Software Engineering Institute
Carnegie Mellon University

1989 CERT Advisories

CERT Division

http://www.sei.cmu.edu

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

This report was prepared for the

SEI Administrative Agent
AFLCMC/AZS
5 Eglin Street
Hanscom AFB, MA 01731-2100

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM17-0052

Table of Contents

1 CA-1989-01: Passwd Hole

Original issue date: January 1989
Last revised: September 16, 1997
Attached Copyright statement

A complete revision history is at the end of this file.

The CERT center received the following information from Keith Bostic from the Computer Systems Research Group at UC-Berkeley on Dec. 21, 1988. This patch has also been posted to comp.bugs.4bsd.ucb-fixes.

Please note that this patch will only work with BSD 4.3. If you have 4.2 please let me know and I will forward the correct patch.

Subject: security problem in passwd
Index: bin/passwd.c 4.3BSD
Description:
There's a security problem associated with the *passwd(1)* program in all known Berkeley systems. This problem is also in most Berkeley derived systems, see your vendor for more information.

Fix: Apply the following patch to the file src/bin/passwd.c and recompile/reinstall it.

```
*** passwd.c.orig       Wed Dec 21 08:57:41 1988
- --- passwd.c  Wed Dec 21 09:00:25 1988
***************
*** 332,337 ****
- --- 332,339 ----
        return (crypt(pwbuf, saltc));
  }
+ #define       STRSIZE 100
+
  char *
  getloginshell(pwd, u, arg)
        struct passwd *pwd;
***************
*** 338,344 ****
        int u;
        char *arg;
  {
!       static char newshell[BUFSIZ];
        char *cp, *valid, *getusershell();
        if (pwd->pw_shell == 0 || *pwd->pw_shell == '\0')
- --- 340,346 ----
        int u;
        char *arg;
  {
!       static char newshell[STRSIZE];
```

```
        char *cp, *valid, *getusershell();
        if (pwd->pw_shell == 0 || *pwd->pw_shell == '\0')
***************
*** 415,423 ****
  getfingerinfo(pwd)
        struct passwd *pwd;
  {
!       char in_str[BUFSIZ];
        struct default_values *defaults, *get_defaults();
!       static char answer[4*BUFSIZ];
        answer[0] = '\0';
        defaults = get_defaults(pwd->pw_gecos);
- --- 417,425 ----
  getfingerinfo(pwd)
        struct passwd *pwd;
  {
!       char in_str[STRSIZE];
        struct default_values *defaults, *get_defaults();
!       static char answer[4*STRSIZE];

        answer[0] = '\0';
        defaults = get_defaults(pwd->pw_gecos);
***************
*** 429,435 ****
         */
        do {
                printf("\nName [%s]: ", defaults->name);
!               (void) fgets(in_str, BUFSIZ, stdin);
                if (special_case(in_str, defaults->name))
                        break;
        } while (illegal_input(in_str));
- --- 431,437 ----
         */
        do {
                printf("\nName [%s]: ", defaults->name);
!               (void) fgets(in_str, STRSIZE, stdin);
                if (special_case(in_str, defaults->name))
                        break;
        } while (illegal_input(in_str));
***************
*** 440,446 ****
        do {
                printf("Room number (Exs: 597E or 197C) [%s]: ",
                        defaults->office_num);
!               (void) fgets(in_str, BUFSIZ, stdin);
                if (special_case(in_str, defaults->office_num))
                        break;
        } while (illegal_input(in_str) || illegal_building(in_str));
- --- 442,448 ----
        do {
                printf("Room number (Exs: 597E or 197C) [%s]: ",
                        defaults->office_num);
```

```
!               (void) fgets(in_str, STRSIZE, stdin);
                if (special_case(in_str, defaults->office_num))
                        break;
        } while (illegal_input(in_str) || illegal_building(in_str));
***************
*** 452,458 ****
        do {
                printf("Office Phone (Ex: 6426000) [%s]: ",
                        defaults->office_phone);
!               (void) fgets(in_str, BUFSIZ, stdin);
                if (special_case(in_str, defaults->office_phone))
                        break;
                remove_hyphens(in_str);
- --- 454,460 ----
        do {
                printf("Office Phone (Ex: 6426000) [%s]: ",
                        defaults->office_phone);
!               (void) fgets(in_str, STRSIZE, stdin);
                if (special_case(in_str, defaults->office_phone))
                        break;
                remove_hyphens(in_str);
***************
*** 464,470 ****
         */
        do {
                printf("Home Phone (Ex: 9875432) [%s]: ", defaults->home_phone);
!               (void) fgets(in_str, BUFSIZ, stdin);
                if (special_case(in_str, defaults->home_phone))
                        break;
                remove_hyphens(in_str);
- --- 466,472 ----
         */
        do {
                printf("Home Phone (Ex: 9875432) [%s]: ", defaults->home_phone);
!               (void) fgets(in_str, STRSIZE, stdin);
                if (special_case(in_str, defaults->home_phone))
                        break;
                remove_hyphens(in_str);
***************
*** 501,507 ****
        if (input_str[length-1] != '\n') {
                /* the newline and the '\0' eat up two characters */
                printf("Maximum number of characters allowed is %d\n",
!                       BUFSIZ-2);
                /* flush the rest of the input line */
                while (getchar() != '\n')
                        /* void */;
- --- 503,509 ----
        if (input_str[length-1] != '\n') {
                /* the newline and the '\0' eat up two characters */
                printf("Maximum number of characters allowed is %d\n",
!                       STRSIZE-2);
```

```
                /* flush the rest of the input line */
                while (getchar() != '\n')
                        /* void */;
```

Revision History

```
September 16, 1997  Attached copyright statement
```

2 CA-1989-02: Sun Restore Hole

Original issue date: July 26, 1989
Last revised: September 16, 1997
Attached copyright statement

A complete revision history is at the end of this file.

A security hole has been found in SunOS restore. This problem affects SunOS 4.0, 4.0.1, and 4.0.3 systems. It does not appear in SunOS 3.5. The problem occurs because restore is setuid to root. Without going into details, is sufficient to say that this is a serious hole. All SunOS 4.0 installations should install this workaround. Note that a user does need to have an existing account to exploit this hole.

There are two workarounds that will fix the problem. The first is slightly more secure but has some side-effects.

1. Make restore non-setuid by becoming root and doing a
2. `chmod 750 /usr/etc/restore`

   This makes restore non-setuid and unreadable and unexecutable by ordinary users.

   Making restore non-setuid affects the restore command using a remote tape drive. You will no longer be able to run a restore from another machine as an ordinary user; instead, you'll have be root to do so. (The reason for this is that the remote tape drive daemon on the machine with the tape drive expects a request on a TCP privileged port. Under SunOS, you can't get a privileged port unless you are root. By making restore non-setuid, when you run restore and request a remote tape drive, restore won't be able to get a privileged port, so the remote tape drive daemon won't talk to it.)

3. If you do need to have some users run restore from remote tape drives without being root, you can use the following workaround.

   ```
   cd /usr/etc
   chgrp operator restore
   chmod 4550 restore
   ```

   This allows the use of restore by some trusted group. In this case, we used the group 'operator', but you may substitute any other group that you trust with access to the tape drive. Thus, restore is still setuid and vulnerable, but only to the people in the trusted group.

   The 4550 makes restore readable and executable by the group you specified, and unreadable by everyone else.

   Sun knows about this problem (Sun Bug 1019265) and will put in a more permanent fix in a future release of SunOS.

This document is available from: http://www.cert.org/advisories/CA-1989-02.html

CERT/CC Contact Information

Email: cert@cert.org
Phone: +1 412-268-7090 (24-hour hotline)
Fax: +1 412-268-6989
Postal address:

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
U.S.A.

CERT/CC personnel answer the hotline 08:00-17:00 EST(GMT-5) / EDT(GMT-4) Monday through Friday; they are on call for emergencies during other hours, on U.S. holidays, and on weekends.

Using encryption

We strongly urge you to encrypt sensitive information sent by email. Our public PGP key is available from http://www.cert.org/CERT_PGP.key.

If you prefer to use DES, please call the CERT hotline for more information.

Getting security information

CERT publications and other security information are available from our web site: http://www.cert.org/.

* "CERT" and "CERT Coordination Center" are registered in the U.S. Patent and Trademark Office.

NO WARRANTY
Any material furnished by Carnegie Mellon University and the Software Engineering Institute is furnished on an "as is" basis. Carnegie Mellon University makes no warranties of any kind, either expressed or implied as to any matter including, but not limited to, warranty of fitness for a particular purpose or merchantability, exclusivity or results obtained from use of the material. Carnegie Mellon University does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

Conditions for use, disclaimers, and sponsorship information

Revision History

```
September 16, 1997  Attached copyright statement
```

3 CA-1989-03: Telnet Break-in Warning

Original issue date: August 16, 1989
Last revised: September 16, 1997
Attached copyright statement

A complete revision history is at the end of this file.

Many computers connected to the Internet have recently experienced unauthorized system activity. Investigation shows that the activity has occurred for several months and is spreading. Several UNIX computers have had their "telnet" programs illicitly replaced with versions of "telnet" which log outgoing login sessions (including usernames and passwords to remote systems). It appears that access has been gained to many of the machines which have appeared in some of these session logs. (As a first step, frequent telnet users should change their passwords immediately.) While there is no cause for panic, there are a number of things that system administrators can do to detect whether the security on their machines has been compromised using this approach and to tighten security on their systems where necessary. At a minimum, all UNIX site administrators should do the following:

- Test telnet for unauthorized changes by using the UNIX "strings" command to search for path/filenames of possible log files. Affected sites have noticed that their telnet programs were logging information in user accounts under directory names such as "..." and ".mail".

  In general, we suggest that site administrators be attentive to configuration management issues. These include the following:

- Test authenticity of critical programs - Any program with access to the network (e.g., the TCP/IP suite) or with access to usernames and passwords should be periodically tested for unauthorized changes. Such a test can be done by comparing checksums of on-line copies of these programs to checksums of original copies. (Checksums can be calculated with the UNIX "sum" command.) Alternatively, these programs can be periodically reloaded from original tapes.
- Privileged programs - Programs that grant privileges to users (e.g., setuid root programs/shells in UNIX) can be exploited to gain unrestricted access to systems. System administrators should watch for such programs being placed in places such as /tmp and /usr/tmp (on UNIX systems). A common malicious practice is to place a setuid shell (sh or csh) in the /tmp directory, thus creating a "back door" whereby any user can gain privileged system access.
- Monitor system logs - System access logs should be periodically scanned (e.g., via UNIX "last" command) for suspicious or unlikely system activity.
- Terminal servers - Terminal servers with unrestricted network access (that is, terminal servers which allow users to connect to and from any system on the Internet) are frequently used to camouflage network connections, making it difficult to track unauthorized activity. Most popular terminal servers can be configured to restrict network access to and from local hosts.

- Passwords - Guest accounts and accounts with trivial passwords (e.g., username=password, password=none) are common targets. System administrators should make sure that all accounts are password protected and encourage users to use acceptable passwords as well as to change their passwords periodically, as a general practice. For more information on passwords, see Federal Information Processing Standard Publication (FIPS PUB) 112, available from the National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161.
- Anonymous file transfer - Unrestricted file transfer access to a system can be exploited to obtain sensitive files such as the UNIX /etc/passwd file. If used, TFTP (Trivial File Transfer Protocol - which requires no username/password authentication) should always be configured to run as a non-privileged user and "chroot" to a file structure where the remote user cannot transfer the system /etc/passwd file. Anonymous FTP, too, should not allow the remote user to access this file, or any other critical system file. Configuring these facilities to "chroot" limits file access to a localized directory structure.
- Apply fixes - Many of the old "holes" in UNIX have been closed. Check with your vendor and install all of the latest fixes.

If system administrators do discover any unauthorized system activity, they are urged to contact the Computer Emergency Response Team (CERT).

This document is available from: http://www.cert.org/advisories/CA-1989-03.html

CERT/CC Contact Information

Email: cert@cert.org
Phone: +1 412-268-7090 (24-hour hotline)
Fax: +1 412-268-6989
Postal address:

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
U.S.A.

CERT/CC personnel answer the hotline 08:00-17:00 EST(GMT-5) / EDT(GMT-4) Monday through Friday; they are on call for emergencies during other hours, on U.S. holidays, and on weekends.

Using encryption

We strongly urge you to encrypt sensitive information sent by email. Our public PGP key is available from http://www.cert.org/CERT_PGP.key.

If you prefer to use DES, please call the CERT hotline for more information.

Getting security information

CERT publications and other security information are available from our web site: http://www.cert.org/.

* "CERT" and "CERT Coordination Center" are registered in the U.S. Patent and Trademark Office.

NO WARRANTY
Any material furnished by Carnegie Mellon University and the Software Engineering Institute is furnished on an "as is" basis. Carnegie Mellon University makes no warranties of any kind, either expressed or implied as to any matter including, but not limited to, warranty of fitness for a particular purpose or merchantability, exclusivity or results obtained from use of the material. Carnegie Mellon University does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

Conditions for use, disclaimers, and sponsorship information

Revision History

```
September 16,1997  Attached copyright statement
```

4 CA-1989-04: WANK Worm On SPAN Network

Original issue date: October 17, 1989
Last revised: September 17, 1997
Attached copyright statement

A complete revision history is at the end of this file.

On 16 October, the CERT received word from SPAN network control that a worm was attacking SPAN VAX/VMS systems. This worm affects only DEC VMS systems and is propagated via DECnet protocols, not TCP/IP protocols. If a VMS system had other network connections, the worm was not programmed to take advantage of those connections. The worm is very similar to last year's HI.COM (or Father Christmas) worm.

This is NOT A PRANK. Serious security holes are left open by this worm. The worm takes advantage of poor password management, modifies .com files, creates a new account, and spreads to other systems via DECnet.

It is also important to understand that someone in the future could launch this worm on any DECnet based network. Many copies of the virus have been mailed around. Anyone running a DECnet network should be warned.

R. Kevin Oberman from Lawrence Livermore National Labs reports: "This is a mean bug to kill and could have done a lot of damage. Since it notifies (by mail) someone of each successful penetration and leaves a trapdoor (the FIELD account), just killing the bug is not adequate. You must go in an make sure all accounts have passwords and that the passwords are not the same as the account name."

The CERT/CC also suggests checking every .com file on the system. The worm appends code to .com files which will reopen a security hole everytime the program is executed.

An analysis of the worm appears below and is provided by R. Kevin Oberman of Lawrence Livermore National Laboratory. Included with the analysis is a DCL program that will block the current version of the worm. At least two versions of this worm exist and more may be created. This program should give you enough time to close up obvious security holes.

If you have any technical questions or have an infected system, please call the CERT/CC:

Report on the W.COM worm.

R. Kevin Oberman
Engineering Department
Lawrence Livermore National Laboratory
October 16, 1989

The following describes the action of the W.COM worm (currently based on the examination of the first two incarnations). The replication technique causes the code to be modified slightly which indicates the source of the attack and learned information.

All analysis was done with more haste than I care for, but I believe I have all of the basic facts correct.

First a description of the program:

1. The program assures that it is working in a directory to which the owner (itself) has full access (Read, Write,Execute, and Delete).
2. The program checks to see if another copy is still running. It looks for a process with the first 5 characters of "NETW_".

   If such is found, it deletes itself (the file) and stops its process.

   NOTE: A quick check for infection is to look for a process name starting with "NETW_". This may be done with a SHOW PROCESS command.

3. The program then changes the default DECNET account password to a random string of at least 12 characters.
4. Information on the password used to access the system is mailed to the user GEMPAK on SPAN node 6.59. Some versions may have a different address.
5. The process changes its name to "NETW_" followed by a random number.
6. It then checks to see if it has SYSNAM priv.

   If so, it defines the system announcement message to be the banner in the program:

```
      W O R M S    A G A I N S T    N U C L E A R    K I L L E R S

_______________________________________________________________
    \__  ____________  _____    ________    ____  ____   __
_____/
     \ \ \    /\    / /    / /\ \       | \ \  | |    | | / /    /
      \ \ \  /  \  / /    / /__\ \      | |\ \ | |    | |/ /    /
       \ \ \/ /\ \/ /    / ______ \     | | \ \| |    | |\ \   /
        \_\  /__\  /____/ /______\ \____| |__\ | |____| |_\ \_/
         \___________________________________________________/
          \                                                 /
           \    Your System Has Been Officically WANKed    /
            \_____________________________________________/
     You talk of times of peace for all, and then prepare for war.
```

7. If it has SYSPRV, it disables mail to the SYSTEM account.

8. If it has SYSPRV, it modifies the system login command procedure to APPEAR to delete all of a user's file. (It really does nothing.)
9. The program then scans the accounts logical name table for command procedures and tries to modify the FIELD account to a known password with login form any source and all privs.

   This is a primitive virus, but very effective IF it should get into a privileged account.

10. It proceeds to attempt to access other systems by picking node numbers at random.

    It then used PHONE to get a list of active users on the remote system. It proceeds to irritate them by using PHONE to ring them.

11. The program then tries to access the RIGHTSLIST file and attempts to access some remote system using the users found and a list of "standard" users included with the worm.

    It looks for passwords which are the same as that of the account or are blank. It records all such accounts.

12. It looks for an account that has access to SYSUAF.DAT.
13. If a priv. account is found, the program is copied to that account and started. If no priv account was found, it is copied to other accounts found on the random system.
14. As soon as it finishes with a system, it picks another random system and repeats (forever).

Response:

1. The following program will block the worm. Extract the following code and execute it. It will use minimal resources.

   It create a process named NETW_BLOCK which will prevent the worm from running.

   Editors note: This fix will work only with this version of the worm. Mutated worms will require modification of this code; however, this program should prevent the worm from running long enough to secure your system from the worms attacks.

```
$ Set Default SYS$MANAGER
$ Create BLOCK_WORM.COM
$ DECK/DOLLAR=END_BLOCK
$LOOP:
$ Set Process/Name=NETW_BLOCK
$ Wait 12:0
$ GoTo loop
END_BLOCK
$ Run/In-
put=SYS$MANAGER:BLOCK_WORM.COM/Error=NL:/Output=NL:/UIC=[1,4] -
    SYS$SYSTEM:LOGINOUT
```

   Editors note: This fix might only work if the worm is running as SYSTEM. An earlier post made by the CERT/CC suggested the following:

```
        $ Run SYS$SYSTEM:NCP
        Clear Object Task All
        ^Z
```

You must then edit the file SYS$MANAGER:STARTNET.COM, and add the line

```
CLEAR OBJECT TASK ALL
```

AFTER the line which says

```
SET KNOWN OBJECTS ALL
```

This has the side-effect of disabling users from executing any command procedure via DECnet that the system manager has not defined in the DECnet permanent database.

2. Enable security auditing.

   The following command turns on the MINIMUM alarms. The log is very useful in detecting the effects of the virus left by the worm. It will catch the viruses modification of the UAF. $ Set Audit/Alarm/Enable=(ACL,Authorization,Breakin=All,Logfailure=All)

3. Check for any account with NETWORK access available for blank passwords or passwords that are the same as the username. Change them!
4. If you are running VMS V5.x, get a copy of SYS$UPDATE:NETCONFIG_UPDATE.COM from any V5.2 system and run it.

   If you are running V4.x, change the username and password for the network object "FAL".

5. If you have been infected, it will be VERY obvious.

   Start checking the system for modifications to the FIELD account. Also, start scanning the system for the virus. Any file modified will contain the following line:
   $ oldsyso=f$trnlnm("SYS$OUTPUT")

   It may be in LOTS of command procedures. Until all copies of the virus are eliminated, the FIELD account may be changed again.

6. Once you are sure all of the holes are plugged, you might kill off NETW_BLOCK. (And then again, maybe not.)

This document is available from: http://www.cert.org/advisories/CA-1989-04.html

CERT/CC Contact Information

Email: cert@cert.org
Phone: +1 412-268-7090 (24-hour hotline)
Fax: +1 412-268-6989
Postal address:

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
U.S.A.

CERT/CC personnel answer the hotline 08:00-17:00 EST(GMT-5) / EDT(GMT-4) Monday through Friday; they are on call for emergencies during other hours, on U.S. holidays, and on weekends.

Using encryption

We strongly urge you to encrypt sensitive information sent by email. Our public PGP key is available from http://www.cert.org/CERT_PGP.key.

If you prefer to use DES, please call the CERT hotline for more information.

Getting security information

CERT publications and other security information are available from our web site http://www.cert.org/.

* "CERT" and "CERT Coordination Center" are registered in the U.S. Patent and Trademark Office.

NO WARRANTY
Any material furnished by Carnegie Mellon University and the Software Engineering Institute is furnished on an "as is" basis. Carnegie Mellon University makes no warranties of any kind, either expressed or implied as to any matter including, but not limited to, warranty of fitness for a particular purpose or merchantability, exclusivity or results obtained from use of the material. Carnegie Mellon University does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

Conditions for use, disclaimers, and sponsorship information

Revision History

```
September 17,1997  Attached Copyright Statement
```

5 CA-1989-05: DEC/Ultrix 3.0 Systems

Original issue date: October 17, 1989
Last revised: September 17, 1997
Attached copyright statement

A complete revision history is at the end of this file.

Recently, the CERT/CC has been working with several Unix sites that have experienced breakins. Running tftpd, accounts with guessable passwords or no passwords, and known security holes not being patched have been the bulk of the problems.

The intruder, once in, gains root access and replaces key programs with ones that create log files which contain accounts and passwords in clear text. The intruder then returns and collects the file. By using accounts which are trusted on other systems the intruder then installs replacement programs which start logging.

There have been many postings about the problem from several other net users. In addition to looking for setuid root programs in users' home directories, hidden directories '.. ' (dot dot space space), and a modified telnet program, we have received two reports from Ultrix 3.0 sites that the intruders are replacing the /usr/bin/login program. The Ultrix security hole being used in these attacks is only found in Ultrix 3.0.

Suggested steps:

1. Check for a bogus /usr/bin/login.

   The sum program reports:

   ```
   27379    67  for VAX/Ultrix 3.0
   ```

2. Check for a bogus /usr/etc/telnetd.

   The sum program reports:

   ```
   23552    47  for VAX/Ultrix 3.0
   ```

3. Look for .savacct in either /usr/etc or in users' directories.

   This may be the file that the new login program creates. It could have a different name on your system.

4. Upgrade to Ultrix 3.1 ASAP.
5. Monitor accounts for users having passwords that can be found in the /usr/dict/words file or have simple passwords like a persons name or their account name.
6. Search through the file system for programs that are setuid root.
7. Disable or modify the tftpd program so that anonymous access to the file system is prevented.

If you find that a system that has been broken into, changing the password on the compromised account is not sufficient. The intruders do remove copies of the /etc/passwd file in order to break the remaining passwords. It is best to change all of the passwords at one time. This will prevent the intruders from using another account.

Please alert CERT if you do find a problem.

This document is available from: http://www.cert.org/advisories/CA-1989-05.html

CERT/CC Contact Information

Email: cert@cert.org
Phone: +1 412-268-7090 (24-hour hotline)
Fax: +1 412-268-6989
Postal address:

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
U.S.A.

CERT/CC personnel answer the hotline 08:00-17:00 EST(GMT-5) / EDT(GMT-4) Monday through Friday; they are on call for emergencies during other hours, on U.S. holidays, and on weekends.

Using encryption

We strongly urge you to encrypt sensitive information sent by email. Our public PGP key is available from http://www.cert.org/CERT_PGP.key.

If you prefer to use DES, please call the CERT hotline for more information.

Getting security information

CERT publications and other security information are available from our web site: http://www.cert.org/.

* "CERT" and "CERT Coordination Center" are registered in the U.S. Patent and Trademark Office.

NO WARRANTY
Any material furnished by Carnegie Mellon University and the Software Engineering Institute is furnished on an "as is" basis. Carnegie Mellon University makes no warranties of any kind, either expressed or implied as to any matter including, but not limited to, warranty of fitness for a particular purpose or merchantability, exclusivity or results obtained from use of the material. Carnegie Mellon University does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

Conditions for use, disclaimers, and sponsorship information.

Revision History

```
September 17, 1997  Attached copyright statement
```

6 CA-1989-06: DEC/Ultrix 3.0 Systems

Original issue date: October 18, 1989
Last revised: September 17, 1997
Attached copyright statement

A complete revision history is at the end of this file.

This is a repost of the Ultrix 3.0 advisory. We have received the sum output for DECstations.

Recently, the CERT/CC has been working with several Unix sites that have experienced breakins. Running tftpd, accounts with guessable passwords or no passwords, and known security holes not being patched have been the bulk of the problems.

The intruder, once in, gains root access and replaces key programs with ones that create log files which contain accounts and passwords in clear text. The intruder then returns and collects the file. By using accounts which are trusted on other systems the intruder then installs replacement programs which start logging.

There have been many postings about the problem from several other net users. In addition to looking for setuid root programs in users' home directories, hidden directories '.. ' (dot dot space space), and a modified telnet program, we have received two reports from Ultrix 3.0 sites that the intruders are replacing the /usr/bin/login program. The Ultrix security hole being used in these attacks is only found in Ultrix 3.0.

Suggested steps:

1. Check for a bogus /usr/bin/login.

   The sum program should report the following for the DEC supplied login program.

   ```
   27379    67    for VAXstation Ultrix 3.0

                  35559   116     for DECstation Ultrix 3.0
   ```

2. Check for a bogus /usr/etc/telnetd.

   The sum program should report the following for the DEC supplied telnetd program.

   ```
   23552    47    for VAXstation Ultrix 3.0

                  45355    84     for DECstation Ultrix 3.0
   ```

3. Look for .savacct in either /usr/etc or in users' directories.

   This may be the file that the new login program creates. It could have a different name on your system.

4. Upgrade to Ultrix 3.1 ASAP.
5. Monitor accounts for users having passwords that can be found in the /usr/dict/words file or have simple passwords like a persons name or their account name.
6. Search through the file system for programs that are setuid root.
7. Disable or modify the tftpd program so that anonymous access to the file system is prevented.

If you find that a system that has been broken into, changing the password on the compromised account is not sufficient. The intruders do remove copies of the /etc/passwd file in order to break the remaining passwords. It is best to change all of the passwords at one time. This will prevent the intruders from using another account.

Please alert CERT if you do find a problem.

Thank you,

Computer Emergency Response Team (CERT)

This document is available from: http://www.cert.org/advisories/CA-1989-06.html

CERT/CC Contact Information

Email: cert@cert.org
Phone: +1 412-268-7090 (24-hour hotline)
Fax: +1 412-268-6989
Postal address:

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
U.S.A.

CERT personnel answer the hotline 08:00-17:00 EST(GMT-5) / EDT(GMT-4) Monday through Friday; they are on call for emergencies during other hours, on U.S. holidays, and on weekends.

Using encryption

We strongly urge you to encrypt sensitive information sent by email. Our public PGP key is available from http://www.cert.org/CERT_PGP.key.

If you prefer to use DES, please call the CERT hotline for more information.

Getting security information

CERT publications and other security information are available from our web site: http://www.cert.org/.

To subscribe to the CERT mailing list for advisories and bulletins, send email to majordomo@cert.org. Please include in the body of your message

```
subscribe cert-advisory
```

* "CERT" and "CERT Coordination Center" are registered in the U.S. Patent and Trademark Office.

NO WARRANTY
Any material furnished by Carnegie Mellon University and the Software Engineering Institute is furnished on an "as is" basis. Carnegie Mellon University makes no warranties of any kind, either expressed or implied as to any matter including, but not limited to, warranty of fitness for a particular purpose or merchantability, exclusivity or results obtained from use of the material. Carnegie Mellon University does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

Conditions for use, disclaimers, and sponsorship information

Revision History

September 17,1997 Attached copyright statement

7 CA-1989-07: Sun RCP Vulnerability

Original issue date: October 26, 1989
Last revised: September 17, 1997
Attached copyright statement

A complete revision history is at the end of this file.

A problem has been discovered in the SunOS 4.0.x rcp. If exploited, this problem can allow users of other trusted machines to execute root-privilege commands on a Sun via rcp.

This affects only SunOS 4.0.x systems; 3.5 systems are not affected.

A Sun running 4.0.x rcp can be exploited by any other trusted host listed in /etc/hosts.equiv or /.rhosts. Note that the other machine exploiting this hole does not have to be running Unix; this vulnerability can be exploited by a PC running PC/NFS, for example.

This bug will be fixed by Sun in version 4.1 (Sun Bug number 1017314), but for now the following workaround is suggested by Sun:

Change the 'nobody' /etc/passwd file entry from

```
nobody:*:-2:-2::/:
```

to

```
nobody:*:32767:32767:Mismatched NFS ID's:/nonexistant:/nosuchshell
```

If you need further information about this problem, please contact CERT/CC by electronic mail or phone.

This document is available from: http://www.cert.org/advisories/CA-1989-07.html

CERT/CC Contact Information

Email: cert@cert.org
Phone: +1 412-268-7090 (24-hour hotline)
Fax: +1 412-268-6989
Postal address:

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
U.S.A.

CERT/CC personnel answer the hotline 08:00-17:00 EST(GMT-5) / EDT(GMT-4) Monday through Friday; they are on call for emergencies during other hours, on U.S. holidays, and on weekends.

Using encryption

We strongly urge you to encrypt sensitive information sent by email. Our public PGP key is available from http://www.cert.org/CERT_PGP.key.

If you prefer to use DES, please call the CERT hotline for more information.

Getting security information

CERT publications and other security information are available from our web site:
http://www.cert.org/

* "CERT" and "CERT Coordination Center" are registered in the U.S. Patent and Trademark Office.

NO WARRANTY
Any material furnished by Carnegie Mellon University and the Software Engineering Institute is furnished on an "as is" basis. Carnegie Mellon University makes no warranties of any kind, either expressed or implied as to any matter including, but not limited to, warranty of fitness for a particular purpose or merchantability, exclusivity or results obtained from use of the material. Carnegie Mellon University does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

Conditions for use, disclaimers, and sponsorship information

Revision History

September 17, 1997 Attached Copyright Statement