Logfileanalyse

Logfileanalyse in Dokuwiki

Um keinen Webalizer installieren zu müssen, wurde DokuWiki selbst um ein entsprechendes PlugIn erweitert:

```
http://www.sds-project.fr/howto/doku.php?id=wiki:statdisplay
http://www.sds-project.fr/howto/doku.php?id=wiki:logstats
```

Zusätzlich musste hierzu dann eine Grafikbibliothek für PHP installiert werden:

```
apt-get install php5-gd
```

Dann noch die

```
/etc/php5/apache2/php.ini
```

anpassen und das Modul aktivieren in der Sektion „Dynamic Extensions“

```
extension=gd.so
```

Ab jetzt logt DokuWiki selbst mit:

http://www.kvfg.info/doku.php?id=statistik

Webalizer

Webalizer wurde am 28.05.2010 wieder entfernt, weil awstats einfach hübscher ist.

Nunja - jetzt kommt er doch noch auf die Platte um auch die Zugriffe außerhalb des Dokuwikis besser dokumentieren zu können:

```
sudo apt-get install webalizer
```

Webalizers Webfrontend landet dann in

```
/var/www/webalizer/
```

Dann muss die Konfigurationsdatei angepasst werden:

```
vi /etc/webalizer/webalizer.conf
```

Und außerdem muss ein Cronjob erstellt werden:

```
crontab -e
```

Der cronjob soll jede Stunde laufen:

```
0 *  * * *  /usr/bin/webalizer
```

Jetzt muss auf die Webalizerseite noch ein Passwortschutz drauf:

```
vi /etc/apache2/sites-available/default
```

Hier dann eintragen:

```
<Directory /var/www/webalizer/>
AuthType Basic
AuthName "Protected Area"
AuthUserFile /etc/apache2/htaccess/webalizer.auth
require valid-user
</Directory>
```

 Nicht vergessen: Diese Anweisungen gehören in die default und in die default-ssl 

Damit das funktioniert muss das entsprechende Verzeichnis auch angelegt werden:

```
mkdir /etc/apache2/htaccess
```

In diesem Verzeichnis dann

```
htpasswd -c /etc/apache2/htaccess/webalizer.auth benutzername
```

Jetzt noch den Apachen neu starten

```
/etc/init.d/apache2 restart
```

und alles ist gut.

awstats

Webalizer sieht einfach altbacken aus. Awstats ist viel viel hübscher. Deswegen kommt der jetzt auch noch auf die Platte und dann wählen wir aus:

```
apt-get install awstats
```

Konfiguration

Dann wird die Apache Conf bearbeitet:

```
vi /etc/apache2/apache2.conf
```

Hier dann am Ende eingefügt:

```
# awstats
ScriptAlias /awstats /usr/lib/cgi-bin/
Alias /awstats-icon/ /usr/share/awstats/icon/
```

Jetzt bearbeiten wir die Konfigurationsdatei:

```
vi /etc/awstats/awstats.conf
```

Hier die Zeilen finden und den Inhalt abändern:

```
LogFile="/var/log/apache2/access.log"
SiteDomain="www.kvfg.info"
```

oder die awstats.conf kopieren und unter neuem Namen - z.B. kvfg.conf - hier wieder ablegen. Dies muss dann bei den folgenden Schritten berücksichtigt werden.

Eigentlich sollte für den Apache wohl LogFormat=4 passen, aber dann liest awstats keine Betriebssystem und Browserkennungen aus dem Log aus. Deswegen nutzen wir (testweise)

```
LogFormat=1
```

Cron muss nun auch noch angepasst werden, damit die Logs an den richtigen Stellen gesucht und gefunden werden und die richtigen Skripte aufgerufen werden - und hierzu editieren wir

```
vi /etc/cron.d/awstats
```

und schreiben den Inhalt wie folgt um:

```
*/10 * * * * root /usr/lib/cgi-bin/awstats.pl -config=/etc/awstats/kvfg.conf
-update > /dev/null
```

Der Apache wird nun einmal neu gestartet:

```
/etc/init.d/apache2 restart
```

und awstats einmal von Hand aufgerufen:

```
/usr/lib/cgi-bin/awstats.pl -config=kvfg.conf
```

Im Browser überprüfen wir dann, ob es Fehlermeldungen gibt (meist wegen dem doppelten Aufruf des Plugins für die Hashfiles) und korrigieren diese.

In Zukunft wird der händische Aufruf wie folgt erfolgen:

```
/usr/lib/cgi-bin/awstats.pl -config=kvfg.conf -update
```

Nun fehlt noch der .htaccess Schutz für die Webseite, der wie bei webalizer, aber mit den folgenden Pfaden in die entsprechende Siteconfig des Apachen eingetragen wird:

```
<Directory /usr/lib/cgi-bin/>
AuthType Basic
AuthName "Protected Area"
AuthUserFile /etc/apache2/htaccess/awstats.auth
require valid-user
</Directory>
```

 Nicht vergessen: Diese Anweisungen gehören in die default und in die default-ssl 

Dann den Schutz setzen

```
htpasswd -c /etc/apache2/htaccess/awstats.auth benutzername
```

Jetzt noch den Indianer neu starten:

```
 /etc/init.d/apache2 restart
```

Das müsste es weitestgehend gewesen sein.

Konfiguration 2

Wer sicherer sein will, nimmt für den Eintrag in /etc/cron.d/awstats den folgenden Eintrag vor, der jedoch nicht immer reibungslos funktioniert:

```
*/10 * * * * www-data /usr/lib/cgi-bin/awstats.pl -
config=/etc/awstats/kvfg.conf -update > /dev/null
```

Damit die Leserechte für die Apache access.log im zweiten Fall richtig gesetzt werden können, schreiben wir die Logfiles nun für eine andere Benutzergruppe (alternative Konfigurationsvorschläge ändern die Rechte auf 644 - aber die Welt hat hier IMHO nichts zu suchen):

```
vi /etc/logrotate.d/apache2
```

Hier dann die entsprechende Zeile suchen und adm durch www-data ersetzen:

```
create 640 root www-data
```

Jetzt noch die Leserechte der schon vorhandenen Logdatei anpassen:

```
chgrp www-data /var/log/apache2/access.log
```

Sollte schlicht nichts passieren, dann liegt dies an falsch gesetzten Rechten auf irgendwelchen Ordnern über den oben angegebenen Dateien: Da darf der Apache nicht rein. Man muss in diesem Fall die Ordnerrechte z.B. auf 755 setzen ... mit allen Nebenwirkungen.