Home
PUBLIC
Stack Overflow
Tags
Users
FIND A JOB
Jobs
Companies
TEAMS What's this?
Free 30 Day Trial

Best practices for holding passwords in shell / Perl scripts?

Asked 12 years, 2 months ago Active 3 years ago Viewed 12k times

Ask Question

11

I've recently had to dust off my Perl and shell script skills to help out some colleagues. The colleagues in question have been tasked with providing some reports from an internal application with a large Oracle database backend, and they simply don't have the skills to do this. While some might question whether I have those skills either (grin), apparently enough people think I do to mean I can't weasel out of it.

So to my question - in order to extract the reports from the database, my script is obviously having to connect and run queries. I haven't thus far managed to come up with a good solution for where to store the username and password for the database so it is currently being stored as plaintext in the script.

Is there a good solution for this that someone else has already written, perhaps as a CPAN module? Or is there something else that's better to do - like keep the user / password combo in a completely separate file that's hidden away somewhere else on the filesystem? Or should I be keeping them trivially encrypted to just avoid them being pulled out of my scripts with a system-wide grep?

Edit: The Oracle database sits on an HP-UX server.
The Application server (running the shell scripts) is Solaris.
Setting the scripts to be owned by just me is a no-go, they have to be owned by a service account that multiple support personnel have access to.
The scripts are intended to be run as cron jobs.
I'd love to go with public-key authentication, but am unaware of methods to make that work with Oracle - if there is such a method - enlighten me!

perl oracle bash ksh

share improve this question follow — edited Apr 9 '17 at 9:55 Cœur 30.6k 16 168 219 — asked Sep 17 '08 at 6:50 GodEater 2,907 2 23 29

add a comment

13 Answers

Active Oldest Votes

7

Best practice, IMHO, would be to NOT hold any passwords in a shell / Perl script. That is what public key authentication is for.

share improve this answer follow — edited Oct 8 '08 at 7:43 brian d foy 118k 31 191 531 — answered Sep 17 '08 at 6:55 Redbeard 978 5 8

Is that easy to set up with Oracle (10g I think) ? – GodEater Sep 17 '08 at 6:57

1 After a bit of hunting on the Oracle website, it seems "Oracle Wallet" is what I want. Now to go badger my DBAs to get it implemented. Thanks very much :) – GodEater Sep 17 '08 at 7:18

add a comment

5

If the script is running remotely from the server.

1. Make your reports views
2. Give the user you are logging into ONLY access to select on the report views
3. Just store the password.

That way, all that the user can do, is select the data for its report. Even if someone happened to get the password, they would be limited as to what they could do with it.

share improve this answer follow — answered Sep 17 '08 at 7:07 Matthew Watson 13.6k 8 59 81

add a comment

4

Personally I hold passwords in configuration files which are then distributed independently of the application, and can be changed to the specific machine/environment. In shell scripts you can source these within the main script.

However, in Perl there are a variety of approaches. You may wish to investigate Getopt::Long for command line options (and additionally Getopt::ArgvFile to store those in a simple configuration file), or look at something like Config::IniFiles for something with a little more power behind it. These are the two types I generally use, but there are other configuration file modules available.

share improve this answer follow — answered Sep 17 '08 at 12:58 Barbie 81 3

add a comment

1

There is no good solution. You can obfuscate the passwords a bit, but you can't secure them.

If you have control over your DB setup, you could try to connect by a named pipe (at least mysql supports that) without a password and let the OS handle the permissions.

You could also store the credentials in a file with restrictive permissions.

share improve this answer follow — answered Sep 17 '08 at 6:53 moritz 12.2k 1 35 59

The database is running on a physically different server than the one which executes the shell script. To make matters worse, the OS of the database server is HP-UX, and the OS of the shell script server is Solaris, so I'm not sure the named-pipe approach would work. – GodEater Sep 17 '08 at 6:57

add a comment

1

Since you've tagged ksh & bash I'm going to assume Linux.

Most of the problem is that if the user can read the script and locate the method you used to hide / encrypt the file then they will also be able to do the same thing manually.

A better way may be do the following:

1. Make your script so it can only be seen/read/opened by you. chmod 700 it. Hardcode passwords away.
2. Have a "launcher" script that is executable by the user and does a sudo .

This way the user can see your launcher script, examine it to see it only has the single command line. They can run it and it works, but they don't have permissions to read the source for the script that is sudo'd.

share improve this answer follow — answered Sep 17 '08 at 6:57 Paul Hargreaves 1,618 3 21 23

HP-UX for the database server, Solaris for the server the shell scripts are executing on actually ;) The scripts are intended to execute as a 'service account' that a lot of people have access to (for support reasons), and are launched by a cron job. Would the sudo approach still work in that case? – GodEater Sep 17 '08 at 6:59

Most crons won't work if you put the sudo into it directly - e.g. a crontab with "0 0 0 0 0 sodo myjob.sh" It would work fine in a cron job because the cron job is just calling the launcher script and the launcher script is going to run sudo. – Paul Hargreaves Sep 17 '08 at 7:17

add a comment

1

I'm not sure what version of Oracle you are running. On older version of Oracle (pre 9i Advanced Security) some DBA's would `CREATE USER OPS$SCOTT IDENTIFIED BY EXTERNALLY` and set `REMOTE_OS_AUTHENT` to true.

This would mean that your remote sun machine could authenticate you as SCOTT and then your Oracle DB would accept that authentication.

This is a bad idea.

As you could image any Windows XP with a local user of SCOTT could then log into your DB without a password.

Unfortunately it's the only option that i know of Oracle 9i DBs to not store username/passwords in your script or somewhere else accessible by the client machine.

What ever your solution it's worthwhile having a look through Oracle's Project Lockdown before committing.

share improve this answer follow — answered Sep 17 '08 at 9:19 Mark Nold 5,332 6 27 33

add a comment

1

For storing passwords you could do a two step encryption routine, first with a hardcoded key in your script itself, and optionally a 2nd time with a key stored in a file (which is set using file permissons to have restricted access).

In a given situation you can then either use a key file (+ key from script), or if the situation requirements aren't that great he can just use the encyrption using the key is hardcoded in the script. In both cases the password would be encrypted in the config file.

There is no perfect solution because somehow you have to be able to decrypt and obtain the cleartext password...and if you can do it someone else can too if they have the right info.

Especially in the situation where we give them a perl script (vs. an exe) they can easily see how you do the encryption (and the hardcoded key)...which is why you should allow the option to use a keyfile (that can be protected by filesystem permissions) as well.

Some practical examples for how to implement is here

share improve this answer follow — edited Sep 17 '08 at 17:05 — answered Sep 17 '08 at 16:22 Chris Madden 2,142 2 14 11

add a comment

0

In UNIX, I always make these scripts setuid and store the user and password info in a file that's heavily protected (the entire directory tree is non-readable/searchable by regular users and the file itself is readable only by the owner of the script).

share improve this answer follow — answered Sep 17 '08 at 6:57 paxdiablo 747k 200 1438 1805

1 I thought setting scripts (of any kind) suid was bad practise? I'm sure I've read that somewhere... – GodEater Sep 17 '08 at 7:00

1 Here is a great reference for why suid is generally not advised. samag.com/documents/s=1149/sam0106a/0106a.htm – Paul Hargreaves Sep 17 '08 at 9:04

setuid scripts are fine as long as the scripts themselves cannot be tricked (that's a whole other subject). – paxdiablo Sep 19 '08 at 6:06

Will Deich's super command is good for managing suid execution for scripts: it's sort a sudo for scripts. Cf. ucolick.org/~will/#super – Charles Stewart Feb 1 '10 at 13:09

add a comment

0

Keep them in a separate file, trivially encrypted, and make a separate user in the database with read only access to necessary tables. If you think the file has been read, then you can shut off access to just that user.

If you want to get fancy, a SUID program could check the /proc//exe and cmdline (in Linux), and only then release the username.

share improve this answer follow — answered Sep 17 '08 at 7:01 Tony_I

add a comment

0

I have / had a similar issue with developers deploying SQL code to MSSQL (in fact to any database on that MSSQL server, so role had to be SysAdmin) using ANT from a Solaris server. Again I did not want to store the username and password in the ANT build.xml files so my solution, which I know is not ideal, is as follows:

1. Store name / value pairs for username and password in a plain text file
2. Encrypt file (on Solaris) and use a pass phrase only known to certain admins
3. Leave only the encrypted file on the Solaris system
4. ANT build.xml runs a sudo decrypt and prompts for pass phrase, which is entered by admin
5. ANT sources decrypted file loading username and password as variables for the SQL string
6. ANT immediately deleted the plaintext file
7. ANT deploys code and exits

This all happens in a matter of seconds, and the sql username and password is never visibly accessible on the server. As the code is deployed by allowed admins in production, the developers never need to include it in their code.

I am sure it could be better, but...

JB

share improve this answer follow — answered Sep 17 '08 at 11:15 Jonathan Bourke 215 2 10

add a comment

0

It's a shame I never saw this thread before -- it looks very interesting. I'll add my two cents for anyone coming upon the thread in the future.

I'd recommend using OS authentication on the db server itself -- REMOTE_OS_AUTHENT is still FALSE.

If you're invoking the script from another machine, setup a phrase-less SSH key and use SSH to get there. You can then pipe back the SQL results to the calling machine and it can process this information further.

Doing this avoids having to code a password anywhere. Of course, if a malicious administrator were to hijack the phrase-less key and use it, he or she could also access the user account on the DB host and could then do any operations the OS authenticated DB user could. To mitigate this you could reduce the database permissions for that OS user to the bare minimum -- let's say "read only".

Ingo

share improve this answer follow — answered Jun 12 '09 at 21:31 Ingo

add a comment

0

> On windows create a Folder and a File within it containing the passwords in clear text. Set the user who would run the scheduled job(script or batch) as the only person with read/write access to this folder and file. (remove even administrator). To all other scripts, add code to read the clear text password from this restricted file.

This should suffice for few.

Keywords: Password HardCoding

share improve this answer follow — edited Jul 14 '09 at 10:26 — answered Jul 14 '09 at 10:21 karthik

add a comment

0

There are commercial or more advance solutions such as cyberark AIM can do it better, but doing it for free and out of box, I have been piggy back the SSH public/private key because for one, SSH key pairs most likely already created conform the security policy; secondly, SSH key pairs are already have a set of standard protocol to protect the keys by the file permission, continuous system hardening (like tripwire), or key rotation.

This is how I did it:

1. Generate the ssh key pairs if not yet. The key pairs and directory will be protected by default system protocol/permission. ssh-keygen –t rsa –b 2048
2. use the ssh public key to encrypt the string and stored in same .ssh directory $ echo "secretword" | openssl rsautl -encrypt -inkey ~/.ssh/id_rsa.pub -pubin -out ~/.ssh/secret.dat
3. use ssh private key to decrypt the key, and pass the parameters to scripts/AP in the realtime. The script/programe to include the line to decrypt in realtime: string=`openssl rsautl -decrypt -inkey ~/.ssh/id_rsa -in ~/.ssh/secret.dat`

PS - I have been experimenting CYBERARK AIM agentless solution. it's sort of pain requires significant changes/API changes for the API/script. will keep you posted how that goes.

share improve this answer follow — answered Nov 20 '17 at 22:34 Kexin Z 35 1

add a comment

Linked
1 Perl: Handing $dbh off to module. Security and performance?

Related
3850 How can I check if a directory exists in a Bash shell script?
1507 How do I prompt for Yes/No/Cancel input in a Linux shell script?
1162 YYYY-MM-DD format date in shell script
1055 How can I declare and use Boolean variables in a shell script?
903 Propagate all arguments in a bash shell script
1456 Check existence of input argument in a Bash shell script
817 How to call one shell script from another shell script?
900 Replace one substring for another string in shell script
757 How do I pause my shell script for a second before continuing?

Hot Network Questions
Motor will not go above half thrust. Why and how to fix?
A colorful little Connect Wall
How to migrate data from MacBook Pro to new iPad Air
What does "blaring YMCA — the song" mean?
empty while loop: bad practice?
Tatamibari: an introduction
After what time interval do the closest approaches of Mercury to the Earth repeat?
Can the Battle Master fighter's Precision Attack maneuver be used on a melee spell attack?
How to calculate current flowing through this diode?
Density of States of Supercells
What is Qui-Gon Jinn saying to Anakin by waving his hand like this?
When and why did the use of the lifespans of royalty to limit clauses in contracts come about?
How does the title "Revenge of the Sith" suit the plot?
DO I have the correct idea of time dilation?
Do far-right parties get a disproportionate amount of media coverage, and why?
Why is SQL Server's STDistance Very Slightly Different Than The Vincenty Formula? (Same Up To ~0.0001km)
BJT and signal source
Why do people call an n-sided die a "d-n"?
Why does C9 sound so good resolving to D major 7
What happens if my Zurich public transportation ticket expires while I am traveling?
Could we send a projectile to the Moon with a cannon?
Mountain inside a "crater" after a meteor impact?
To become a better guitar player or musician, how do you balance your practice/training on lead playing and rhythm playing?
Have any other US presidents used that tiny table?

Question feed

Your Answer

Not the answer you're looking for? Browse other questions tagged perl oracle bash ksh or ask your own question.