

RE: local admin account password

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/focus-ms/2003-11/0166.html>

From: shimi (shimi_at_shimi.net)

Date: 11/27/03

Date: Thu, 27 Nov 2003 06:35:12 +0200 (IST)
To: "Rockliff, Max M.J." <Max.Rockliff@woodside.com.au>

My point was to give a solution for the problem "remember 1000 different passwords". That way you don't need 1000 different passwords. You need to only know one password, and if a cracker cracks a machine on your network, that password is useless for him for the rest of the network.

OF COURSE that you can use that trick and run a monthly change of ALL the passwords in the organization, by just replacing the special keyword; The advantage of "no single password"+"no central password repository available" still exists – and that was my point – a solution for most of the problems raised during the thread:)

On Thu, 27 Nov 2003, Rockliff, Max M.J. wrote:

- > *With all due respect, how is this any different to randomly generating a*
- > *long password string and storing it in a secured location(locked safe,*
- > *encrypted database, etc). Ultimately the password is static and*
- > *therefore discoverable (learnt or lophtrcracked etc), which is the actual*
- > *vulnerability.*
- >
- > *Now if this could be combined with a database system that periodically*
- > *changed the password, permitted authorised users (maybe even using*
- > *Windows authentication) to check out a particular password for a defined*
- > *period, after which the system would automatically change the password*
- > *thereby locking out the user who had access – of course the system*
- > *should let the current user extend the use of the password, perhaps*
- > *within defined limits. Such a system would ensure that no one actually*
- > *knows a password unless they formally check one out, that the passwords*
- > *would be sufficiently complex to slow down crackers and that passwords*
- > *would be changed with enough frequency to further mitigate the risk of*
- > *cracking. The system itself should keep a log of all successful and*
- > *unsuccessful check-outs, along with a record of password maintenance,*
- > *encrypt the database and create alerts in the event of unsuccessful*
- > *requests or failure to maintain passwords.*
- >
- >
- > -----Original Message-----

SecurityFocus Microsoft: RE: local admin account password

> From: shimi [mailto:shimi@shimi.net]
> Sent: Thursday, 27 November 2003 1:40 AM
> To: eallen@bcpl.net
> Cc: focus-ms@securityfocus.com
> Subject: RE: local admin account password
>
>
>
> I have an idea.
>
> Find a given word (keep it secret!); Concat it into the machine's name;
>
> Take the concated string and run it through md5().
> (<http://www.php.net/md5>)
>
> Put the result as the admin password.
>
> Next time you need the password, you need to know: Your secret word +
> The
> machine name. (Do note – they WILL BE case sensitive, so make them all
> small or all big, or, whatever you want...)
>
> All you need is a webserver and PHP enabled to generate your passwords
> in
> future use... (secret word isn't stored anywhere!)
>
> I have set up a living example at <http://shimi.staff.fresh.co.il/md5.php>
>
> Here is the source code for that PHP file:
>
> ---- CUT HERE ----
>
> <form action="<?=\$_SERVER["PHP_SELF"];?>" method="post"> Machine's Name:
> <input type="text" size="30" name="machine">
 Secret passphrase:
> <input type="password" size="30" name="password">

> <input type="submit" value="Generate me my password!">
> </form>
>
> <?
> if(\$_POST) {
> \$machine = \$_POST["machine"];
> \$password = \$_POST["password"];
> echo "For administrative access to machine \$machine, use the password:
> ".md5(\$machine.\$password); } ?>
>
> ---- CUT HERE ----
>
>
> > -----Original Message-----
> > From: Eli Allen [mailto:eallen@bcpl.net]
> > Sent: Tuesday, November 25, 2003 12:47 PM

RE: local admin account password

SecurityFocus Microsoft: RE: local admin account password

> > *To: focus-ms@securityfocus.com*
> > *Subject: local admin account password*
> >
> >
> > *Say you have more then 1000 systems, how do you handle the local admin*
>
> > *account password on the machines? (assuming it needs to be available*
> > *for extreme cases to get into the machine as you'd normally just use a*
>
> > *domain*
> > *login)*
> >
> > *A few ways I can think of (in order from what I think is worst to*
> > *best):*
> > *1) use the same password on all boxes. Obviously insecure*
> >
> > *2) Use a different password on all boxes and a big filling cabinet to*
> > *secure it (as its impossible to memorize). Don't think this would*
> > *work in the real world so not worth using.*
> >
> > *3) Use a password scheme where the password is basically the same on*
> > *all box*
> > *except its based on something specific about the server. This means*
> > *if*
> > *someone figures out the scheme (cracking a single box and figuring it*
> > *out or just gets told) they basically made this as good as the first*
> > *idea I list.*
> >
> > *4) Only use domain accounts so delete the local ones. But this means*
> > *no more recovery console and don't think cached logins will work.*
> > *With so many boxes and hence lots of admins you may not have logged*
> > *onto the box and so not have cached login in the cache even if you*
> > *increased the logins that can be cached.*
> >
> > *5)My main idea/plan is to store all the passwords on a central SQL*
> > *server. This way you can easily have a different random passwords for*
> > *the admin accounts on all the boxes.*
> >
> > *The DB file would be encrypted with EFS so only the limited user SQL*
> > *runs under has access to the file and another user just used for doing*
>
> > *backups of this file. This means an attacker can't use an OS break-in*
>
> > *to get to the data and needs to compromise SQL or one of those two*
> > *user accounts. SQL would be set to integrated auth and only allow the*
>
> > *domain groups who are allowed access to the admin password in. (i.e.*
> > *using the access rights already existing)*
> >
> > *For data recovery (this DB is very important not to lose) there are*
> > *two main considerations, one the file is small as the DB has very*

RE: local admin account password

SecurityFocus Microsoft: RE: local admin account password

> > little info in it and two it doesn't get updated very often. The
> > backup user can make a zip backup of the DB whenever it gets changed
> > and then encrypt the file (PGP or something like it with the private
> > key stored on a/multiple CD-R(s) somewhere safe) Then this file could
> > be copied to lots of employee's desktops. Its encrypted so they can't
>
> > read it and with lots of people having the file the likelihood of
> > everyone's copy being damaged from HDD failure is low. (Yes will use
> > tape backup of the file too including off site storage but tape is
> > slow and should only be used if necessary) If there is an emergency
> > the managers could easily allow the file to be decrypted and then
> > attached to any SQL server available relatively quickly.

> >
> > Access to this file can be made by any utility that can make use of
> > stored procedures. There would be basically two stored procs, one to
> > get a password from the DB and one to set the password in the DB both
> > would have 3 values (machine name, username, and password) passed in
> > and out (obviously depending on which you use). This way if a person
> > decides to try and dump the DB and get all the passwords the stored
> > proc can do something about it (alert management, stop it from
> > happening, or something like that) This way its easy to write
> > whatever interface you want to be able to do access the DB and the app

>
> > itself doesn't really need to be secure as the authentication is based
>
> > on the user that app is run by.

> >
> > Yes I realize it has a central point of attack at the DB but I think
> > that can be secured well enough and the design is secure that its
> > still better then the other methods.

> >
> > Any comments? Thanks

> >
> > Eli Allen
> > eallen@bcpl.net

> >
> >
> >

> > -----
> >
>

> ---
> >
> >
> >
> >
> >

> > -----
> >

SecurityFocus Microsoft: RE: local admin account password

>

> -----

> >

>

>

--

Best regards,
Shimi

"Outlook is a massive flaming horrid blatant security violation, which also happens to be a mail reader."

--The best way to accelerate a Windows machine is at 9.81 m/s^2 --

"Windows is a 32-bit port of a 16-bit GUI for an 8-bit OS on a 4-bit CPU made by a 2-bit company that can't stand 1 bit of competition."

