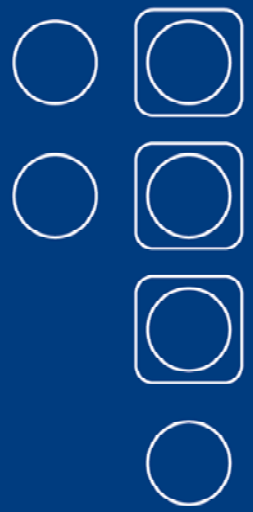
ITUG
SUMMIT
2004

How to Harden your NonStop Server A Security Show and Tell Part 1 of 3 - GUARDIAN

Carl Weber
GreenHouse Software & Consulting

06Oct2004, P-24-U, Marriott Salon 5



greenHouse

I T U G

The International HP NonStop Users Group
independent, not-for-profit, user run

Idea

- Cover ALL security aspects within ONE track
- Talk and show
- Demonstrate hot spots and present solutions

Session overview

- Part 1 of 3 – GUARDIAN
  Carl Weber (GreenHouse)

  - How secure is it?
  - Can be broken in? Easily?
  - Is there an easy way to prevent it?
  - Solutions!

Session overview

- Part 2 of 3 - OSS

  Roland Lemoine (HP)

  - Are we like Unix?

    10 common Unix security holes
  - OSS security features:

    Leverage Safeguard features for OSS
  - SSL - enable your middleware

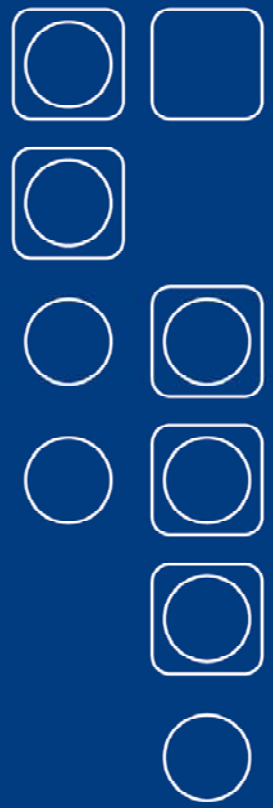
    (iTP Webserver, Java and WebLogic,etc...)

ITUG SUMMIT 2004

Session overview

- Part 3 of 3 - LAN
  - Thomas Burg (comForte)

  - TCP/IP: Extending the reach of NonStop security requirements
  - Are there only "script-kiddies" out there?
  - Why a firewall is not enough
  - Best practices in network security



greenHouse

ITUG

The International HP NonStop Users Group
independent, not-for-profit, user run

Brief intro Carl Weber

- Started with Tandem(*) Germany 1978
- 'In security' since 1985, when SAFEGUARD was introduced
- Started GreenHouse in 1994 as an Alliance Partner (www.GreenHouse.de)
- Specialized in Security consulting and reviews, product and tool development, PRIV system code, code specialties

(*) to me it still is Tandem ...

Well known truths

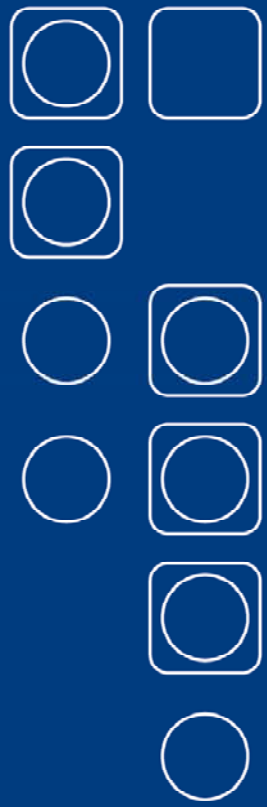
Ignorance doesn't solve the problem

... it just lets you sleep better...

Once you lost your integrity

... the rest is easy ...

Good judgment comes from experience.
Experience comes from bad judgment.



Well known truths

Everybody has his price

... trust me ...

In theory,
there is no difference between
theory and practice;
in practice, there is.

Chuck Reid

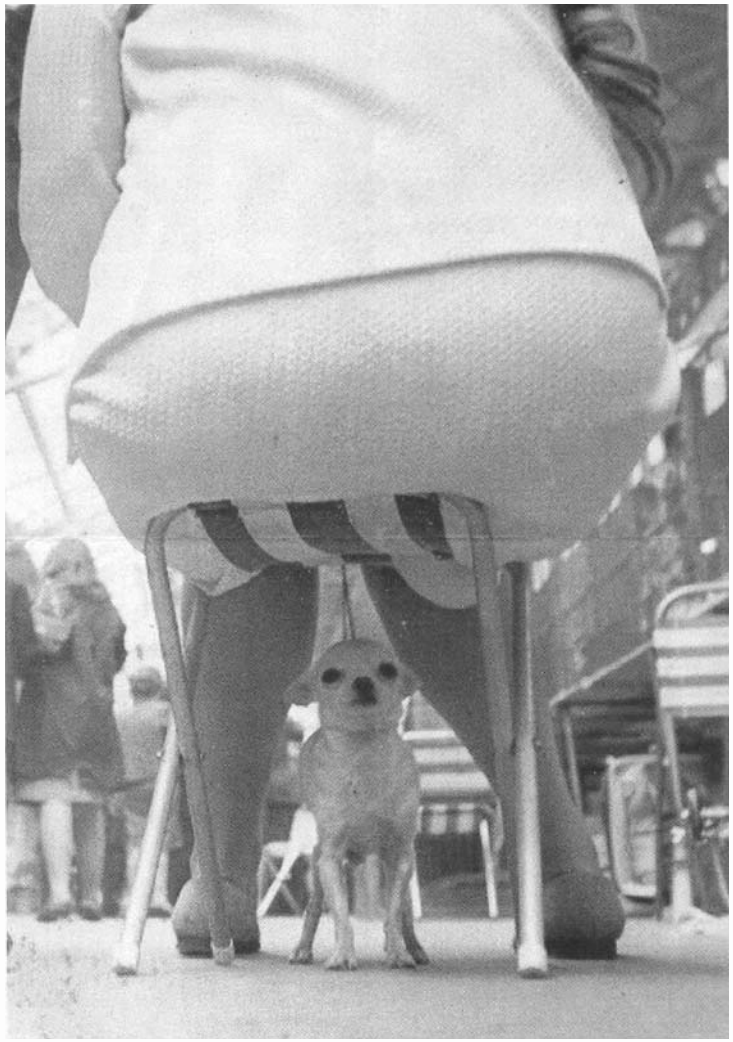
What you possibly think about me ...

Security people do have a good heart

... but a sick mind ...

well ...

... and you still feel secure...?



Important ...

- SAFEGUARD does not introduce a better security, but a better granularity, and auditing
  (an error 48 in GUARDIAN is as solid as in SAFEGUARD)
- Automatic tools are nice to watch – but it is better to understand, what they do, and what they do **NOT** do!
- Train yourself , and/or hire a trustworthy expert
- Test your system before intruders do
- Look at ALL THREE aspects covered in these three talks: GUARDIAN, OSS & LAN

Questions

- NonStop Systems are considered to be FailSafe – but what about their security?
- Does/can SAFEGUARD protect all system assets?
- … but GUARDIAN/SAFEGUARD does have two certificates:
  - NCSC (C2) and
  - GISA (F2 @Q3 and F7)
- So what … ???

Questions

- Can be broken into the system, or an application?
- Is it possible to gain access to ID's without the knowledge of the password?
- In case there are real threats - is there a solution?

General

- All my attacks start from a NON PRIV logged on TACL
  - NO SUPER.SUPER (255,255)
  - NO SUPER group (255,n)
  - NO group Manager (n,255)

  or already running resource I have access to
  - SQLCI, SCF etc.
- Sounds like a first hurdle – but all your administrators, operators, developers and system users do have interactive access to your system!

General

... and in one of the next presentations you even learn, how to get a password from the LAN, allowing you to get access to the system, WITHOUT having credentials!

General

- All GUARDIAN demos run on \BEECH of GreenHouse in Germany (S7000, G06.23)
- Connected by VPN through the Internet
- Used software:
  - MPWD (authentication service; access to the system)
  - SECOM (command level security, ID hopping)
  - Free- and ShareWare tools
  - Special demo programs (TAL/pTAL)
  - TACL macros

... and here we go ...

PATHWAY-Threat

- Access to the application ID
- Physical access to application data
- Worst case:
  Interactive access to SUPER.SUPER

- My classic way to break into a system

PATHWAY-Threat

- Weak point is insufficient default security of PATHWAY monitor
- Unknown security mechanism
- System applications are often started from SUPER.SUPER
  (do you use SUPER.SUPER in the day-to-day business?)
- Requirement to succeed an attack:
  Interactive access to the system with ANY ID

PATHWAY-Threat

- PATHWAY system (PATHMON)

  - PAID — is the ID of the starting user
  - Owner — by default the starting user; can be configured differently!
  - Security — by default “N”; can be configured differently!

PATHWAY-Threat

- PAID (Process Access ID)

  - derived from the starting user
  - propagated to all programs (= Servers), started from PATHMON
  - a PRIV ID gives management users access rights they should not get to

PATHWAY-Threat

- Owner

  - Set to PAID by default
  - can easily be changed to any other user ID
  - ID allowed to manage PATHMON

PATHWAY-Threat

- Security

  - Set to “N” by default
  - allows ALL system users to manage this PATHWAY system
  - can easily be changed to any other GUARDIAN security vector
  - related to “Owner”

PATHWAY-Attack

- Search for PATHMON's, running SUPER.SUPER (or any other interesting ID)

```
$GHS1 ARROW 23> status *,user super.super,prog $system.sys*.pathmon
Process             Pri PFR %WT Userid  Program file                Hometerm
$GHS        0,46    167     005 255,255 $SYSTEM.SYSTEM.PATHMON      $ZHOME
$S600       0,54    180     005 255,255 $SYSTEM.SYSTEM.PATHMON      $ZHOME
$GHS     B  1,58    167     001 255,255 $SYSTEM.SYSTEM.PATHMON      $ZHOME
$S600    B  1,74    180     001 255,255 $SYSTEM.SYSTEM.PATHMON      $ZHOME
$GHS1 ARROW 24>
```

PATHWAY-Attack

- Check PATHMON security setting

```
$GHS1 ARROW 24>pathcom $ghs;info pathway
PATHWAY
  MAXASSIGNS 100                [CURRENTLY 63]
  MAXDEFINES 0                  [CURRENTLY 0]
  .
  .
  MAXTERMS 60                   [CURRENTLY 0]
  MAXTMFRESTARTS 5
  OWNER \BEECH.255,255
  SECURITY "O"
$GHS1 ARROW 25>
```

PATHWAY-Attack

- Introduce a new server, such as SQLCI, FUP, BACKUP etc.
- SUPER.SUPER even gives access to ANY other system ID WITHOUT the need to know a password, AND: This break in is NOT audited in SAFEGUARD!

PATHWAY-Showtime

- Showtime ... ($GHS1.ITUG)

  - starting an insecure SUPER.SUPER PATHMON
  - demonstrating interactive access to SUPER.SUPER

  - starting a correct secured SUPER.SUPER PATHMON
  - demonstrating its robustness

PATHWAY - Solution

- Do NOT start any PATHWAY application from a privileged system ID
  - SUPER.SUPER
  - SUPER.xxx
  - xxx.MANAGER
- Set PATHWAY management security to “O”
- Define a PATHMON manager, which can be different from the PATHMON PAID

PATHWAY - Solution

- Put an ACL on the PATHMON process name
- Activate the PATHWAY log, and check it on a regular basis
- Make sure only authorized users can change the configuration files

PATHWAY - Solution

- Use the FreeWare tool GetPWSS to check all your pathway applications within seconds

  http://www.greenhouse.de/freeware/GetPWSS.html

- Use command level security products to give management access rights on (sub)command level
  (who is allowed to restart which server at what time from which IP address ...)

SPOOLER-Threat

- My second classic way to break into a system
- Same problem as with PATHWAY

SPOOLER-Threat

- SPOOLERs are often started from SUPER.SUPER at cold load time
- Weak point is unknown security mechanism
- Requirement: Interactive access to the system with ANY SUPER-Group ID

SPOOLER-Threat

- Management access is granted to:
  - the starting ID
  - all SUPER-group members
  - SUPER.SUPER
  - optional to group managers

SPOOLER-Threat

- Attack

  - Search for SPOOL, running SUPER.SUPER

```
$GHS1 ARROW 27> status *,prog $system.sys*.spool
Process             Pri PFR %WT Userid  Program file                Hometerm
$SPLS    B  0,43    150     001 255,255 $SYSTEM.SYSTEM.SPOOL        $ZTNP0.#PTPAAAA
$SPLS       1,38    150     001 255,255 $SYSTEM.SYSTEM.SPOOL        $ZTNP0.#PTPAAAA
$GHS1 ARROW 28>
```

SPOOLER-Attack

- Introduce a new print process, which is a normal GUARDIAN program, such as FUP, SCF, SQLCI etc.
- A SUPER.SUPER running SPOOL allows even interactive access to SUPER.SUPER (same procedure as with PATHWAY: Introduce a print process [= SPOOLER server])

SPOOLER - Solution

- Do NOT start a SPOOLER from SUPER.SUPER
- Consider running different SPOOLER systems, where the starting ID is the owner/manager
- Consider using ACLs on supervisor and collector processes
- Use command level security products to control access to SPOOLER systems

USERID/LUSERID-Threat

- Wrong security setting
- Unknown additional alternate file
- Requirement: Interactive access to the system with ANY ID and READ access to USERID/LUSERID

USERID/LUSERID-Threat

- READ access allows a FUP COPY which discloses unencrypted passwords
- READ/WRITE access allows the injection of a new password for EVERY user
- Additional alternate key copies each entry into a separate file, which can be used for a brute force attack

USERID/LUSERID-Solution

- Both files have to be secured to: "----"
  where the owner has to be: SUPER.SUPER
- Check with
  FUP INFO<file>,DETAIL
  for alternate file entries in
  $SYSTEM.SYSTEM.USERIDAK and
  $SYSTEM.SAFE.LUSERID
- Use the FreeWare tool FileTree to display all alternate key files of a given file

USERID/LUSERID-Solution

- Use FreeWare tool PWCRYPT to encode all unencrypted passwords
- Use FreeWare tool PWCHECK to find users WITHOUT a password
- Patch PASSWORD program
  (available since 1985...; needed for NON SAFEGUARD shops)
- Use appropriate SAFEGUARD settings

Alias Users - Threat

- Do you know all SUPER.SUPER related Alias users?
- Tandem engineers often place a SUPER.SUPER Alias onto the system, that makes life easier for them...
- Insufficient knowledge of SAFEGUARD
- Incomplete SAFEGUARD setup

Alias Users - Threat

- Unexpected access to SUPER.SUPER, where SUPER.SUPER is not used to logon…
- Requirement: Access to SAFECOM and insufficient OBJECTTYPE USER
- SUPER.SUPER used by a ‘wrong’ person (once)

Alias Users - Solution

- Check all Alias users
- Use the FreeWare tool MyUser to list all GUARDIAN/Alias user relations
- Delete/Freeze those users, not introduced/known by you
- Have OBJECTTYPE USER defined

SAFEGUARD - Threat

- Undefined OBJECTTYPEs
- Wrong understanding of ACL evaluation
- Wrong object ACLs

SAFEGUARD - Threat

- Each user can introduce a SUBVOL ACL, when OBJECTTYPE SUBVOL is not defined
- My classic way: Introduce a non existing ACL for subvol $SYSTEM.SYSTEM or any other interesting collection of files, do a file copy, and delete the ACL ...
- Check ACL evaluation, and find a hole...
  (SAFECOM INFO SAFEGUARD)

SAFEGUARD - Attack

- Add an ACL e.g. on SUBVOL level
- Access the required data
- Re-set the ACL

SAFEGUARD - Solution

- Understand SAFEGUARD
- Know what you do
- Introduce ***ALL*** OBJECTTYPEs
- Set up the evaluation rules for an easy understanding
- Check ACL evaluation with FreeWare tools
  - CRYSTAL
  - SECINFO

xxxCSTM - Threat

- Insufficient user default security, which is propagated to CSTM-files, especially of SUPER.SUPER's
  - FUPCSTM
  - TACLCSTM

- This is true for TACL Macros (MYMACS etc.) as well!

xxxCSTM - Attack

- Insert data into FUPCSTM, such as:
  - LICENSE <mycode>
- Then visit SUPER.SUPER and ask him, to do 'something' that activates the CSTM-file you changed
- Remove the code from FUPCSTM

- Insert data into TACLCSTM
  - what about a LOGOFF as first statement?

xxxCSTM - Solution

- Secure all CSTM files to “OOOO”
- No shared default locations
- No shared USER IDs
- Default security has to be “OOOO”, optionally “UUOO”
- Individualize all users
- Differentiate between functional and individual users

TACL Macro - Threat

- Same as CSTM-threat
- Hard coded passwords in TACL Macros

TACL Macro - Attack

- Search for MYMAC files and check for passwords

TACL Macro - Solution

- All users TACL Macros should be secured to: “OOOO”
- Do NOT have passwords hard coded anywhere; use products which support this!

Libraries - Threat

- Classic Trojan Horse
- Easy to install
- Difficult to find
- … do you know what I'm talking about???

… I love this method …

Libraries - Threat

- Adds code to an executable
- Can easily spoof passwords
- Can change the behavior of a program
  - copy data
  - change data
  - etc. etc. etc.
- Requirement:
  - write & execution access on program
  - execution access on library

Libraries - Attack

- Add a LIB to
  - TACL/FTPSERV to intercept USER_AUTHENTICATE_ : You get all passwords in the clear
  - any Tandem utility, and change the command behavior
  - ... be creative (or is it subversive?)!

Library-Showtime

- Showtime ... ($GHS1.ITUG)

  - logging on to a TACL that has a library attached: The classic Trojan Horse

Libraries - Solution

- Check all executables on your system.
  Use the FreeWare tool: SHOWLIB
- Remove suspect libraries.
  Use the FreeWare tool: BINDLIB
- Set the security of all executables to:
  “xOxO” to prevent any LIB binding
  Use the FreeWare Tool: SECURE

Libraries - Solution

- Generally:
  - Secure all executables to: “OOxO”
  - Secure all system EDIT files to: “xOOO”
  - Secure all system files to: “OOOO”
  - Secure all application files to: “OOOO”

Portconf - Threat

- PORTCONF causes LISTNER to start malicious code

Portconf - Attack

- Check security of PORTCONF and add an entry.
- Because LISTNER normally runs SUPER.SUPER, the defined resource runs SUPER.SUPER...

Portconf - Solution

- Check PORTCONF for suspicious entries
- Secure PORTCONF that only the system administrator can change it
- Do not start LISTNER from SUPER.SUPER – there is no need!

Search Path - Threat

- Before a resource is executed, TACL tries to find it in the search path
- A typo causes an error, but a program, named like a typo may cause a disaster...
- Requirement: Create access in a search path

Search Path - Attack

- Write a small program, that purges all files of the user, executing it
- Place this program in the search path and name it like a typo, e.g. EDOT
- ... lean back and wait ...

Search Path - Solution

- Introduce SAFEGUARD ACLs for all system wide search path locations: Deny CREATE for unauthorized users
- Inform you users to check their search path settings, and add an ACL as well

Alternate Key Files - Threat

- Alternate key files hold sensitive data, up to a complete data record
- Are not displayed by the INFO command, but require INFO,DETAIL
- Are easily overlooked

Alternate Key Files - Attack

- Add an alternate key file to a sensitive file, where the record contains the entire data record

Alternate Key Files - Solution

- Use FUP and check all your sensitive data files for unknown alternate key file entries
- Use FreeWare program FILETREE to display all alternate key files

Accessing Data on Disk - Threat

- A PURGE does not WIPE the data, it updates the Disks Free List Table
- Data is still available, and can be retrieved by ANY user, that is allowed to create a file

Accessing Data on Disk - Attack

- Create a big file
- Position the EOF to the last byte
- Perform a READ/COPY

Accessing Data on Disk - Solution

- Use CLEAR-ON-PURGE option
- Use the WIPE tool from GreenHouse
  - wipes files up to their physical EOF
  - wipes the space between the logical EOF and the physical EOF
  - wipes space between files

Denial of Service - Threat

- Exhaustive use of system resources
  - CPU power
  - system resources (internal tables)
  - disk and disk directory space
- Causes unavailable system and services
- May even cause a system HALT

Denial of Service - Attack

- By Intention

  Corrupting a CPU

```
?Nolist
?Source $system.system.extdecs0 (alter_priority_)
?List
Proc Test Main;
Begin
  While 1 do begin alter_priority_(199);
End;
```

Denial of Service - Attack

- By Intention

Corrupting a volume

```
?Nolist
?Source $system.system.extdecs0 (file_create_)
?List
Proc Test Main;
Begin
  String .system[0:35] := „$system“;
  Int     Len := 7;
  While 1 do begin File_Create_(SYSTEM:36,Len);
End;
```

Denial of Service - Attack

- By Intention

  Corrupting a CPU by flooding LISTNER with incomplete FTP calls

Denial of Service - Attack

- By error

  Wrong and/or no error handling in the error handling

Denial of Service - Attack

- By Tandem utilities

  - DIVER
  - TANDUMP

Denial of Service - Solution

- Code reading
- Exhaustive logic and error debugging
  Check error handling in error handling
- No compilers on production systems
- Test/development isolated from production

Denial of Service - Solution

- Use ListLib ShareWare to harden LISTNER
- Use PURGETMP FreeWare to keep track of ‘orphaned’ temporary disk files
- Revoke LICENSE flag from DIVER and TANDUMP, at least set a tight security

Covert Channel - Threat

- Information leakage to listener
- Hidden data channel

Covert Channel - Attack

- Changing the priority
  (ticker channel)
- Checking CPU buys values
- Checking date and time
- Checking EOF, file creates etc.

Covert Channel - Solution

- Code reading
- Procedure call check against negative list
- Exhaustive logic (20%) as well as error tests (80%)
- No production data for tests

Ghost Processes

- Started from a temporary file
- Very difficult to track down
- At least you should know about them

Social Engineering

- Works on ANY platform at any site
- Misuse of helpfulness
- Use of unthoughtfulness
  (do not think about what you do...)
- Most efficient non technical method

Best practice

- No code licensing - except you know what you do
- No PROGIDing code – use ID hopping products instead
- No Orphaned files
- No Shared IDs
- Stringent default security (OOOO)
- Control of functional users

Tools

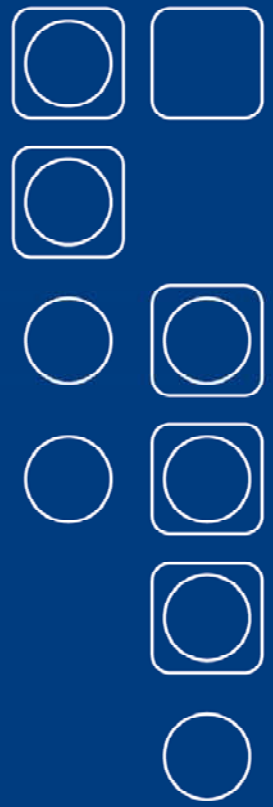
All mentioned tools are
Free- or ShareWare from GreenHouse
and can be found at:

www.GreenHouse.de

ITUG SUMMIT 2004

Third Parties

Baker Street Software
Bowden Systems
CAIL
comForte
Cross-EL
Crystal Point
CSP
GreenHouse
Gresham Software Labs
Insession Technologies
K2Defender
Unlimited Software Associates
XYPRO

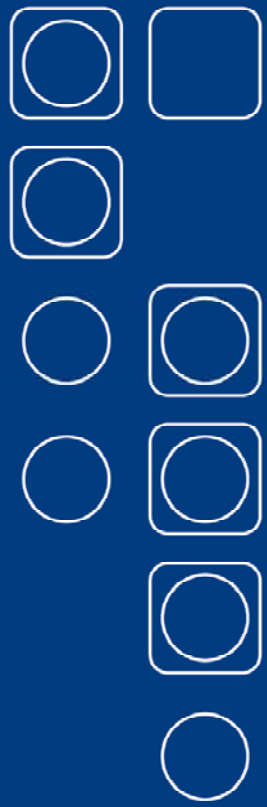


greenHouse

ITUG
The International HP NonStop Users Group
independent, not-for-profit, user run

ITUG SUMMIT 2004

Questions?



greenHouse

ITUG

The International HP NonStop Users Group
independent, not-for-profit, user run

Thank you for listening!

Now it is OSS time!