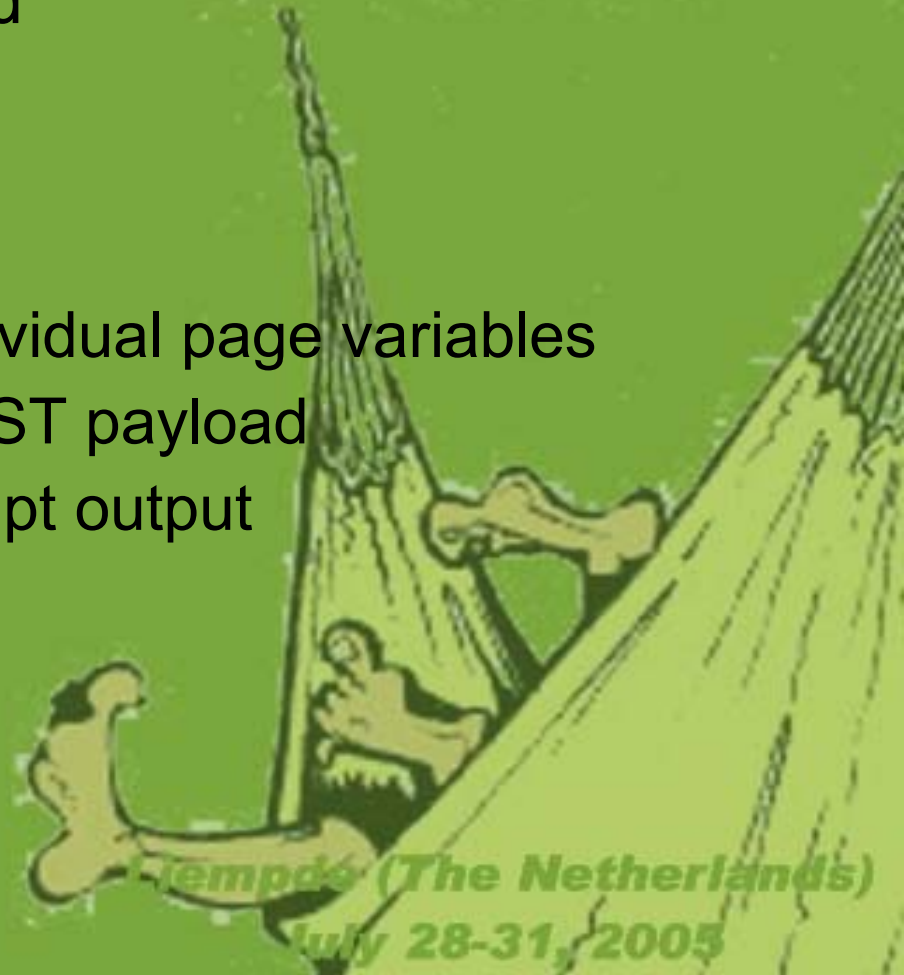
Server variables

Individual cookies

Individual page variables

POST payload

Script output



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Actions (I)

What are the actions?

Reject request with status code [403,500, ..]

Reject request with redirection

Execute external binary on rule match

Log request



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Actions (II)

Stop rule processing and let the request through

Rule chaining

Skip next n rules on match

Pauses for a number of milliseconds



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

File Upload

Intercept files being uploaded through the web server

Store uploaded files on disk

Execute an external script to approve or reject files (ClamAV anti-virus)



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Other

Change the identity of the web server: We can change the Server header to whatever we want, also will change the version in all server messages, like error, forbidden, etc.
Easy to use internal chroot functionality.



Tempde (The Netherlands)
July 28-31, 2005

What The Hack!

How does it works

1. Parse the request.
2. Perform canonicalization and anti-evasion actions.
3. Perform special built-in checks.
4. Execute input rules:

If the request is allowed, then it will reach the handler where it executes.

After the request:

1. Execute output rules.
2. Log the request.



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Security models

A security is the posture we take at the time of setting rules for our protection systems.

There are two security models:

Positive Model

Negative Model



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Security models: Positive Model

We allow what we know is right (safe).

Like network firewall model “Deny All - Allow what we need”.

Pros:

Better performance

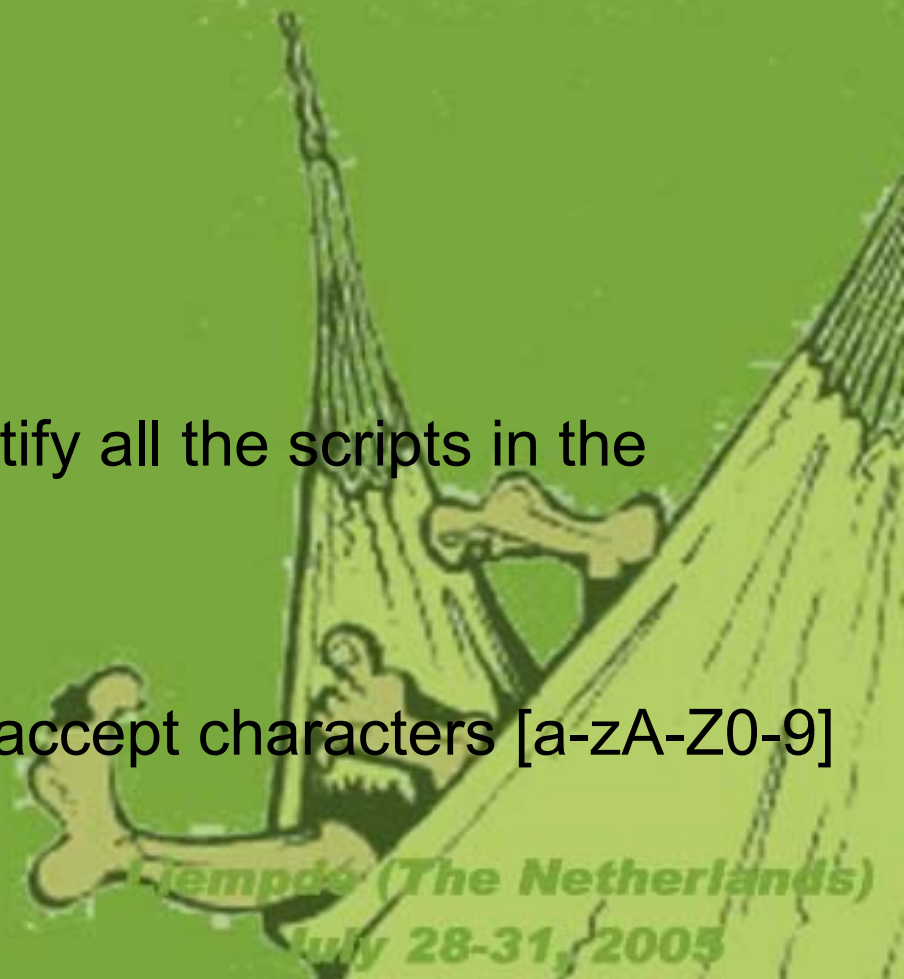
Less false positives

Cons:

More time to implement, we need to identify all the scripts in the application, and create rules for them.

Example:

Page log.asp, the field Login could only accept characters [a-zA-Z0-9] and could be 12 char long.



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Security Models: Negative Model (I)

Deny what is dangerous

Do we know everything that is bad for our applications?

Pros:

Less time to implement, we create a general that affect the whole application.

Cons:

More false positives

More processing time

Example XSS:

There are a lot of tags and places to look for XSS, we can miss some of them leaving a hole in our application.



Tempde (The Netherlands)

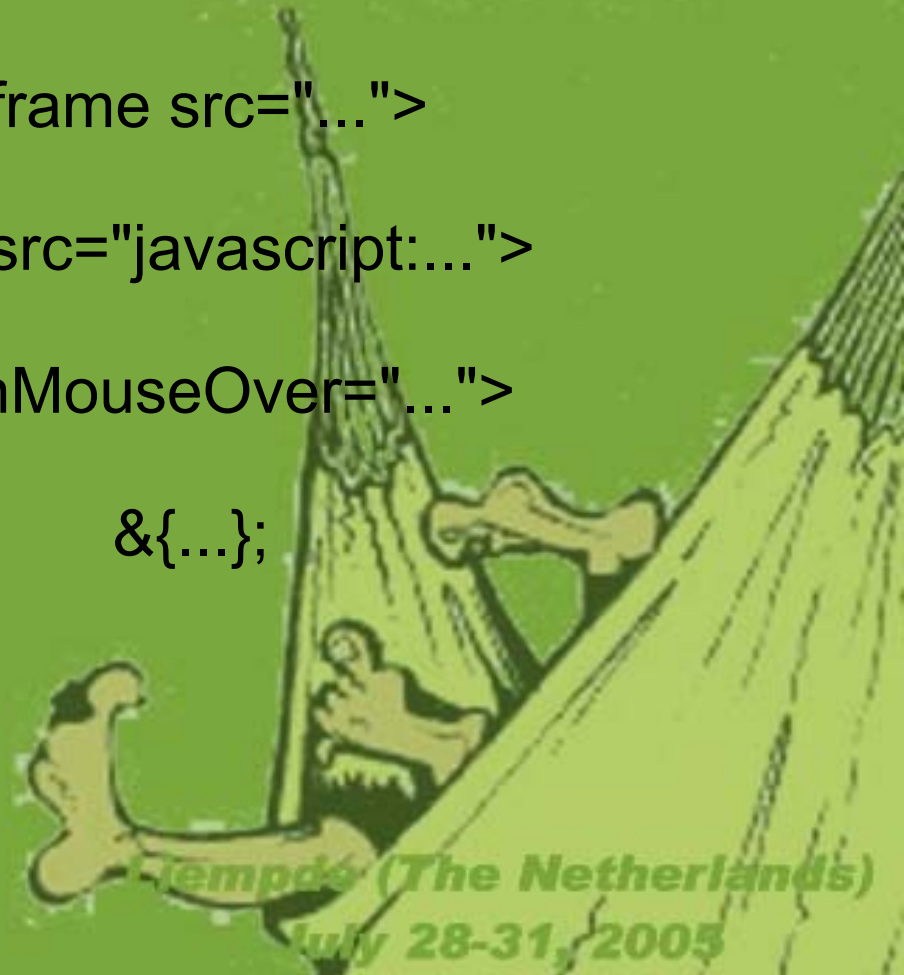
July 28-31, 2005

What The Hack!

Security Models: Negative Model (II)

```
<object>...</object>  
<embed>...</embed>  
<applet>...</applet>  
<script>...</script>  
<script src="..."></script>
```

```
<iframe src="...">  
  
<b onMouseOver="...">  
&{...};
```



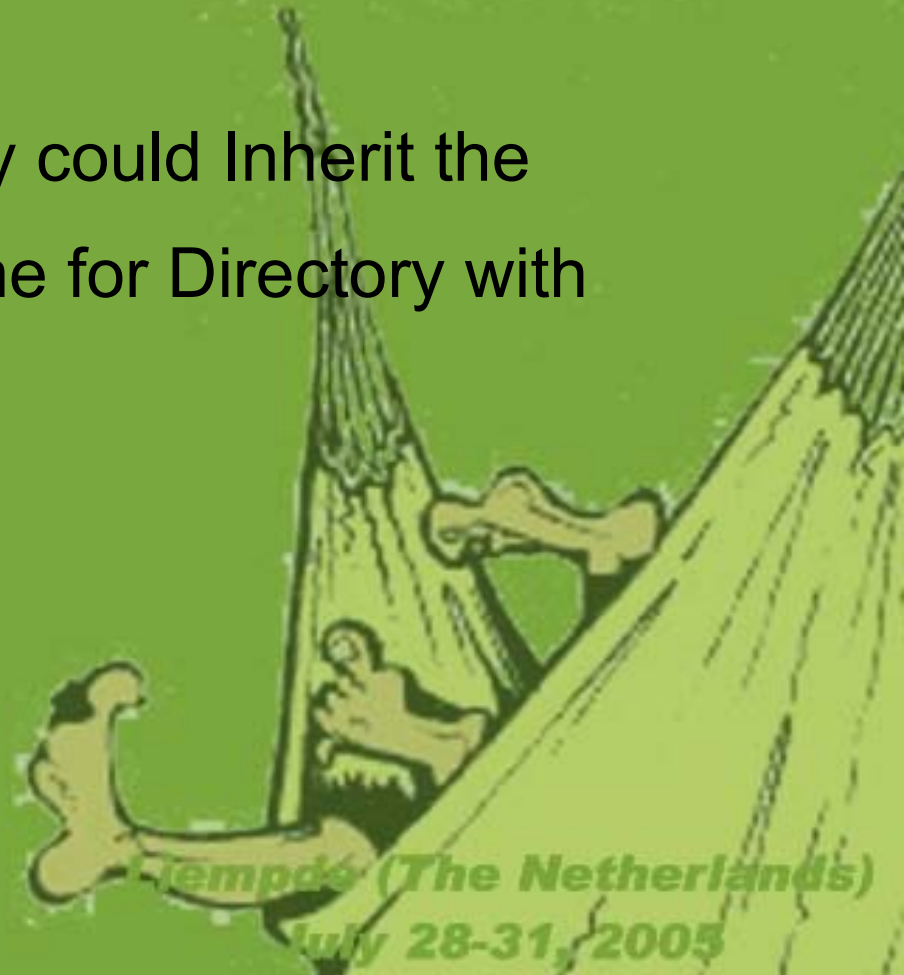
Tempde (The Netherlands)
July 28-31, 2005

What The Hack!

Configuration

Per Context configuration:

Inheritance: Virtual Host and Directory could Inherit the configuration of the Main Server. Same for Directory with Virtual Host.



*Tempde (The Netherlands)
July 28-31, 2005*

What The Hack!

Thoughts...

What about my web server, i don't use Apache...

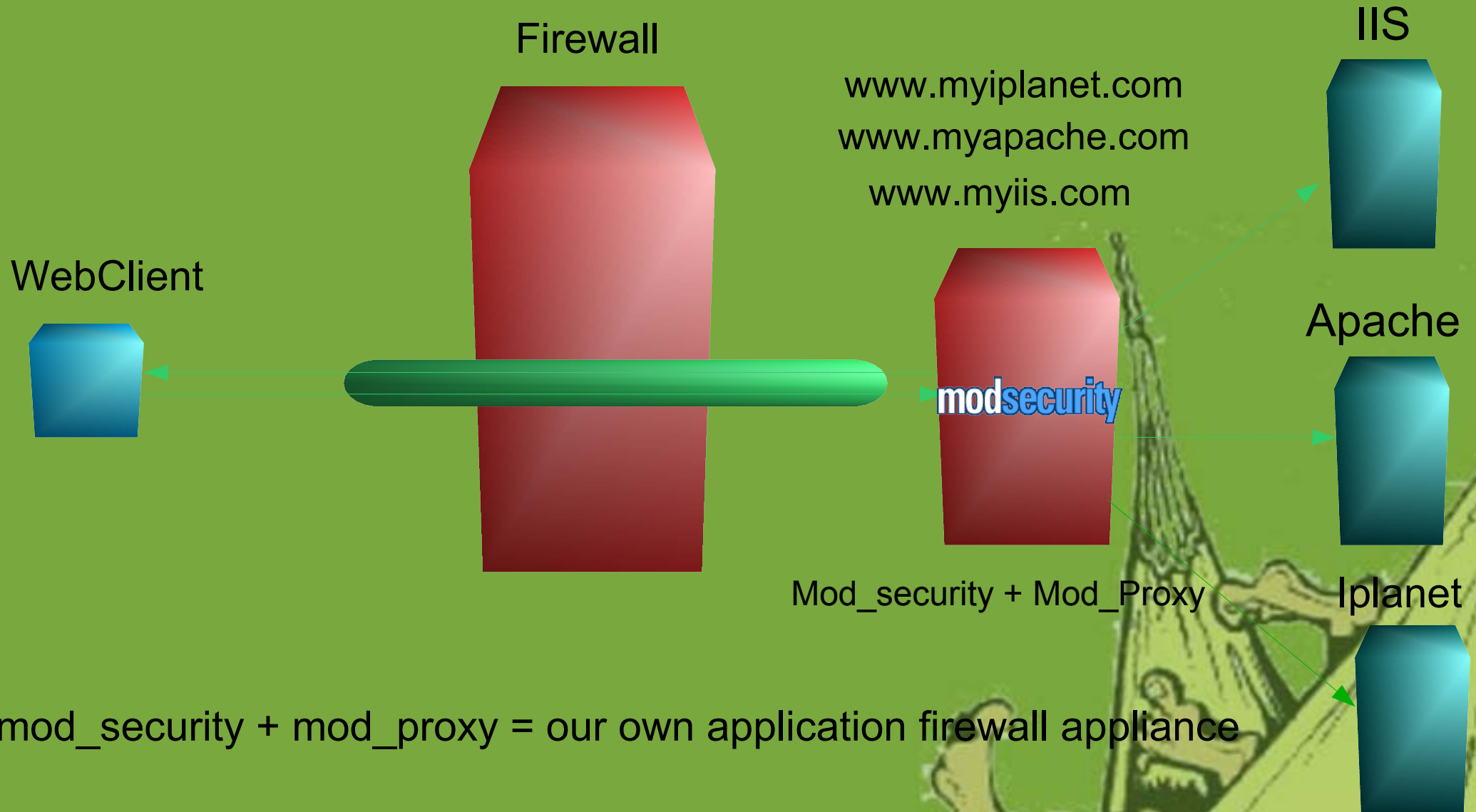


Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Reverse Proxy Model (I)



mod_security + mod_proxy = our own application firewall appliance

What The Hack!

Temple (The Netherlands)

July 28-31, 2005

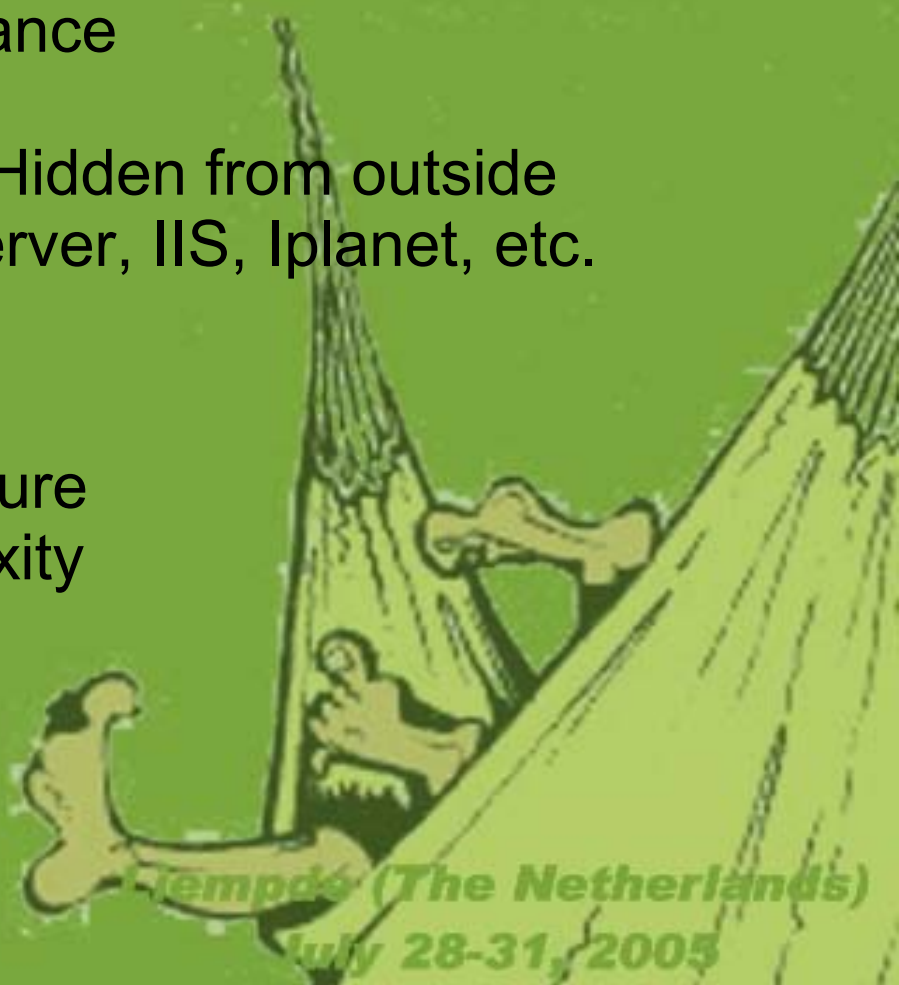
Reverse Proxy Model (II)

Advantages:

- ▶ Single point of access
- ▶ Increased Performance
- ▶ Network Isolation
- ▶ Network Topology Hidden from outside
- ▶ Protect any Web server, IIS, Iplanet, etc.

Disadvantages:

- ▶ Single point of Failure
- ▶ Increased Complexity



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Extending Features

Strip Comments

Cookie crypto-signing

Link crypto-signing

Hidden field signing

Web based logs console

Some of these features are present in commercial firewalls, so we thought it will be great if an Open Source project like Modsecurity could do the same.



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Strip Comments (I)

Using the Libxml2 it allow us to build an HTML tree of the parsed code sended to the user.

Having that, we could walk the tree looking for comments

```
<!-- xxxxxx ->
```

and cut them off.

So just adding the directive:

```
SecStripCommentCode On/Off
```



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Strip Comments (II)

Modsecurity will clean all commented code before sending the requested page to the user.

We considered the common situation of commented script code to allow backward compatibility with old browsers. This comments are not stripped.

We are working in the deletion of comments inside the code of different script languages (javascript, tcl, etc).



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Cookie Signing

Another feature is the crypto-signing of cookies, to prevent tampering (cookie poisoning, session fixation)

With directives:

SecSignCookies On/Off

SecEncryptionPassword “password”

We are using Cryptlib and do the crypto-sign with the Advanced Encryption Standard (AES) algorithm.

Still working on it



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Link Signing (I)

As an extra layer of security we have added the option to sign all links browsables from “Entry points”, so users are able only to follow the “intended flow” of the application.

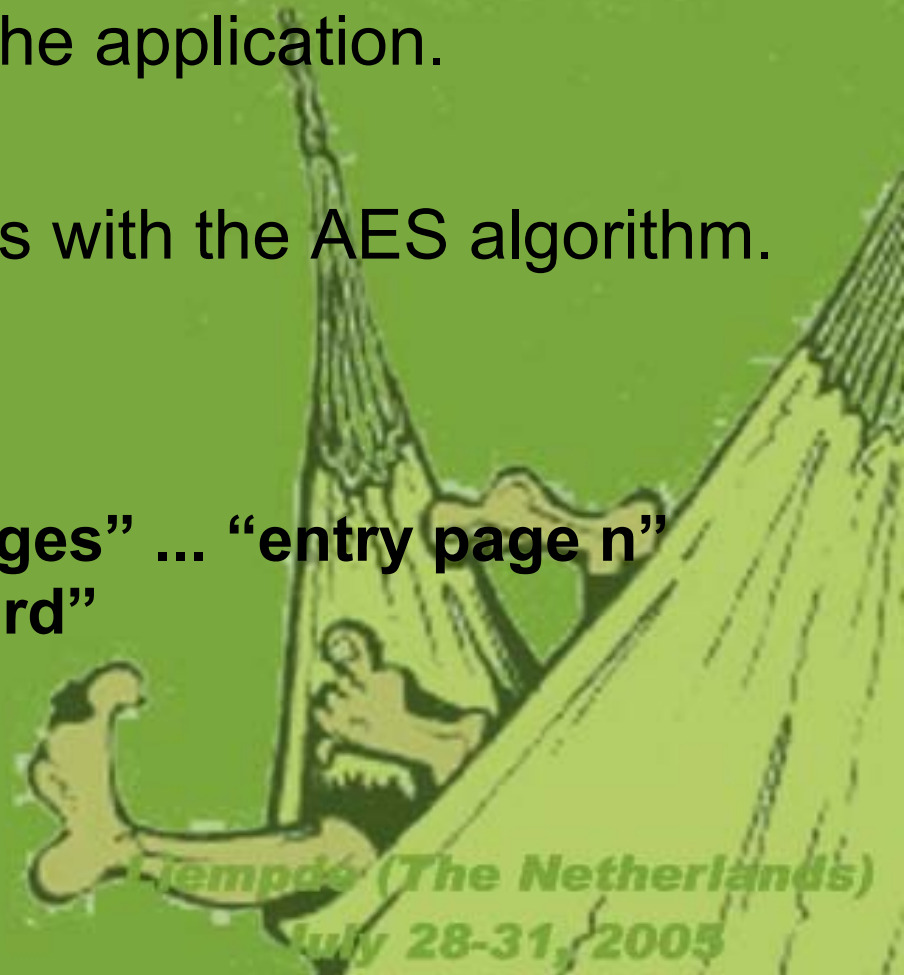
Again we use Cryptlib, to sign the links with the AES algorithm.

The directives to control this:

SecSignLinks On/Off

SecEntryPoint “/index.asp” “/images” ... “entry page n”

SecEncryptionPassword “password”



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Link Signing (II)

Signing the links we can tackle this threats:

Predictable Resource Location

Forceful Browsing: the attacker "forces" a URL by accessing it directly instead of following links.

Automated scanners like Nikto, dirb, and others, will be foiled.

Example of a link cripto-signed:

<http://www.securesign.com/help.asp?pagina=ayuda-login.asp&Secsign=MIHVBgkqhkiG9w0BBwOggccwgcQCAQAxcaNvAgEAoBsGCSqGSIlb3DQEFDDAObAhyxt2Mf3s4KQICAfQwlwYlKoZ...6DQDpC>

What The Hack!

Tempde (The Netherlands)

July 28-31, 2005

Form and Hidden Fields crypto-signing

Another security measure is the crypto-signing of form hidden fields and signing the forms itself, to prevent the values from being modified in the quantity, names, etc. of the inputs of the form.



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Log Console (I)

Web based Log Console

Facilitate the log analysis task

Snort Acid Style

Some of the listing that it provides:

Top 10 attacks or attackers

Today attacks

Last 15 Attacks

Other characteristics:

Attack details

Search

Delete records



Jempde (The Netherlands)

July 28-31, 2005

What The Hack!

Log Console (II)



Tempde (The Netherlands)
July 28-31, 2005

What The Hack!

Log Console (III)



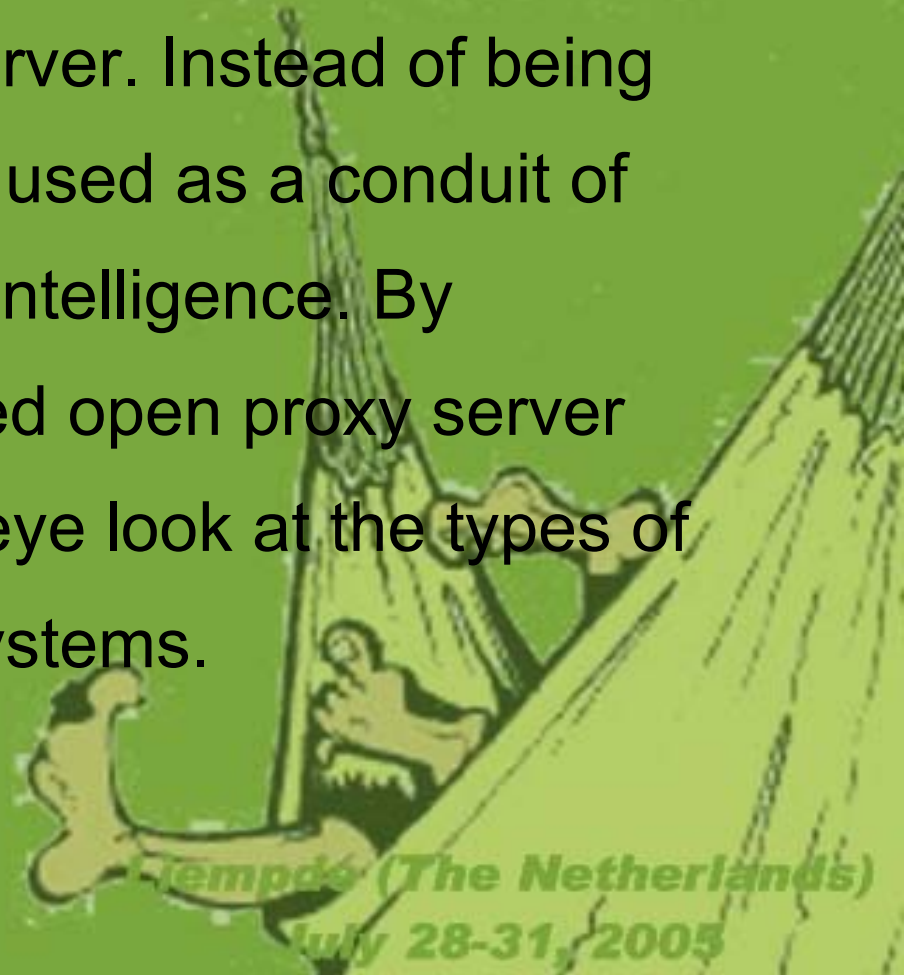
*Tempde (The Netherlands)
July 28-31, 2005*

What The Hack!

Open Proxy Honeypots (I)

Web App Security Consortium (WASC)

This project will use one of the web attacker's most trusted tools against him - the Open Proxy server. Instead of being the target of the attacks, we opt to be used as a conduit of the attack data in order to gather our intelligence. By deploying multiple, specially configured open proxy server (or proxypot), we aim to take a birds-eye look at the types of malicious traffic that traverse these systems.



What The Hack!

Tempde (The Netherlands)
July 28-31, 2005

Open Proxy Honeyspots (II)

The honeypot systems will conduct real-time analysis on the HTTP traffic to categorize the requests into threat classifications outlined by the Web Security Threat Classification and report all logging data to a centralized location.



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Examples

SQL Injection

XSS Scripting

Fine grained checks

Buffer Overflows

Positive Model

File extensions

Output filtering, errors.

Resource location prediction

Strip comment code

Blog/Forums spam protection



Tempde (The Netherlands)

July 28-31, 2005

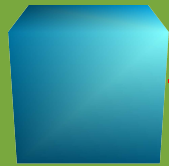
What The Hack!

Example Scenario

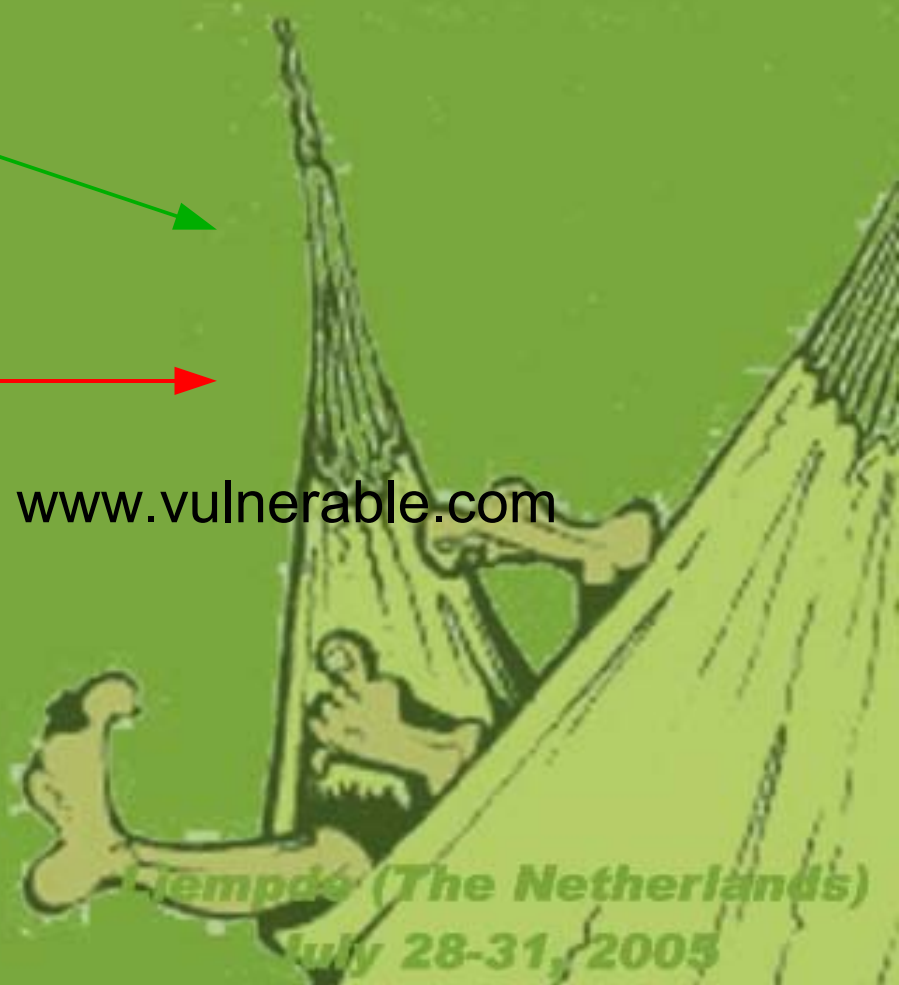
www.secure.com

modsecurity

mod_security + mod_proxy



www.vulnerable.com



Tempde (The Netherlands)
July 28-31, 2005

What The Hack!

SQL Injection

Vulnerable parameter: Login

Test String:

' or 1=1;--

Modsecurity rule:

Secfilter '+-- redirect:http://www.secure.com/error123.html

Positive way:

SecfilterSelective ARG_login ![a-zA-Z]+\$



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

XSS Prevention

Vulnerable parameter: pagina

Test String:

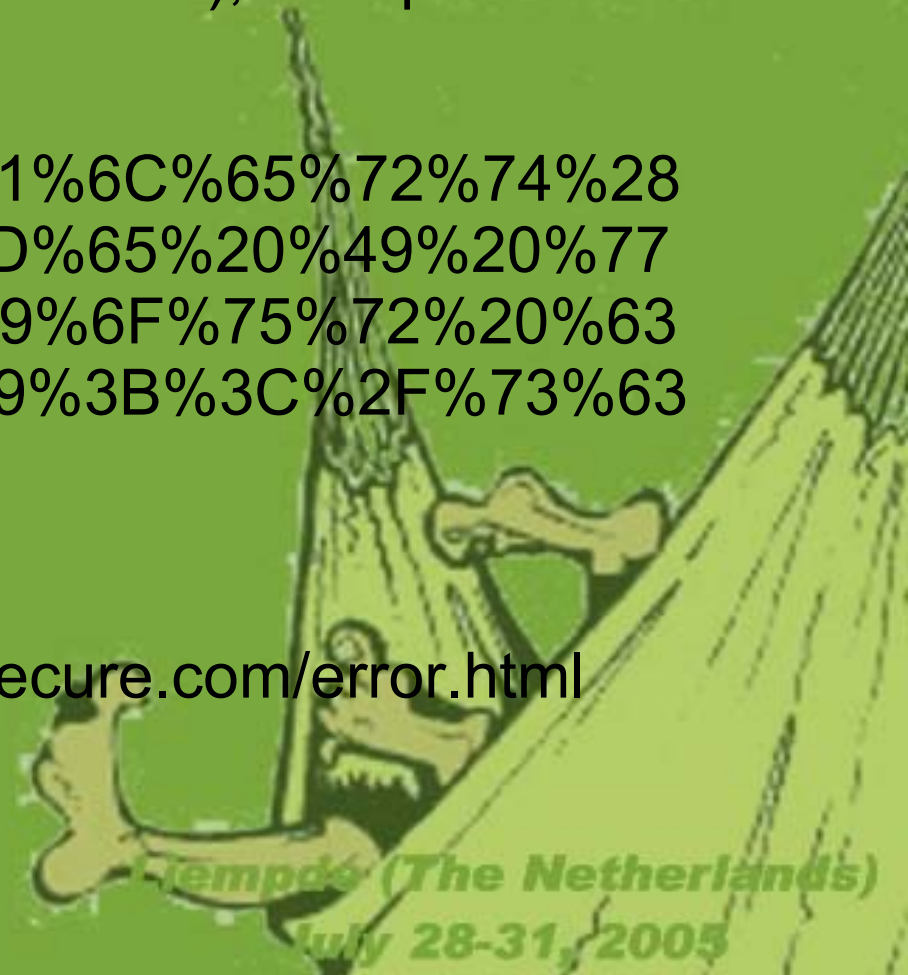
```
<script>alert('Next time I will eat your cookies ');</script>
```

Hex Encoded:

```
%3C%73%63%72%69%70%74%3E%61%6C%65%72%74%28%27%4E%65%78%74%20%74%69%6D%65%20%49%20%77%69%6C%6C%20%65%61%74%20%79%6F%75%72%20%63%6F%6F%6B%69%65%73%20%27%29%3B%3C%2F%73%63%72%69%70%74%3E
```

Modsecurity rule:

Secfilter “<(.\|n)+>” redirect:http://www.secure.com/error.html



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Fine Grained (I) Positive Model

Vulnerable parameter: login

Test String:

Test12-
121212

Modsecurity rule:

```
SecfilterSelective ARG_login ![a-zA-Z]+$
```

Look at the server version!



Tempde (The Netherlands)
July 28-31, 2005

What The Hack!

File Extension Protection

File extension to protect: *.txt

Requested file: admin/password.txt

Modsecurity rule:

SecfilterSelective SCRIPT_FILENAME .+\.txt\$

Positive way: SCRIPT_FILENAME !.+\.asp\$

If we use Link Signing we don't need to worry about this rules.



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Output Errors Protection

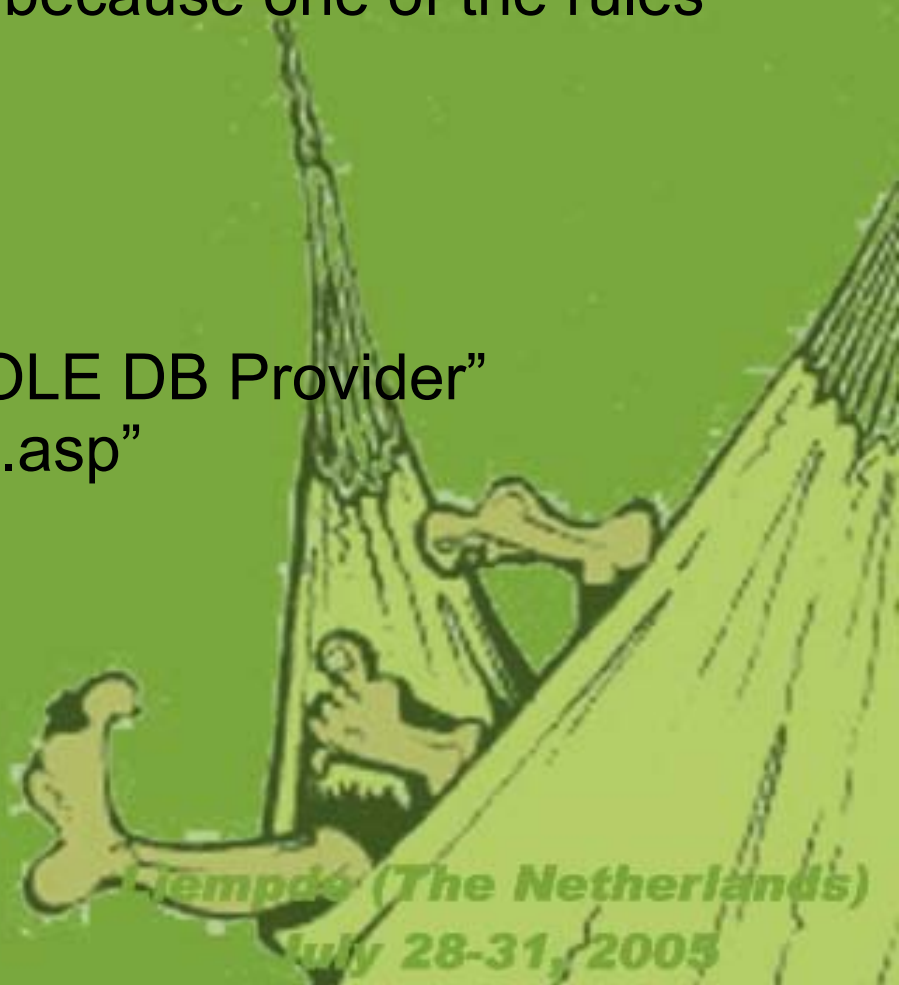
Vulnerable parameter: password

Test String:

' (single quote) and 1 char in the login because one of the rules before.

Modsecurity rule:

```
SecfilterSelective OUTPUT "Microsoft OLE DB Provider"  
"redirect:http://www.secure.com/default.asp"
```



Tempde (The Netherlands)
July 28-31, 2005

What The Hack!

Strip Comment Code

Vulnerable page: help.asp

Commented code:

```
<!-- remember that admin zone is in /admin/ -->
```

Modsecurity option:

SecStripCommenCode On



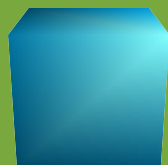
Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Blog Spam Scenario, Wordpress

www.myblog.com



Web Client

modsecurity

Apache/Wordpress

Here Modsecurity is working in the same server that is protecting.



Tempde (The Netherlands)
July 28-31, 2005

What The Hack!

Blog/Forums Spam Protection

Vulnerable parameter: comments

Spam message:

“Cheap viagra”

“Cheap vLagra”

“Cheap v1agra”

Modsecurity Rule:

```
Secfilter ARGS "v[i!L1]agra"  
redirect:http://www.myblog.com/spam.swf
```



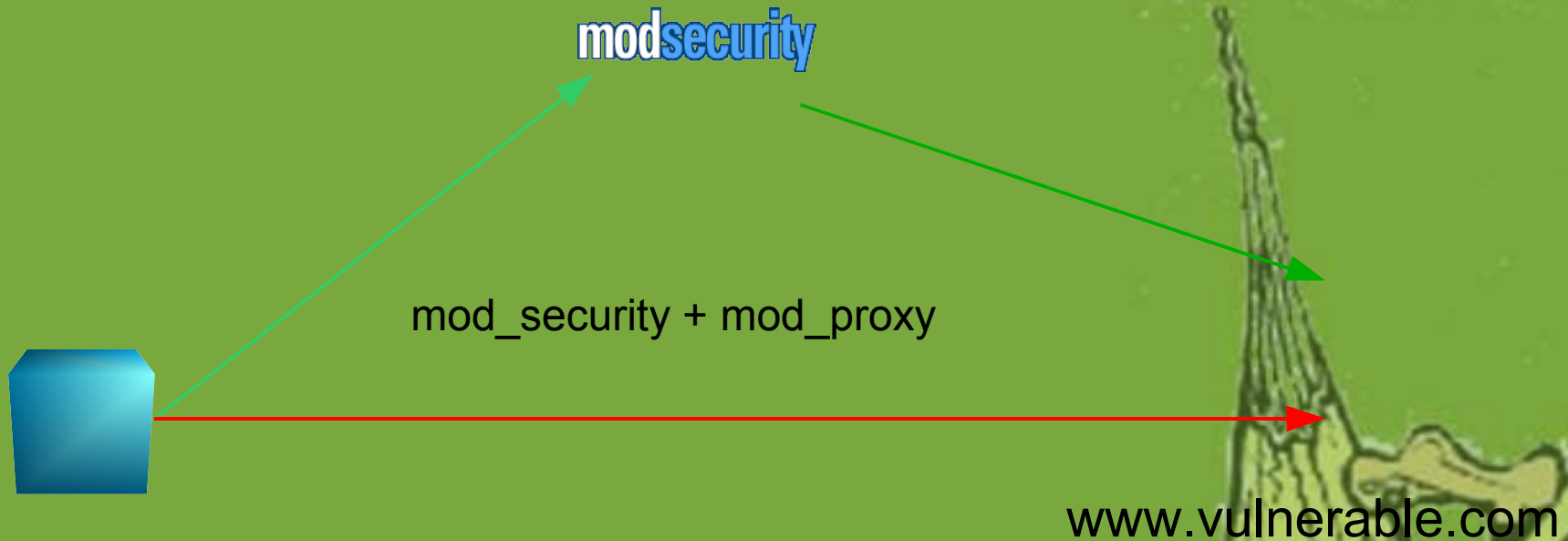
Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Link Signing Activated

www.securesign.com



www.vulnerable.com

What The Hack!

Tempde (The Netherlands)
July 28-31, 2005

Resource Location Prediction

Attack tool: Nikto

Attack options:

```
nikto -host www.vulnerable.com
```

```
nikto -host www.securesign.com
```

Modsecurity options:

```
SecSignLinks On
```

```
SecEncryptionPassword "our_Pass*Word-$"
```



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Resource Location using Nikto (I)



What The Hack!

*Tempde (The Netherlands)
July 28-31, 2005*

Resource Location using Nikto (II)



Tempde (The Netherlands)
July 28-31, 2005

What The Hack!

XSS with crypto-sign links enabled

Vulnerable parameter: pagina

Test String:

```
<script>alert('Next time I will eat your cookies ');</script>
```

Modsecurity rule:

```
Secfilter "<(.\n)+>"
```

Same for all other type of injection and variable manipulation, that involves a link, or direct access to URL.



What The Hack!

Tempde (The Netherlands)
July 28-31, 2005

File Extensions with Sign Links On

Vulnerable file extension: *.txt

Requested file: /admin/password.txt

Modsecurity rule:

SecfilterSelective SCRIPT_FILENAME .+\.txt\$

Positive way: SCRIPT_FILENAME !.+\.asp\$

We don't need to worry about this rules anymore!



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Conclusions

Modsecurity is a great choice for protecting your web applications

Easy to configure

Very effective

Remember that Modsecurity is an extra layer in our protection scheme, we have to secure our applications whenever we can.

There is still much work to do to improve this features, they are an alpha version.



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Question, doubts?



Tempde (The Netherlands)
July 28-31, 2005

What The Hack!

References

Download Modsecurity:

www.modsecurity.org

Mailing List:

mod-security-users@lists.sourceforge.net

Cryptlib:

<http://www.cs.auckland.ac.nz/~pgut001/cryptlib/>

Libxml2:

<http://xmlsoft.org/>



Tempde (The Netherlands)

July 28-31, 2005

What The Hack!

Thank you!



Advanced web application defense with Modsecurity

Daniel Fernandez Bleda
dfernandez@isecauditors.com

Christian Martorella
cmartorella@isecauditors.com

Coding is no easy
But programmers are patient



Tempde (The Netherlands)
July 28-31, 2005

What The Hack!