Nikolay Okhitin, Fotolia

Secure authentication with one-time passwords

WHISPERED ONCE

A one-time password won't compromise security if it falls in the wrong hands. OPIE and OTPW bring the safety of one-time password security to Linux. **BY UDO SEIDEL**

Despite the biometrics boom, passwords are still the most popular means of authentication. In hostile environments, rogue users try to sniff or log password entries. You can foil these attempts by using one-time passwords. A one-time password becomes obsolete after it is used.

Even if an attacker were to sniff the password en route to the authentication server, the password would be useless. For a one-time password to work, the cli-

Listing 1: Initializing OPIE

```
01 # opiepasswd
02 Adding root:
03 You need the response from an OTP
   generator.
04 New secret pass phrase:
05         otp-md5 499 te3049
06         Response:
07 ^C
08 # opiepasswd -c
09 Adding root:
10 Only use this method from the
   console; NEVER
11 from remote. If you are using telnet,
   xterm,
12 or a dial-in, type ^C now or exit
   with no
13 password. Then run opiepasswd without
   the -c
14 parameter.
15 Sorry, but you don't seem to be on
   the console
16 or a secure terminal.
17 # opiepasswd -cf
18 Adding root:
19 Only use this method from the
   console; NEVER
20 from remote. If you are using telnet,
   xterm,
21 or a dial-in, type ^C now or exit
   with no
22 password. Then run opiepasswd without
   the -c
23 parameter.
24 Using MD5 to compute responses.
25 Enter new secret pass phrase:
26 Again new secret pass phrase:
27
28 ID root OTP key is 499 te5843
29 DANG TOOK HUNT GYM HICK PAW
30 # cat /etc/opiekeys
31 root 0499 te5843     6f1dba738c197a64
32                    Feb 16,2008 05:42
```

ent must have some means for determining what password to use, and the server must know what password to expect.

Techniques

Security experts have developed several techniques for generating one-time passwords. Some methods base a new password on a mathematical manipulation of the previous password – or on a mathematical manipulation of the current time. Another technique known as challenge-response starts with the server sending a random number to the client. The client then calculates a response using a process that is known to both parties.

Of course, an attacker who sniffs a couple of these challenges and responses could theoretically uncover the method. This crypto-analysis technique, which is often called *known plaintext*, has been described in several scientific publications. But if both partners apply a hash function after calculating the response, a sniffer will find it far more difficult to uncover the original value. The result looks very much like a random number.

These kinds of calculations are difficult to do in your head, so users often employ an electronic device called a *token*, which looks something like a pocket calculator. Figures 1 and 2 show examples of some popular tokens. Another option is to set up a mobile phone or PDA with the necessary software to act as a hardware-based token.

Figure 1: The Digipass Pro 300 by Vasco relies on the challenge-response approach. The user types the challenge via the keypad on the token and reads off the response from the display.

Software-Based Solutions

Of course, tokens are relatively expensive; also, the technology is often patented, or else the internal mechanisms are not fully disclosed as a security measure. If you prefer to avoid the effort and expense of a hardware token, you can also use a software-only solution.

Software-based one-time password systems have been around for several years are are even enshrined in a number of Internet RFCs. The S/Key system, which was developed in 1995 by Bellcore, is defined in RFC 1760. S/Key originally relied on MD4 encryption. Its successor, OTP, which is specified in RFC 2289, can also use MD5 and SHA hashes.

Universal OPIE

A pair of open source projects known as OPIE [3] and OTPW [4] provide one-

The 40 Watt Rackmount Server

Our new R410-EE-1U rackmount server uses an average of just 40 watts. That's less power consumption than most incandescent light bulbs.

With prices starting from just **£548** + VAT, they are easy on your budget and easy on the environment.

- Dual-Core AMD Athlon 64-bit platform
- 1U rackmount
- Low power Serial ATA disks
- Cool & quiet
- Choice of Linux OS

Prices as of 1st July 2008. Goto www.dnuk.com for full details.

Digital Networks United Kingdom

www.dnuk.com sales@dnuk.com 0161 343 5333

Figure 2: The Secur-ID-Token SID700 by RSA/EMC encodes the current time of day and the internal key. The display shows a different PIN every minute.

time password tools for Linux. The leading OTP software implementation on Linux comes courtesy of the OPIE project (One-Time Passwords in Everything).

OPIE is easily installed from the packages that exist for many distributions, and easily built from the sources. The installation adds OTP-capable programs for *login*, *su*, and *ftpd*, as well as the *pam_opie.so* library, a number of tools, and the */etc/opiekeys* configuration.

The first step is to initialize the OTP system (see Listing 1). Users handle this step themselves by logging on to the system and executing the *opiepasswd* command (Line 1). The results might be confusing at first glance (Line 3); by default, the tool assumes that the user is not logged on locally at the console.

Because network traffic is often sniffable and insecure, *opiepasswd* expects an OTP. To avoid a chicken and egg problem, users must declare (by setting the *-c* option) that they are working at the secure console (see Line 8 in Listing 1).

If the command catches the user lying, it will refuse to cooperate. Users who take security seriously should avoid the *-f* option (Line 17), which ignores the subsequent warning.

The process is user-specific; in other words, any user wanting to work with one-time passwords needs to run the command individually.

After completing the initialization, a user entry is added to the */etc/opiekeys* file. This file also contains the seed (*te5843* in this case), the hash (*6f1dba738c197a64*), the newly-generated one-time password, and the sequence number (499 in this example – Lines 31 and 32).

To generate valid one-time passwords later, users need their own password, the seed, and the sequence number. There is no need to memorize all this – with the exception of the user password. The other two credentials are provided and displayed by the server.

Safety Net

The next step is to integrate the authentication mechanism with the PAM stack (see Listing 2). The *pam_unix* or *pam_unix2* modules do most of the work. These modules are tagged with a *sufficient* control flag, but as you want to replace the *pam_unix.so* or *pam_unix2.so* libraries with the *pam_opie.so* library, you should modify the configuration accordingly.

Note that it is possible to configure your system so that, if OPIE fails for any reason, users can still use legacy passwords to authenticate.

Once you have modified the PAM configuration, your system is OTP-capable. Some services, such as the SSH daemon, still need some manual attention before they start using one-time passwords. In the case of SSH, you need the following line in the server configuration file */etc/sshd/sshd_config*:

Listing 2: pam_opie.so

```
01 ...
02 auth    sufficient    pam_opie.so
03 # You can leave out this line if
   you have tested OPIE:
04 auth    sufficient    pam_unix.so
   nullok try_first_pass
05 ...
```

Listing 3: SSH Login with OPIE

```
01 $ ssh root@rechner.example.com
02 otp-md5 498 te5843 ext
03 Response:
04 # cat /etc/opiekeys
05 root 0498 te5843 2b84befd37cacb9f
   Feb 16,2008 05:58
06 #
```

Understanding S/Key and OTP

A one-time password system consists of a server and a generator. Users are required to authenticate against the server, and the generator calculates the one-time password for this purpose. The mathematical underpinnings are provided by hash functions or irreversible algorithms: S/Key uses MD4, and OTP uses MD4, MD5, and SHA. The algorithm ensures that an attacker cannot deduce the next password just by gaining knowledge of its predecessor.

Users need to initialize the OTP system on the server side by choosing a password. The server appends a random, or user-defined, seed to the password string (Figure 3) and hashes the resulting string *n* times to generate the first one-time password. Finally, the server stores the username, the seed, the figure *n*, and the OTP.

A user wanting to authenticate against the server is sent a challenge including the seed and the figure n-1 (Figure 4). The local generator helps the user calculate a one-time password. This calculation is basically the same as the server-side initialization phase, the difference being that the hash is only run n-1 times.

The user sends the results to the server, which then hashes the incoming string once more and compares the results with the one-time password it has stored. If the two hashes match, everything is okay; the server stores the OTP passed in to it, instead of the original OTP, and decrements n by one.

OTP-managed passwords are 64-bit values from a technical point of view, however, users can enter them in the form of short words. A program would convert an entry such as *TUSK JOIN ROBE HUNK HAVE CARL* to the internal bit representation.

With OTP providing the cryptographic underpinnings, it is just a question of integrating this framework with the various authentication programs on Linux. These programs include *login* and *sudo*, session managers such as *xdm*, *kdm*, and *gdm*, or external services such as the SSH daemon or FTP servers. Linux uses Pluggable Authentication Modules (PAM, [5]) to provide a standardized interface.

If you use one-time passwords to authenticate, you need to add a line to the *auth* section of your PAM configuration. The required control flag depends on the configuration of your *auth* stack and the desired system behavior.

Listing 4: Creating three OTPs with opiekey

```
01 # opieinfo
02 497 te5843
03 # opiekey -5 -n 3 `opieinfo`
04 Using the MD5 algorithm to compute response.
05 Reminder: Don't use opiekey from telnet or dial-in
06 sessions. Sorry, but you don't seem to be on the
07 console or a secure terminal.
08 Warning: Continuing could disclose your secret pass
09 phrase to an attacker!
10 Enter secret pass phrase:
11 495: MUSH ACT GRIM SEE MAID LIES
12 496: HAD FED WORD ROY STAB ACID
13 497: IO INK RIG DAME RULE TUM
14 #
```

```
ChallengeResponseAuthentication
yes
```

Listing 3 shows an SSH login using OPIE. After successfully authenticating, OPIE updates the */etc/opiekeys* file, adding the new sequence number and the hash of the last password used.

Sowing and Harvesting

Users need *opiekey* to generate one-time passwords. The generator in Listing 4 expects the user password, the seed, and the current sequence number. Users can run *opieinfo* to view this information. OPIE also has a mechanism that generates a list of OTPs in case a user doesn't have a generator.

Other generators in addition to *opiekey* are also available. The Java program JOTP [6] will run on a Java-capable cellphone or on a normal website, although the website must be trustworthy. Palm owners can run Palmkey [7] or Pilotp [8], and desktop users can run Optcalc [9].

The *opiepasswd -d* command disables a user entry in */etc/opiekeys* and thus bans the user from the OPIE system (see Listing 5). The system overwrites the password hash with a series of asterisks (*), although the sequence number and seed remain visible.

The OTPW Alternative

The OTPW software-based solution does not use the method specified by RFC 2289 but relies instead on a 160-bit version

Listing 5: Disabling OPIE for a User

```
01 user1@rechner$ opiepasswd -d
02 Updating user1:
03 Disable user1's OTP access? (yes or no) yes
04 ID user1 is disabled.
05 user1@rechner$ su -
06 Passwort:
07 # grep user1 /etc/opiekeys
08 user1 0359 te2880 ****************  Feb 16,2008 08:37
09 #
```


Join us on
18 november
at the international press center in Brussels

Open source ERP systems

ProFOSS
For Professional users of Free & Open Source Software

Profoss event

Learn from Case studies, Meet with professionals
Discuss with users, developers and providers

www.profoss.eu info@profoss.eu

Sponsors
Microsoft
hp invent
trustable

Media Partners
it professional
LINUX MAGAZINE

Partners
Sun microsystems
zeropoint.it
@wt.be Agence Wallonne des Télécommunications
Cluster TIC INFOPOLE
DMoon.be
O'REILLY
ALUNYS
Internet Society BELGIUM

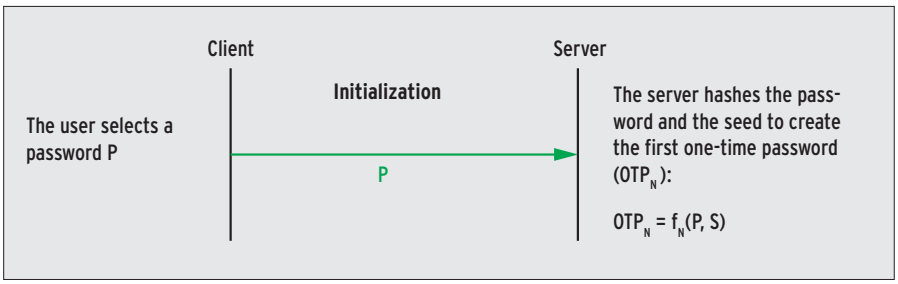

Figure 3: To initialize, the generator sends a password to the server. The server hashes the password and a seed to calculate the first one-time password.

of the RIPEMD hash. OTPW includes a modified version of the program login (*demologin*) and an alternative module for integration with the PAM stack. Users are issued passwords in the form of a list, which is similar to the legacy TAN lists issued by banks.

When authenticating, the user types a string comprising the list entry and their own password. The OTPW server stores the RIPEMD hashes of all valid one-time passwords (along with a number) in the *.otpw* file below the user's home directory. The program overwrites used passwords with dashes, thus preventing reuse.

The OTPW package is far smaller than OPIE; the source code comprises just 18 files. A simple *make* will create the *demologin* and *otpw-gen* programs, as well as the *pam_otpw.so* PAM library.

For Linux systems with PAM, OTPW requires only the *otpw-gen* generator and the *pam_otpw* module. The user initializes the OTPW system by running *otpw-gen* (Listing 6). After entering a password, *otpw-gen* creates a list of OTPs and displays the results.

Listing 6: Setting up OTPW

```
01 # otpw-gen -h 5
02 Generating random seed ...
03
04 If your paper password list is stolen, the thief should not gain
05 access to your account with this information alone. Therefore, you
06 need to memorize and enter below a prefix password. You will have to
07 enter that each time directly before entering the one-time password
08 (on the same line).
09
10 When you log in, a 3-digit password number will be displayed.  It
11 identifies the one-time password on your list that you have to append
12 to the prefix password. If another login to your account is in progress
13 at the same time, several password numbers may be shown and all
14 corresponding passwords have to be appended after the prefix
15 password. Best generate a new password list w en you have used up half
16 of the old one.
17
18 Enter new prefix password:
19 Reenter prefix password:
20
21 Creating '~/.otpw'.
22 Generating new one-time passwords ...
23
24 OTPW list generated 2008-03-16 10:23 on testvm3.seidelnet.de
25
26 000 a7Sj rWoC  001 %URK VvmD  002 EoQa sgon  003 IQhJ kVMG  004 QsS% H=aU
27
28           !!! REMEMBER: Enter the PREFIX PASSWORD first !!!
28 #
```

The *-p1* parameter tells *otpw-gen* to output the OTPs as a list of four-letter words, for example:

```
hare lane fyfe self lucy
```

Deleting the *.otpw* file disables the use of one-time passwords for the account.

It makes sense to print the list. Users are responsible for keeping track of how many valid one-time passwords they still have.

If you want to save paper, check the content of *.otpw* when you log in. Used OTPs are tagged with -. Integration of OTPW with the PAM system follows the same steps as for OPIE.

According to the documentation, adding this entry

```
session  optional  pam_otpw.so
```

tells OTPW to let you know how many OTPs you have left when you log in. This command did not work in our lab. The manual steps for the SSH daemon are similar to those for OPIE.

Users create one-time passwords by concatenating their user passwords with

Pluggable Authentication Modules (PAM)

PAM defines four categories for the authentication process: *auth*, *account*, *password*, and *session*. The *auth* category handles the authentication itself, while *password* defines whether and how a user can change their password. PAM uses *account* to manage access based on the user account and *session* to handle the environment setup.

PAM has a selection of various modules in each category and organizes them in a stack. Each module is tagged with a control flag. This approach lets admins define how PAM reacts to successful or unsuccessful processing of a module. The following flags exist: *required*, *requisite*, *sufficient*, and *optional*. If a module flagged *required*, *requisite*, or *sufficient* fails, the complete authentication process fails. If the module is tagged *requisite*, PAM immediately stops processing the stack.

After successfully processing a module flagged *required*, *requisite*, or *optional*, the next PAM library steps up. PAM views the category as successfully processed if the module is flagged *sufficient*.

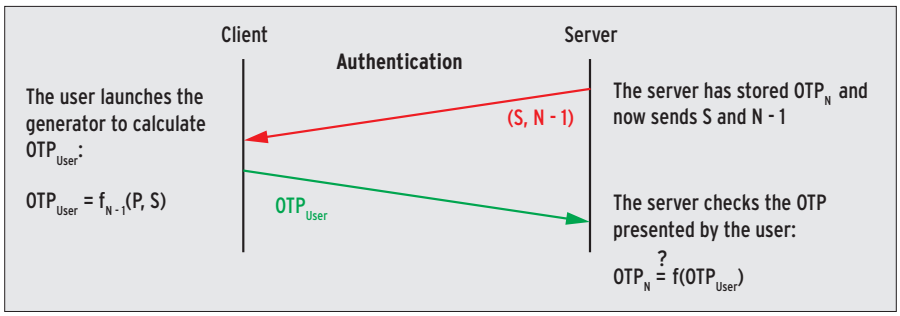

Figure 4: During authentication, the server presents the seed and a counter. The user runs a generator to calculate the one-time password and returns it to the server for validation.

the strings in the list generated by *otpw-gen*.

When a user attempts to log in, OTPW creates a symbolic link for *.otpw.lock* in the user's home directory. If the user cancels the login attempt by pressing Ctrl + C, the symbolic link is kept. The user is locked out while the link exists, as it prevents the use of OTPW.

On top of this, OTPW does not normally support simultaneous logins for security reasons. According to the program documentation, the user enters an extended one-time password in this case. The extended OTP comprises the user password and three strings from the list. We were unable to test this behavior in our lab.

THE AUTHOR

Dr. Udo Seidel is math and physics teacher who has been a big Linux fan since 1996. Since completing his PhD, he has worked as a Linux/Unix trainer, system administrator, and senior solution engineer. Today, he heads the Linux/Unix team at Amadeus Data Processing GmbH in Erding, Germany.

INFO

[1] Digipass 300 Pro: *http://www.vasco.com*

[2] RSA Secur ID: *http://www.rsa.com*

[3] Onetime Password In Everything (OPIE): *http://www.inner.net/opie*

[4] One-Time Password (OTPW): *http://www.cl.cam.ac.uk/~mgk25/otpw.html*

[5] Pluggable Authentication Modules (PAM): *http://www.kernel.org/pub/linux/libs/pam/*

[6] Java OTP Calculator (JOTP): *http://www.cs.umd.edu/~harry/jotp/*

[7] Palmkey: *http://palmkey.sf.net*

[8] Pilot OTP Generator: *http://www.valdes.us/palm/pilOTP/*

[9] OTP and S/Key Calculator for X-Window: *http://killa.net/infosec/otpCalc/*

One-time passwords are useful for insecure environments with a danger of password sniffing. The OPIE and OTPW implementations are easily integrated with popular Linux distributions thanks to PAM. ■

symbian smartphoneshow
21-22 OCTOBER 2008
EARLS COURT 2, LONDON
Innovation in action

See you at the show!

Now in its tenth year, the Smartphone Show continues to bring together the leaders and visionaries shaping the future of mobile.

Keynote speakers include:

Nigel Clifford, CEO, Symbian
Mats Lindoff, CTO, Sony Ericsson
Rob Shaddock, Corporate Vice President, Motorola
Kai Öistämö, Executive Vice President, Devices, Nokia
Benoit Schillings, CTO, Trolltech
Mr Ho-Soo Lee, EVP of Mobile Solutions Center, Samsung
David Wood, EVP Research, Symbian (moderator)

Gold Sponsor: SAMSUNG
Silver Sponsors: accenture, LG, Tele Atlas
Bronze Sponsors: Forum NOKIA, freescale, MOTOROLA, NAVTEQ, orange Partner, UIQ TECHNOLOGY

■ 4000 delegates ■ 120 exhibitors ■ 60 seminars ■ 10 keynotes ■ one show

Show features

FREE keynotes and seminars. Access to industry leaders and the opportunity to learn more in smaller focused sessions.

Mobile DevFest. The launch of Symbian's premier developer conference, now incorporated within the smartphone show.

FREE developer training. At mobile devfest - get up to speed with new technologies in hands-on labs run by Symbian experts.

Largest exhibition. 2008 will host the largest exhibition ever seen at the smartphone show.

Product demos and showcases. Don't miss the chance to experience the hottest pre-release demos, newest handsets and the latest industry news.

Symbian party. Party with us on 21 October! Celebrate the show's success and network with other attendees in an informal environment.

Register NOW online - Reserve your place at the Smartphone Show online at smartphoneshow.com
Use the booking code: CQ1905ADLM

smartphoneshow.com