Das Sicherheitsloch in VAX/VMS Version 4.4/4.5

Ereignisse um einen Betriebssystemfehler

Stefan Weirauch

Karlsruhe, im Dezember 1987

(Electronic Mail: weirauch@iravcl.ira.uka.de)

Inhalt
---

0. Vorwort

Ein paar persönliche Worte vorweg:
VAX/VMS ist ein ausgereiftes, über viele Jahre gewachsenes, multifunktionales Betriebssystem. Diese Multifunktionaltät bringt eine überdurchschnittliche Selbstverantwortung des Systemmanagers bezüglich der Systemsicherheit mit sich. Relativ zu dieser Tatsache betrachtet, kann man VMS als eines der sichersten Betriebssysteme werten.

Dieses Dokument soll die Vorgänge und Vorfälle bezüglich des Betriebssystemfehlers in VMS 4.4/4.5 zusammenfassen. Damit sollen die Darstellungen in der Presse je nachdem bestätigt, korrigiert bzw. widerlegt werden.
Es soll aber auch diejenigen Wissenshungrigen zufriedenstellen, deren Neugier durch Presseerklärungen von DEC nicht gestillt werden konnte.

Nicht weiter ausgeführt werden soll hier der sogenannte NASA-Hack, soweit er nicht in direktem Zusammenhang mit dem Fehler steht. Als zu müßig muß der Versuch angesehen werden in dieser Sache auf all die ungenauen, verfälschten oder einfach komplett falschen Darstellungen in den Medien einzugehen.

Auch kein Ziel dieser Dokumentation soll es sein, DEC in ein schlechtes Licht zu stellen, dafür ist das, was die Entwicklungsabteilungen von DEC (Hard- wie Software) hervorbringen, einfach zu gut - vor ihnen habe ich einen großen Respekt, im Gegensatz zum Management und zu den Pressestellen (s.u.).

Schließlich bezweckt diese Zusammenfassung auch, die abschließenden Wünsche an DEC verständlich zu machen.

1. Reaktionen auf einen Betriebssystemfehler

1.1. Erste Erwähnungen in öffentlichen Medien

Zuerst wurde in halböffentlichen Fachkreisen über ein Sicherheitsloch in VMS gemunkelt. Ein bedeutendes Medium dieser Fachwelt ist beispielsweise INFO-VAX - ein Electronic-Mail Verbund, dem VAX-Betreiber aus allen Nationen mit Schwerpunkt USA angeschlossen sind. Hier erschien als erster den Fehler betreffenden Hinweis folgende Mail:

```
>   Date: Wed, 6 May 87 10:39 EDT
>   From: welch%UMASS.BITNET@wiscvm.wisc.edu
>   Subject: VMS Security hole?
>
>   I'm posting this for a co-worker who just returned from DECUS:
>   I've heard (at DECUS) that there is a "GLARING" security hole
>   in VMS v4.5. The problem apparently is that a user can aquire
>   very high privs. if they do the correct series of command (or
>   something like that).  The message on VAXNOTES (at DECUS) that
>   detailed this was erased by DEC almost as soon as it was
>   submitted.
>   At DECUS all DEC would say is "DEC will not comment on security
>   problems in any release of VMS.  If you think that you are having
>   security problem contact your support person."  This is annoying.
>      If there is a way for someone to bypass security I want to know
>   about it, preferably from DEC and not from notes on COMPUSERVE.
>      If anyone knows if this "HOLE" exists please let the net know.
>   BUT PLEASE DO NOT PUT THE DETAILS HERE.  The last thing we need is
>   for the whole world to know how to do it.
```

Damit wurde eine Lawine von diesbezüglichen Mails ausgelöst, hier eine kleine Auswahl:

```
>   Date: Mon, 11 May 87 11:22 EDT
>   From: Robert M. Gerber <RGERBER%NYBVX1.BITNET@wiscvm.wisc.edu>
>   Subject: Re: VMS Security Hole (Mack Truck Size).................
>
>   In response to JWELCH@UMASS.BITNET:
>   Yes that security hole does exist,
>   yes DEC knows very much about it.
>   And it's large enuf to drive a Mack Truck through it.
>   Where I am currently an ex-DEC employee just started. He gave
>   himselfpriv's.  He had just TMPMBX & NETMBX to start with.....
>   ------Robert Gerber     These opions...etc,.....

>   Date: Mon, 11 May 87 11:54 EDT
>   From: Robert M. Gerber <RGERBER%NYBVX1.BITNET@wiscvm.wisc.edu>
>   Subject: Re: VMS Security Hole (Mack Truck Size)......
>
>   I just talked to DEC Customer Support Center/Colorado Springs...
>   The problem is in only in VMS Versions 4.4 & 4.5 and they have
>   a patch.
>   To break in all you need is an ID and possibly TMPMBX priv.  '
>   This is all I will say on this problem...
>   Call DEC.....................
>   -----Robert
```

Einige Mails spiegelten eine gewisse Hilflosigkeit wider, aber immerhin wurde die Botschaft verbreitet, DEC habe einen Patch (ein 'Fehlerausmerzungsprogramm'). Viele VAX - Betreiber waren dennoch verständlichweise ungeduldig, und so schrieb schließlich einer von ihnen:

> Date: 7 Jun 87 23:09:27 EDT
> From: *Hobbit* <AWalker@red.rutgers.edu>
> Subject: security patch
>
> Why doesn't someone just *post* the Patch input file to infovax, if DEC is dragging their heels so? If there are explanatory comments detailing the security hole in the header of this, you could even strip those out leaving just the patch data, if it'd make you feel better.
> What are the legal ramifications of this? DEC would seem to have some sort of responsibility here, and if they've sold people a "secure" OS that every high school kid is now cruising freely around in the middle of, well, hey.

Nur kurz darauf wurde dann tatsächlich ein Patch in Form einer Kommandodatei über INFO-VAX verschickt. Dieser Patch enthielt nun aber relativ genaue Kommentare (s. 2.2.). Daraufhin wurde die öffentliche Diskussion weniger kommentierend und hitziger.

1.2. Öffentliche Diskussion

Etwa 4 Wochen nach den ersten Gerüchten wurde über die Umstände diskutiert, unter denen jemand nur den Fehler finden konnte:

> Date: Fri, 29 May 87 09:31 EDT
> From: DHASKIN%CLARKU.BITNET%wiscvm.wisc.edu@relay.cs.net
> Subject: VMS/uVMS mandatory security patch (now called IMPPAT010)
>
> ... Colorado did claim that (a) accomplishing this is *not* trivial -- one would really have to know VMS well, and (b) one really requires access to VMS source code to discover/accomplish it. ...

> Date: Tue, 9 Jun 87 23:22:00 MDT
> From: Ed Cetron <cetron%ced@cs.utah.edu>
> Subject: Re: security patch
>
> ...after reading it and the fiche it was quite obvious that if it could be reverse engineered, one of the following two conditions needed to be met:
> 1. the reverse engineer would need to have access to the fiche, in which case the fiche is actually sufficient WITHOUT the patch.
> 2. the guy is a true genius, and patch or no patch this guy has already hacked your system into submission. ...

Zu diesen Umständen bzw. Vorraussetzungen: siehe 2.1.

Auch diskutiert wurde über die Folgen der Publikation des Patches:

> Date: Thu, 11 Jun 87 09:45 CDT
> From: Dan Stewart <STEWART_SYS%uta.edu@relay.cs.net>
> Subject: Security patch.
>
> ... Now with the patch published, potential hackers have been given a very good clue as to where to dig. ...

1.3. Kundenbetreuung von DEC

Schon bei einem ähnlich gravierenden Fehler in der VMS Version 4.2 zeigte sich DEC sehr verschlossen - es gab keine Informationen selbst in der Folgeversion 4.3, die (im Gegensatz zur Version 4.6 Nachfolger von 4.4/4.5) relativ schnell distributiert werden konnt wurde nicht darauf hingewiesen, daß ein sehr ernstes Sicherheitsl ausgemerzt war. Die Politik bei Fehlern gleich welcher Art sieht also so aus: Der Kunde wird nicht informiert, auf konkrete Nachfragen, bekommt er eventuell einen provisorischen, kommentarlosen Update.
Der Grund für diese Verschwiegenheit scheint plausibel, die Bekanntmachung eines Fehlers wird verhindert, man gewinnt Zeit bis eine neue Version oder ein einzelner Update verteilt werden kann.
Warum versagte dieses Prinzip in diesem Jahr ?
Es kann nur unter folgenden Bedingungen funktionieren:

1.: Ein Fehler wird frühzeitig erkannt und eine entsprechende Korrektur in Form eines Updates entwickelt.
2.: Dieser Update (d.h. eine neue Betriebssystemversion oder lediglich ein Update des fehlerhaften Teiles) kann schnell an alle VAX-Betreiber (nicht nur direkte DEC Kunden) verteilt werden.

zu 1.: Wann und wer von DEC den Fehler zuerst kannte ist nicht belegbar.
Begonnen wurde die Entwicklung der Korrektur (IMPPAT 010) erst im Dezember 1986, der Fehler war bereits 9 Monate al
zu 2.: Obwohl der Patch Ende Januar 1987 im wesentlichen und mit Sicherheit nur kurz danach vollständig entwickelt war, begann die Auslieferung erst Ende Mai 1987.

Unter diesen Umständen mußte das Prinzip natürlich kläglich scheitern. Einen einfach zu entdeckenden Fehler, der die Integrit des Systems ernsthaft bedroht, so zurückhaltend zu behandeln, muß als grob fahrlässig angesehen werden.

DEC wurde von dieser Sache überrollt und hatte erhebliche Probleme seine Kunden mit dem Patch zu versorgen (s. 1.5.), außerdem wurden die Kunden nicht auf die Notwendigkeit hingewiese diesen Patch einzufahren, obwohl die DEC-Techniker wenigstens ein pro Monat bei den meisten Installationen vor Ort sind.

Einige System Manager erhielten den Patch dann Ende Mai:

> Date: Wed, 20 May 87 08:58:17 SET
> From: KA%DDAESA10.BITNET@wiscvm.wisc.edu
> Subject: VMS Security Hole
>
> I would like to inform our European VMS users that DEC in
> Germany and Holland have the patch that is needed to plug this
> security hole.
> I received it this week from TSC in Munich and it is rather
> short. There was no explanation as to why it was needed and it wa
> described as an "unofficial" patch, the "official" one being made
> available with VMS 4.6.
> No doubt, the other European offices have it too.
> Jenny Franks,
> European Space Operations Centre,

Tatsache jedoch ist, daß ihn die meisten erst viel später erhalten haben; z.B.:

> Date: Mon, 29-JUN-1987 15:58 +0200
> To: info-vax@kl.sri.com
> Subject: the infamous SECURESHR patch
>
> I just received DEC's official mandatory update (it finally made
> it to Germany) and noted the existence of an ECO 6 which has not
> been published over the net.
> I just want to direct yout attention to this ECO since it fixes
> (guess what) another hole, this time however only applicable to
> hackers with GRPPRV. ...
>
> W.J.Moeller, GWDG, D-3400 Goettingen, F.R.Germany

Viele haben den Patch noch sehr viel später erhalten; zugute halten muß man DEC hier allerdings, daß mittlerweile auch alle die VAX-Betreiber kostenlos versorgt wurden, die keinen Software-Wartungsvertrag mit DEC haben oder sogar nicht einmal direkt Kunden von DEC sind.

1.4. Eine Konsequenz des Fehlers - der NASA-Hack

Die Reaktion auf einen derartigen Fehler seitens eines Hackers kann nur seine konsequente Ausnutzung beim Eindringen in VMS-Systeme sein.
Folgerichtig machte sich also eine Gruppe junger Computer-Enthusiasten im Frühling 1987 auf, das größte unkommerzielle DECNet der Welt SPAN (Space Physics Analysis Network) heimzusuchen, und dort diesen Fehler auszunutzen. So nebenbei wurden dabei auch rund 20 Rechner der NASA 'geknackt'. So bekam die Geschichte den Beinamen 'NASA-Hack', wodurch sie sich natürlich besser verkaufen ließ. Tatsache ist jedoch, daß diese 20 der insgesamt etwa 135 besiegten Systeme eher zu den uninteressanteren gehörten. Viele Forschungsinstitute, die u.a. SDI Forschung betreiben, sind an SPAN angeschlossen - daß es bei der NASA keine geheimen Informationen zu holen gab, mag wohl stimmen, an SPAN sind jedoch einige Wölfe im Schafspelz angeschlossen.
Daß weder hier noch bei der NASA vandaliert, zerstört oder geraubt wurde, belegt die Grundeinstellung jener Hacker, denen es nicht auf irgend einen Profit ankommt, sondern neben einer gewissen persönlichen Genugtuung, auf die Darlegung der Schwächen von verteilten Computersystemen.
Wer hier von Computer-Terroristen spricht, der müßte Schülerlotsen Verkehrsraudis schimpfen, und der möchte vermutlich von eigener Schuld ablenken, die durch diese Groß-Demonstration zutage trat.

An dieser Stelle sei nur noch an einen Ausspruch eines der kompetentesten EDV-Juristen Deutschlands erinnert.
Prof. Dr. Ulrich Sieber bestätigte nämlich," ...daß sich einzelne der Hacker hier sogar Verdienste erworben haben..." .
(Panorama, ARD, 15.09.87)

1.5. DEC Stellungnahmen

FAZ, 15.09.87:
Bezüglich des NASA-Hacks:
"Ein Sprecher der deutschen Niederlassung von Digital Equipment sagte in München auf Anfrage, ihm sei von diesen Vorgängen nichts bekannt."

Panorama, 15.09.87:
"Auf konkrete Fragen nach Systemfehlern blieb die Computerfirma Digital eine konkrete Antwort schuldig und zog sich auf eine allgemeine Bekräftigung ihrer Sicherheitsvorkehrungen zurück."

Computerwoche v. 25. 09. 87:
Klaus Kemmler (Leiter Produktmarketing, DEC München):
"Allein die Dokumentation für VMS umfaßt rund 500000 Seiten; das entspricht etwa 20 Megabyte an reinem Code. Bei einem Betriebssystem mit diesem Umfang stößt man nicht durch Zufall auf solch einen Fehler - da waren VMS-Experten am Werk."
(siehe dazu Abschnitt 2.1.)

DECUS Bulletin Nr. 35, Nov. 87:
(DEC München) :
"Wir sind das erste Mal mit einem solchen Fall konfrontiert worden und waren deshalb den logistischen Problemen nicht gewachsen.

2. Der Fehler

2.1. Wie man in ein Sicherheitsloch fällt

Entgegen allen Aussagen über die Schwierigkeit, den Fehler zu entdecken, selbst ohne Patch und ohne Micro-Fiche, sei hier die Vorgehensweise eines normalen VAX Benutzers dargestellt, der aus reiner Neugier, die mögliche Existenz eines Sicherheitslochs nicht ausschließend, sich ein wenig im System umschaut.
Und dies ist bereits ab Frühling 1986 möglich gewesen - solange existierte der Fehler bereits.

VMS HELP, bietet manigfaltige Information über das gesamte Betriebssystem und seine Utilities, ein HELP Begriff ist z.B.

NewFeatures_V44

Die Systemsicherheit betreffend findet man dort:

```
>    New and Changed Features for Version 4.4
>
>    ...
>
>     o  Security --- New features include a new DCL command, SET
>        RIGHTS_LIST and a new attribute, DYNAMIC.  SET RIGHTS_LIST
>        adds and removes identifiers from the process and system
>        rights list.  You can assign the DYNAMIC attribute to
>        identifiers to enable nonprivileged users to add or remove
>        identifiers they hold from their process rights list.  For
>        more information on changes to the security system services,
>        see the New and Changed Features section of the VAX/VMS System
>        Services Reference Manual.
>
>     ...
>
>     o  System Services --- $CHECK_ACCESS, $GETUAI, and $SETUAI are
>        new services.  See the New and Changed Features section of the
>        VAX/VMS System Services Reference Manual.
>
>    ...
```

Mit $SETUAI soll ein entsprechend privilegierter Benutzer aus einem eigenen Programm heraus Einträge im UAF (User - Authorization - File) modifizieren können. Darüber gibt HELP auch genauere Informationen:

```
>    $SETUAI
>      The Set User Authorization Information ($SETUAI) service is used
>      to modify the user authorization file (UAF) record for a
>      specified user.
>
>         Format:
>
>       SYS$SETUAI  [nullarg] ,[nullarg] ,usrnam ,itmlst ,[nullarg]
>                  ,[nullarg]
>
>         Arguments:
>
>      nullarg
>      ...
>      Place-holding argument.  This argument is reserved to DIGITAL.
>
>      usrnam
>      ...
>      Name of the user whose user authorization file (UAF) record is
>      modified.  The usrnam argument is the address of a descriptor
>      pointing to a character text string containing the user name. Th
>      user name string may contain a maximum of 12 alphanumeric
>      characters.
>
>      itmlst
>      ...
>      Item list specifying which information from the specified user's
>      UAF (user authorization file) record is to be modified. The itml
>      argument is the address of a list of one or more item descriptor
>      each of which specifies an item code.  The item list is terminat
>      by an item code of 0 or by a longword of 0.
```

Man beachte die 'null-arguments'.

VAX PASCAL Programmierern ist die Datei SYS$LIBRARY:STARLET.PAS nicht unbekannt, es ist die Quelldatei für ein in eigene Programme einbindbares Modul, das Deklarationen aller System-Services sowie benötigter Konstanten enthält.
Man schaut dort gelegentlich nach, wie die Deklaration aussieht, damit man den System-Services auch die richtigen Parameter übergib
SYS$SETUAI ist dort nicht nur widersprüchlich zum $SETUAI HELP deklariert, sondern noch zusätzlich recht interessant kommentiert:

```
>    (* $SETUAI                                                          *
>    (*                                                                 *
>    (*   Modify User Authorization Information                         *
>    (*                                                                 *
>    (*    $SETUAI  [efn] ,[contxt] ,usrnam ,itmlst ,[iosb] ,[astadr]   *
>    (*      ,[astprm]                                                  *
>    (*                                                                 *
>    (*    efn    = event flag to be set at completion                 *
>    (*                                                                 *
>    (*    contxt = address of a context longword (UAF IFI & ISI)       *
>    (*                                                                 *
>    (*    usrnam = address of user name descriptor                     *
>    (*                                                                 *
>    (*    itmlst = address of a list of item descriptors               *
>    (*                                                                 *
>    (*    iosb   = address of a quadword I/O status block              *
>    (*                                                                 *
>    (*    astadr = address of entry mask of AST routine                *
>    (*                                                                 *
>    (*    astprm = value to be passed to AST routine                   *
>    (*                                                                 *
>    (*                                                                 *
>
>    [ASYNCHRONOUS,EXTERNAL(SYS$SETUAI)] FUNCTION $SETUAI (
>        %IMMED EFN : UNSIGNED := %IMMED 0;
>        %REF CONTXT : UNSIGNED := %IMMED 0;
>        USRNAM : [CLASS_S] PACKED ARRAY [$13..$u3:INTEGER] OF CHAR;
>        %REF ITMLST : [UNSAFE] ARRAY [$14..$u4:INTEGER] OF $UBYTE;
>        VAR IOSB : [VOLATILE]$UQUAD := %IMMED 0;
>        %IMMED [UNBOUND, ASYNCHRONOUS] PROCEDURE ASTADR := %IMMED 0;
>        %IMMED ASTPRM : UNSIGNED := %IMMED 0) : INTEGER; EXTERNAL;
```

Ich will hier vorwegnehmen, daß an dieser Stelle sogar mehr Parameter kommentiert sind, als wirklich existieren.
Interessant mag dies einem in RMS (Record Management System - erlaubt selbst in Assembler unkomplizierte Dateibearbeitung) versierten Programmierer vorkommen, denn er weiß, daß IFI (Interna File Identifier) und ISI (Internal Stream Identifier) die internen Dateireferenzen sind, die die Verbindung zur Datei bei Schreib- un Leseoperationen nach dem Öffnen der Datei darstellen.
Als unprivilegierter Benutzer versucht er nun diese Routine zu benutzen und bekommt erwartungsgemäß eine Fehlermeldung - er habe nicht die nötigen Privilegien, darüber hinaus bekommt er dann aber, vielleicht zu seiner Überraschung Werte über den CONTXT Parameter von der Routine zurück.
Und ohne viel 'Trial & Error' wird er herausfinden, was man mit diesen Werten anfangen kann.
Dazu mehr im folgenden Abschnitt.

So schnell findet man also einen Fehler, wofür man doch eigentlich Micro-Fiches oder zumindest den Patch braucht !

2.2. Der INFO-VAX Patch und der System-Service SYS$SETUAI

Anhand, des in INFO-VAX veröffentlichten Patch's soll hier näher die vorgesehene aber auch die wirkliche Funktionsweise von SYS$SETUAI beschrieben werden.

Die Zahlen in Klammern deuten auf nachfolgende Kommentare.

```
>   Date: Sun, 7 Jun 87 23:22:00 MDT
>   From: Ed Cetron <cetron%ced@cs.utah.edu>
>   To: AWalker@red.rutgers.edu, info-vax@kl.sri.com  (1)
>   Subject: Re:  security patch
>
>   Digital TSC (2) has indicated that the security patch should be
>   disseminated as widely as possible so here it is. As usual, neithe
>   I nor the CED nor the Univ of Utah take any responsibility for the
>   patch after the network mail systems do their damndest....
>   as well as all the rest of the standard disclaimers...
>
>   this patch was correct, and worked, and passed the checksum before
>   i mailed it.
>
>       -ed
>
>   The command file below is the patch for the security problem
>   discussed at DECUS.  You must be running VMS V4.5 (3).  Instructio
>   for applying it are:
>
>       1)  Place in file SYS$COMMON:[SYSUPD]SECURESHR.PAT as is.
>       If you edit it, it will not pass checksum checks.
>
>       2)  Execute @SYS$COMMON:[SYSUPD]SECURESHR.PAT.
>
>   (4) 3)  Either re-boot OR as I did, run SYS$SYSTEM:INSTALL and
>       REPLACE SYS$SHARE:SECURESHR.EXE.  This is the image that is
>       patched.
>
>   (5)
>   $ CHECKSUM SECURESHR.PAT
>   $ X='CHECKSUM$CHECKSUM'
>   $ IF X.NE.%X652628B1 THEN GOTO IC ! 652628B1
>   $ ON WARNING THEN EXIT
>   $ SET DEFAULT SYS$COMMON:[SYSUPD]
>   $ COPY SYS$COMMON:[SYSLIB]SECURESHR.EXE SECURESHR.EXE
>   $ PATCH/JOURNAL=SECURESHR/OUTPUT=SECURESHR SECURESHR
>   !   ECO05   LMPxxxx     23-Jan-1987          (6)
>   !       MODULE: SYSUAISRV
>   !       Additional tweaks to ECO04.
>   !
>   !   ECO04   LMP0429     14-Jan-1987
>   !       MODULE: SYSUAISRV
>   !       Minor tweaks to ECO03.  Also, tweaks to GRPPRV handling.
>   !
>   !   ECO03   LMP0424     16-Dec-1986
>   !       MODULE: SYSUAISRV
>   !       Properly handle the context field.
>
```

```
>    DEFINE GETUAI=7C40      (7)
>    DEFINE SETUAI=7C40+37C
>
>    SET ECO 03       (8)
>
>    REP/INS GETUAI+1B3
>    '   BLSSU   GETUAI+212'
>    EXIT
>    '   BRB     GETUAI+212'
>    EXIT
>
>    REP/INS SETUAI+1BD
>    '   BLSSU   SETUAI+21D'
>    EXIT
>    '   BRB     SETUAI+21D'
>    EXIT
>    UPDATE
>
>    SET ECO 04       (9)
>
>    REP/INS GETUAI+86
>    '   BLSSU   GETUAI+99'
>    EXIT
>    '   BRB     GETUAI+99'
>    EXIT
>
>    REP/INS SETUAI+81
>    '   BLSSU   SETUAI+96'
>    EXIT
>    '   BRB     SETUAI+96'
>    EXIT
>
>    REP/INS GETUAI+295
>    '   BBS     #2,B^0D4(FP),GETUAI+2C2'
>    EXIT
>    '   BBC     #2,B^0D4(FP),GETUAI+2A5'
>    EXIT
>
>    REP/INS SETUAI+2DC
>    '   BBS     #2,B^0D4(FP),SETUAI+303'
>    EXIT
>    '   BBC     #2,B^0D4(FP),SETUAI+2ED'
>    EXIT
>    UPDATE
>
>    SET ECO 05       (10)
>
```

```
>    REP/INS SETUAI+314
>    '    MOVL     #24,R0'
>    '    RET'
>    EXIT
>    '    MOVL     #24,(SP)'
>    '    BRW      SETUAI+50B'
>    EXIT
>
>    REP/INS SETUAI+329
>    '    MOVZWL   #291C,R0'
>    '    RET'
>    EXIT
>    '    MOVZWL   #291C,(SP)'
>    '    BRW      SETUAI+50B'
>    EXIT
>
>    REP/INS SETUAI+386
>    '    MOVL     #14,R0'
>    '    RET'
>    EXIT
>    '    MOVL     #14,(SP)'
>    '    BRW      SETUAI+50B'
>    EXIT
>
>    REP/INS SETUAI+3A0
>    '    MOVZWL   #290C,R0'
>    '    RET'
>    EXIT
>    '    MOVZWL   #290C,(SP)'
>    '    BRW      SETUAI+50B'
>    EXIT
>
>    REP/INS SETUAI+3AA
>    '    MOVZWL   #2914,R0'
>    '    RET'
>    EXIT
>    '    MOVZWL   #2914,(SP)'
>    '    BRW      SETUAI+50B'
>    EXIT
>
>    REP/INS SETUAI+471
>    '    MOVZWL   #28E4,R0'
>    '    RET'
>    EXIT
>    '    MOVZWL   #28E4,(SP)'
>    '    BRW      SETUAI+50B'
>    EXIT
>
>    REP/INS SETUAI+4D3
>    '    MOVL     #0C,R0'
>    '    RET'
>    EXIT
>    '    MOVL     #0C,(SP)'
>    '    BRW      SETUAI+50B'
>    EXIT
>    UPDATE  (11)
>
>    EXIT
>    $ COPY SECURESHR.EXE SYS$COMMON:[SYSLIB]SECURESHR.EXE
>    $ DELETE SECURESHR.EXE.*
>    $ EXIT
>    $ IC:WRITE SYS$OUTPUT "INCORRECT CHECKSUM; VERIFY CONTENTS OF FILE
```

Kommentare zur Info-Vax Mail von Ed Cetron bzgl. SECURESHR-Patch (7.6.1987)

(1) : Verteilung in den Listen 'Security' und 'Info-Vax'
(2) : TSC = Telephone Support Center (Online Software Support)
(3) : Patch ebenfalls auf VMS 4.4 anwendbar
(4) : Reboot unnötig und sinnlos, nach Anwendung der Prozedur
Ausführen folgender DCL Kommandos ausreichend:

```
$ install := $install/command
$ install replace sys$share:secureshr    ! reinstallieren
```

(5) : Die ab hier zu extrahierende Kommando Prozedur 'pacht' das vorhandene Image SYS$SHARE:SECURESHR.EXE und erzeugt davon eine neue Version
(6) : Ab hier eigentliches Patch Kommando File
ECO05, ECO04, ECO03 bezeichnen die einzelnen Patch Update Level
ECO = Engineering Change Order
(7) : Basisadressen der Routinen SYS$GETUAI und SYS$SETUAI innerhalb Routinen - Bibliothek SECURESHR.EXE
(8) : Routinen SETUAI und GETUAI sind sehr ähnlich aufgebaut, daher fast gleiche Fehler und deren Beseitigung, GETUAI öffnet SYSUAF.DA (SYStem User Authorization File, charakterisiert jeden Benutzer, insb. seine Rechte) nur zum lesen, d.h. eigentliche 'Bug' - Ausnutzung nur mit SETUAI.
ECO 03: Behandlung des CONTXT Parameters wird übersprungen, dami verschwindet dieser Parameter ganz.
Durch ihn wurden dem aufrufenden Programm zwei Werte übergeben (interne Datei Referenzen), durch die Zugriff auf eine Datei mögli ist, ohne sie selbst öffnen zu müssen. Unter VMS wirken Datei-Schu Mechanismen nur beim Öffnen einer Datei, d.h. hier: SYSUAF.DAT wird in den Routinen privilegiert geöffnet und Schreib-/Lesezugrif ist nun durch jene interne Datei Referenzen unprivilegiert möglich, solange die Datei geöffnet bleibt (s.(10)). Dieser privilegierte Zugriff seitens der Routinen (ermöglicht durch entsprechendes Installieren) ist kein Fehler, sondern zur ordnungs gemäßen Funktion notwendig.
So soll z.B. jeder unprivilegierte Benutzer eigene Daten aus SYSUAF.DAT mit Hilfe von SYS$GETUAI abrufen können, z.B. das Datum des eigenen letzten Einloggens u.a.; SYS$SETUAI soll Gruppen Management ermöglichen, d.h. ein 'Group Manager' (ein Benutzer, dem das Privileg GRPPRV zugeordnet ist) kann die SYSUAF Einträge einer abgegrenzten Benutzer Gruppe modifizieren; als Limit gelten dabei die Einträge des Group Managers selbst, somit kann er keinem Benutzer aus seiner Gruppe mehr Rechte einräumen, als ihm selbst zustehen.
(9) : ECO 04 : Ausmerzung eines logischen Fehlers in der Privilegie Überprüfung. Durch ihn reichte bereits GRPPRV - Privileg um auf jeden beliebigen Benutzereintrag schreibend bzw. lesend zuzugreife
(10) : ECO 05 : (betrifft nur SYS$SETUAI) Bei jedem Ausstieg aus d Routine als Folge eines Fehlers wurde der korrekte Fehler Code zurückgegeben, aber die Datei nicht geschlossen. Hier werden also die eintsprechenden Rücksprünge (RET) durch einen Sprung ersetzt, der zu einem Programmteil führt, in dem die Datei vor dem Rückspru geschlossen wird.
(11) : Durch den UPDATE Befehl übernimmt die PATCH Utility die Modifikationen in eine namensgleiche Datei mit höherer Versionsnummer.

Dieses Patch Kommando File ist NICHT identisch mit dem Patch, der in Form eines Magnetbandes von DEC distributiert wurde.

1.: Der distributierte Patch (namens IMPPAT 010) enthält einen weiteren ECO - Level (ECO 06), der einen weiteren (etwas unkritischeren) logischen Fehler beseitigt. Das Datum dieser Modifikation ist nicht bekannt, denn:

2.3. Der 'Mandatory Update Patch' IMPPAT 010

Bei uns an der Universität Karlsruhe ist besagter Patch nach telefonischer Anforderung und trotz Software-Wartungs-Vertrags erst Mitte Juni eingetroffen, zu diesem Zeitpunkt war bereits der INFO-VAX Patch angwandt worden, ein Zeichen dafür, daß vielerorts die Selbsthilfe der VAX-Betreiber schneller funktionierte als DEC's Kundenbetreuung - einer vieler Vorteile offener Netze.

Zu diesem Patch nun zusammenfassende Daten:

Entstehung:	Mitte Dez. 1986 - Ende Jan. 1987
Verfügbarkeit: vorab über TSC ab (nur auf Anfrage)	ca. Mai 1987
in INFO-VAX am (als Patch Command File)	8. Juni 1987
Distribution: als IMPPAT 010	18. Mai 1987 - Ende 1987 ?

Inhalt: SECURESHR.EXE;3 Creation Date: 6. Mai 1987
(per PATCH modifiziertes VMS 4.4 Image, ohne Patch-Text, identisch mit 4.6 Image)

Standard KITINSTAL
(d.h. bei Installieren des neuen images mit der Standard Prozedur VMSINSTAL wird völlig unnötig das System herunter gefahren (Reboot), dies mag ein Zeichen dafür sein, daß IMPPAT 010 relativ überstürzt zusammengebastelt wurde)

2.4. Die Suche nach der Ursache und Spekulationen

Das Loch basierte im wesentlichen auf zwei Fehlern:

1.: Über einen optionalen Parameter konnte man sich die interne Dateireferenzen auf die Datei SYSUAF.DAT zurüchgeben lassen.

2.: Bei einem beliebigen Fehlerausstieg aus der Routine blieb diese Datei geöffnet.

Ich habe mir den Quellcode von SYS$SETUAI und SYS$GETUAI genau angeschaut. Aus Rücksicht vor dem Copyright muß ich ihn hier leide fortlassen.
Nach eingehender Analyse läßt sich folgendes sagen:

Ein Programmierer mag bei der Fehlerbehandlung vergessen, an das Schließen einer Datei zu denken, aber ist es noch wahrscheinlich, daß er dies in schönster Konsequenz bei den sieben Fehlerausstiege in SYS$SETUAI vergißt ?
Nun, möglich mag auch das sein - aber gibt es einen Parameter nur aus Versehen ? Nein.
Schön - aus wissenschaftlicher Neugier möchte man natürlich erfahren, was denn jener CONTXT Parameter ursprünglich bezwecken sollte (als Ansatz um die wirkliche Ursache des Fehlers zu entdecken).

Ende August führte ich verschiedene Telefongespräche mit verschiedenen Stellen in verschiedenen DEC Vertretungen in Deutschland.
Nach erfolgloser Befragung des 'Telephone Support Centers' (der dortige 'Spezialist' erzählte mir lediglich etwas von 'null-arguments' - mehr gebe die Dokumentation nicht her...) wurde mir empfohlen, meiner Frage auf dem Wege des 'Software Performance Reports' (SPR) nachzugehen. Ein SPR ist so etwas wie ein Multifunktions-Mecker-Reklamier-Problemerklär-Vorschlag-Zettel für DEC Kunden.
Am 1. September schickte ich also so ein Formular ab.
Normalerweise erhält der Absender einige Tage später eine Bestätigung der lokalen DEC Geschäftsstelle und etwa 6 Wochen später die endgültige Antwort.
Nach etwa 8 Wochen hatte ich keines von beiden erhalten und hakte nach.
Etwa 3 Tage dauerte es, bis ich mit einer 'zuständigen Person' verbunden werden konnte. Deren Statement (sinngemäß):
In der Presse habe DEC ausreichend Stellung bezogen. (siehe 1.5.)
Es gab keinen Bruchteil einer Beantwortung - keine Begründung der ausbleibenden schriftlichen Reaktion auf meinen SPR, der dort (in München) vorlag, aber in keinster Weise bearbeitet wurde.

Inzwischen neigte sich der November dem Ende entgegen und unermüdlich versuchte ich weiterhin, telefonisch an kompetente Informationen zu gelangen. Die Mühe wurde leider nicht belohnt, d.h. vielleicht bis auf einen Ausspruch einer Angestellten der Münchner Vertretung: "...wenn ich Ihnen darüber etwas sage, dann ist hier die Hölle los..."

Wie auch immer, alles hatte der Anschein erweckt, daß es sich hier nicht einfach um einen peinlichen Software Fehler handelt, wie er immer und überall auftritt. Über seine Ursachen allerdings konnte dank der hervorragenden Informationspolitik von DEC weiterhin nur spekuliert werden.
Aufgrund der bekannten Fakten ist allerdings der Schluß zulässig:

Der Fehler hat sich nicht aus Versehen ins System geschlichen.

Warum existierte er dann ?
Folgende Begründungen erscheinen hier möglich:

- Die Funktionalität der System-Services sollte noch erweitert werden, jedoch wurden sie halbfertig ins System aufgenommen.

  Dafür spricht die programmtechnische Schlampigkeit und der 'überschüssige Parameter', dagegen, daß sich an der Funktionalität bis zum heutigen Tage nichts mehr geändert hat.

- Es sollte vorsätzlich ein Fehler eingebaut werden.

  Diese Möglichkeit erscheint mir persönlich sehr wahrscheinlich, denn es gibt dafür einige Argumente.

  Der Autor von SYS$SETUAI hat auch für den groben Fehler in VMS Version 4.2 gesorgt. Hat er den Auftrag, einer bestimmten Gruppe innerhalb oder außerhalb von DEC, das Hacken auf VMS-Vaxen zu erleichtern ?
  Hier sei daran erinnert, daß auch im Ostblock einige Vaxen stehen.

  Die System-Services SYS$SETUAI und SYS$GETUAI werden nicht vom Betriebsystem selbst genutzt, weiterhin sind sie nicht sehr sauber programmiert (z.b. führt ein falscher 'itemcode' zum Prozessabsturz) und können nur modifizieren/informieren, also keine neuen Benutzereinträge schaffen oder alte löschen.
  Zusammengenommen haben sie also keine große Funktionalität.
  Es kann nicht ausgeschlossen werden, daß der Hauptgrund für die Einrichtung dieser Routinen das Einschmuggeln jenes Fehlers war.

All diese Spekulationen lassen sich nach derzeitigem Erkenntnisstand nicht widerlegen, sie mögen kühn klingen, sind aber alle möglich, wenn nicht sogar wahrscheinlich.

3. Ausblicke

3.1. Der normale VAX-Betreiber

...tut gut daran, sich an vorhandene Medien (z.b. INFO-VAX) über offene Netze anzuschließen, um Informationslöcher zu stopfen, die durch eine restriktive Informationspolitik der Computerhersteller entstehen.
Selbsthilfe führt hier oft am schnellsten zum Erfolg.

3.2. Wümsche an DEC

An DEC sollen an dieser Stelle folgende Wünsche, Anregungen bzw. Empfehlungen gehen:

- Ausführliche Information der Kunden; nur wenn kriminellen Computer-Spezialisten (im Gegensatz zu Hackern !) ihr Wissensvorsprung genommen wird, kann der Kunde eigene individuelle Sicherheitsmaßnahmen ergreifen.

- Glaubwürdigere Pressearbeit, d.h. konsequentere Aufklärung vergangener Vorfälle, wodurch das Handeln des Unternehmens transparenter und kundennäher gemacht würde.

- Zusammenarbeit mit externer Kompetenz
  Damit meine ich kompetente Gruppen außerhalb des Unternehmens und außerhalb jener mit DEC kooperierenden Institute und Firmen. Nur wenn hier ein Aufeinanderzugehen angestrebt wird, kann eine konfrontative Situation vermieden werden, die letzt endlich beid Seiten, zumindest aber die des Unternehmens schaden könnte.