Clean WordPress Security Logs

WordPress has some big security issues which, if not managed correctly. We use [https://ithemes.com/security|iThemes Security Pro] which allows us to lock down access to our sites and block most (or all) hacking/cracking attempts.

However, the logs are only stored in the database and no longer able to be cleaned automatically, so two tables (wp_itsec_logs,wp_itsec_lockouts) in each database grow without limits. The first (logs) was over 500k rows when I went to investigate what was using all of our disk space.

To fix this, I wrote a simple Perl script that is run weekly as a cron job, removing anything older than 4 weeks. When I ran this the first time, on a system which has just grown over the years, I removed almost 2 Gigabytes from the half dozen sites that were affected.

This script can be downloaded from our Subversion repository with the command

```
svn co
http://svn.dailydata.net/svn/sysadmin_scripts/trunk/Wordpress/cleanWPSecLogs
```

. Note: this is our working site, and we will make updates and changes without warning, but since we use this for our stuff, we try to not break things! Or, you can simply download the file here.

cleanWPSecLogs

```
#! /usr/bin/env perl

# Copyright (c) 2024, Daily Data, Inc
#
# Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following
# conditions are met:
#
#  Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
#  Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided with
the distribution.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
# BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
# SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
# DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
# INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER
IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
```

```
# OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.[12]
#
# Description
# Script to clean up security log tables on WordPress sites
# Some WordPress site security suites continuously add logs, but do not
offer the ability to clean up entries after
# a certain period of time. This can cause database tables to grow
without relief until they threaten to fill
# the database partition
# This came to light recently when the half dozen WordPress sites we
manage threatened to fill the database partition
# after a few years of monitoring.
#
# Before we ran this script, the 6G partition was 81% full, and after
cleaning up, that dropped to 52%, so almost a
# third of the database size was years of logs, or almost 2G. Of
course, you can increase the disk space available on
# the partition, but we just wanted to keep 4 weeks of logs
#
# Designed to be run from a monthly or weekly cron job, it will look
for all MySQL databases, then check each one in turn
# for the existence of the security log tables. If a database contains
the tables, it will run the script to remove
# anything over $KEEPDAYS days old.
#
# this is a quick script and has only been tested on one single system,
where the root user had access to all MySQL
# functions without password. Modifications will need to be made on
systems which are more secure.
#
# This script is available via subversion at
# svn co
http://svn.dailydata.net/svn/sysadmin_scripts/trunk/Wordpress/cleanWPSe
cLogs
# NOTE: the above repository is a working copy from Daily Data, which
can be modified as required without notification


use strict;
use warnings;

my $KEEPDAYS = 28;

# clean up the output from mysqlshow
# if the name does not match the regular expression
# return null, else return the matched code
sub clean {
   my $name = shift;
   # regex matches any database beginning with c## for our ISPConfig
setup
```

```
    # pretty sloppy regex, but works on our system
    $name =~ m/(c\d+[a-z0-9_]+)/i;
    return $1 ? $1 : '';;
}

# create a temporary sql file which is sent to all valid databases
# NOTE: it is the responsiblity of the script to remove the temp file
(we don't do it here)
sub createSQL {
    use File::Temp;
    my $fh = File::Temp->new( SUFFIX => '.sql', UNLINK => 0, TEMPLATE=>
'/tmp/cleanLogs.XXXXXXXXXX' );
    # NOTE: if you don't want any logs kept, you can use 'truncate
wp_itsec_logs' instead, which is faster and also
    # recovers disk space
    # delete anything older over $KEEPDAYS old from wp_itsec_logs.
timestamp appears to be UT, so will be off by some
    print $fh "DELETE FROM wp_itsec_logs WHERE timestamp
<DATE_SUB(NOW(), INTERVAL $KEEPDAYS DAY);\n";
    # same with wp_itsec_lockouts
    print $fh "delete FROM wp_itsec_lockouts WHERE lockout_expire <
DATE_SUB(NOW(), INTERVAL $KEEPDAYS DAY);\n";
    # next two commands recover disk space from tables. You can ignore
this after the first run as very little will remain
    # between sessions
    print $fh "optimize table wp_itsec_logs;\n";
    print $fh "optimize table wp_itsec_lockouts;\n";
    close $fh;
    return $fh->filename; # returns the name of the temp file
}

my $sqlFile = &createSQL(); # makes the sql file to be run on each
database
# print "$sqlFile\n";

# simple way to get list of all databases, but they need to be cleaned
up afterwards
my @tables = `mysqlshow`;
for ( my $i = 0; $i < @tables; $i++ ) {
    $tables[$i] = &clean( $tables[$i] );
}
#die join( "\n", @tables ) . "\n";

# process each table
for ( my $table = 0; $table < @tables ; $table++ ) {
    next unless $tables[$table]; # skip anything that &clean zapped
    # command passed to mysql to determine if the table wp_itsec_logs
exists in database
    my $command = "echo show tables like \\'wp_itsec_logs\\' | mysql
$tables[$table]";
    #die "$command\n";
```

```
   my $valid = `$command`; # if $command has something, it is valid. If
it is the empty string, no table, so not valid
   #die "'$valid'\n";
   if ( $valid ) {
      print "Found in $tables[$table]\n";
      print `mysql $tables[$table] < $sqlFile`; # run mysql against
database passing the commands from the temp file in
   }
}
# remove the temp file
unlink( $sqlFile ) or die "Could not delete temp file $sqlFile\n";
1;
```

From:
https://wiki.linuxservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://wiki.linuxservertech.com/software/wordpress/cleanlog

Last update: **2024/04/05 23:51**