z/OS Security for Linux on a Zseries
Session 1791

Frank J. De Gilio
IBM Design Center
degilio@us.ibm.com
zSeries Linux is Very Cool

- Great for rapid development environment
- Used widely for intranet server consolidation
- Starting to be looked at for Internet application
- On zSeries Linux gets
  - Rapid Instance (image) creation
  - zSeries hardware qualities of service
  - Glass house SMS support
  - HiperSockets
  - A helping hand from z/OS (if you want)
First things First – Harden That Linux

• Most distributions are still not build as enterprise Linux instances.
• Enterprise Linux instances should have a specific function
• Create hardened instances and clone them
• Keep in mind the services that you need and don’t need.
• Use tools to help perform Linux hardening
Securing Linux Flowchart:

You are never done!

This is a full time job!

Lots and Lots of Research!

Start

Plug all Holes

No you Didn’t

Get Them All?

No

Yes
Linux Hardening With the Help of Tools

• Bastille
  – Comprehensive System View
  – Educational (Especially for new Linux users)
  – Support from the Linux Community
  – Is not officially available for Linux on zSeries
    • Install source
    • Tweak to support Linux on zSeries
    • Used Curses Support

• NMAP
  – Now with a Windows interface
## Nmap Results: Before Hardening

<table>
<thead>
<tr>
<th>Port</th>
<th>State</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>21/tcp</td>
<td>open</td>
<td>ftp</td>
</tr>
<tr>
<td>23/tcp</td>
<td>open</td>
<td>telnet</td>
</tr>
<tr>
<td>25/tcp</td>
<td>open</td>
<td>smtp</td>
</tr>
<tr>
<td>37/tcp</td>
<td>open</td>
<td>time</td>
</tr>
<tr>
<td>79/tcp</td>
<td>open</td>
<td>finger</td>
</tr>
<tr>
<td>80/tcp</td>
<td>open</td>
<td>http</td>
</tr>
<tr>
<td>110/tcp</td>
<td>open</td>
<td>pop-3</td>
</tr>
<tr>
<td>111/tcp</td>
<td>open</td>
<td>sunrpc</td>
</tr>
<tr>
<td>513/tcp</td>
<td>open</td>
<td>login</td>
</tr>
<tr>
<td>514/tcp</td>
<td>open</td>
<td>shell</td>
</tr>
<tr>
<td>515/tcp</td>
<td>open</td>
<td>printer</td>
</tr>
<tr>
<td>767/tcp</td>
<td>open</td>
<td>phonebook</td>
</tr>
<tr>
<td>901/tcp</td>
<td>open</td>
<td>samba-swat</td>
</tr>
<tr>
<td>2049/tcp</td>
<td>open</td>
<td>nfs</td>
</tr>
</tbody>
</table>

These Ports were all open “out of the box”.
## NMAP Results: After Bastille

<table>
<thead>
<tr>
<th>Port</th>
<th>State</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>21/tcp</td>
<td>open</td>
<td>ftp</td>
</tr>
<tr>
<td>23/tcp</td>
<td>open</td>
<td>telnet</td>
</tr>
<tr>
<td>25/tcp</td>
<td>open</td>
<td>smtp</td>
</tr>
<tr>
<td>37/tcp</td>
<td>open</td>
<td>time</td>
</tr>
<tr>
<td>79/tcp</td>
<td>open</td>
<td>finger</td>
</tr>
<tr>
<td>80/tcp</td>
<td>open</td>
<td>http</td>
</tr>
<tr>
<td>110/tcp</td>
<td>open</td>
<td>pop-3</td>
</tr>
<tr>
<td>765/tcp</td>
<td>open</td>
<td>webster</td>
</tr>
<tr>
<td>901/tcp</td>
<td>open</td>
<td>samba-swat</td>
</tr>
<tr>
<td>2049/tcp</td>
<td>open</td>
<td>nfs</td>
</tr>
</tbody>
</table>

Ports for: Sunrpc, Login, Shell, Printer, Phonebook are gone.
<table>
<thead>
<tr>
<th>Port</th>
<th>State</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>21/tcp</td>
<td>open</td>
<td>ftp</td>
</tr>
<tr>
<td>23/tcp</td>
<td>open</td>
<td>telnet</td>
</tr>
<tr>
<td>80/tcp</td>
<td>open</td>
<td>http</td>
</tr>
</tbody>
</table>

These results are misleading since the ftp and telnet ports are managed by inetd.
HiperSockets – Just The Facts

- HiperSockets = Internal Queued Direct IO
- Microcode maintained lookup table
- Three devices for each stack
  - Read Control
  - Write Control
  - Data Exchange
- 1024 Devices across all HiperSockets
- Supports Virtual IP Addressing and Dynamic Virtual IP Addressing
Cool HiperSocket Stack Picture

Maximum frame size 64K
(TCPIP MTU 56K)
HiperSockets – a Network View

- z/VM
  - Linux A: 192.168.0.1
  - Linux B: 192.168.0.2
  - HiperSocket FE: 192.168.0.3

- zLinux
  - 192.168.0.4
  - 192.168.10.1
  - HiperSocket FF

- z/OS
  - 192.168.10.2
## HiperSocket Tables

<table>
<thead>
<tr>
<th>LPAR</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Image</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Linux A</td>
<td>Linux B</td>
<td>z/VM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CHPID</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FE</td>
<td>FE</td>
<td>FE</td>
</tr>
<tr>
<td></td>
<td>Device</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7000 – 02</td>
<td>7004 – 06</td>
<td>7008 – 0A</td>
</tr>
<tr>
<td></td>
<td>Unit Address</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>00-02</td>
<td>04-06</td>
<td>08-0A</td>
</tr>
<tr>
<td></td>
<td>IP Address</td>
<td>192.168.0.1</td>
<td>192.168.0.2</td>
</tr>
</tbody>
</table>
Microcode at Work!!!
VM Guest LAN Support

- Virtual HiperSockets (Virtual Virtual sockets?!)?
- Emulates HiperSockets within a VM image
- Maximum number of unused CHPIDs -1
- 3072 I/O devices per guest LAN
- 1024 guests (TCP/IP stacks)
- Faster communication between Linux images than HiperSockets
Wheels Within Wheels

z/VM

HiperSocket
FB

192.168.20.1
Linux A
192.168.30.1

192.168.20.2
Linux B
192.168.0.2

HiperSocket
FE

192.168.0.3

zLinux

192.168.0.4
192.168.10.1

HiperSocket
FF

192.168.10.2

z/OS
Rsockets – Beyond the Hipe

- Fast – If you like that kind of thing
- VM Guest LANs faster for inter Linux communication
- Don’t expect IIOP flows to be faster…
- More secure communication
  - Unsniffable traffic between connections
  - Reduces the need for SSL
  - Lessens the dependency on encryption
    - Real performance benefits
- Less mercurial configuration than the wire stuff
Using VM to Clone Linux
PAM Who?

- Plugable Authentication Module
- Allows you to create security for each service you provide
- This allows you to limit access to services.
- Create layers of security to access certain functions.
- A bunch of different PAMs are available:
  - PAMSMB – Use NT to authenticate Linux Users
  - CUECAT – Bar code reader based authentication
  - PAMAFS – Use AFS to authenticate
  - LDAPPAM – Use LDAP to authenticate user
LDAP A Security Database

- Lightweight Directory Access Protocol
- Limited function database
- Relatively Static Data
- Based on a Directory (hierarchical) structure
A DN Has Attributes

- Name
- Location data
- Phone number
- Userid
- Password
- Login information
- Stuff like that
Things That Make You Go Hmmm....

Linux A

Linux B

Linux C

HiperSocket

LDAP Server

z/OS

RACF

RACF
In theory a good idea....
- RACF is a good security server
- LDAP interface makes it accessible
- One stop for security on all systems

In Practice not so good...
- RACF schema is not very flexible
- Wont support multiple hosts well
- End up having data in /etc/passwd too
But Wait! There’s More....

- Linux A
- Linux B
- Linux C

- HiperSocket
- LDAP Server
- z/OS
- RACF
- DB2
Start with LDAP PAM

Linux OS

Telnet Server Code

Userid Password

LDAP Client Code

LDAP PAM

Authentication Stub
In to LDAP on z/OS

LDAP Server

DB2
Userid
Password

User Data

RACF
OK
Now Get User Identity

Linux OS

- Telnet Server Code
- LDAP Client Code
- LDAP NSS
- Stub
- LDAP PAM
- OK
- Authentication Stub
- Get Data
From LDAP on z/OS
Ready to Go

Telnet Session Process

Linux OS

Telnet Server Code

LDAP Client Code

LDAP NSS

Stub

LDAP Client Code

LDAP PAM

Authentication Stub

User Data

User Data
O.K. How do I make this Work?

• Install an LDAP client
  – ldap-clientd-3.2.2-1.s390.rpm or an openldap open source package

• Install an LDAPPAM module:
  – PAM LDAP rpm - pam_ldap-56-74.s390.rpm
  – Could be one with the distribution

• Install NSS module:
  – Name differs depending on distribution
  – Probably packaged with distribution
Configure PAM module

File name depends on distribution:
host 192.168.100.140
port 389
base o=nis
binddn cn=admin
bindpw secret
ldap_version 3
pam_login_attribute uid
#pam_check_host_attr yes
Define PAM Options For Each Access

aslan@lingilio:/etc/pam.d >ls
chfn ftpd other ppp rlogin samba su1 xdm
chsh login passwd rexec rsh su sudo

# default configuration: /etc/pam.d/other
auth required /usr/lib/security/pam_warn.so
auth required /usr/lib/security/pam_deny.so
account required /usr/lib/security/pam_deny.so
password required /usr/lib/security/pam_warn.so
password required /usr/lib/security/pam_deny.so
session required /usr/lib/security/pam_deny.so
PAM Configuration Details

Column 1:

- Auth - Who is it?
- Account - Restrictions on this user.
- Session - Before/After execution
- Password - For changing the password
Column 2:
• Requisite - Gotta pass this one
• Required - If this fails others will be checked
• Sufficient - If this passes no other modules will be checked
• Optional - Can determine what the application gets

Column 3:
• Module (path) followed by args
auth sufficient /lib/security/pam_ldap.so
auth requisite /lib/security/pam_unix.so nullok
#set_secrpc
account sufficient /lib/security/pam_ldap.so
account required /lib/security/pam_unix.so
password sufficient /lib/security/pam_ldap.so
password required /lib/security/pam_unix.so nullok use_first_pass
use_authhtok
session required /lib/security/pam_unix.so none # debug or trace
session required /lib/security/pam_limits.so
#session sufficient /lib/security/pam_ldap.so
Define The User To LDAP on z/OS

- Using TDBM back end
- Native authentication
- Schema Additions
  - NIS schema
  - posixAccount schema
- Must have:
  - cn: common name
  - sn: surname
  - uid: userid (must be unique)
  - uidnumber Linux uid number
  - gidnumber Linux gid number
  - home directory: users home directory when logged in
  - Login shell: program to use when the user is logged in
Things to Remember!

• Do not store root in LDAP (I learned the hard way)
  – Also other (nobody...admin...)
• Store user account info into LDAP
• Control all of Linux images authentication with 1 database
  – Attribute host allows an administrator to define which hosts the user can use
• If using HiperSockets, might be able to avoid SSL
• Password is not retrievable from RACF
HiperSockets and Sysplex Distributor

Image 3
(Hidden VIPA 10.9.8.1)

Image 4
(Hidden VIPA 10.9.8.1)

Image 5
(Hidden VIPA 10.9.8.1)

Image 2
(Backup VIPA 10.9.8.1)

Image 1
(Primary VIPA 10.9.8.1)

Traditional Network
HiperSockets and Sysplex Distributor

- Image 3 (Hidden VIPA 10.9.8.1)
- Image 4 (Hidden VIPA 10.9.8.1)
- Image 5 (Hidden VIPA 10.9.8.1)
- Image 2 (Backup VIPA 10.9.8.1)
- Image 1 (Primary VIPA 10.9.8.1)

HiperSocket

Traditional Network
HiperSockets Sysplex Distributor WLM

Image 3 (Hidden VIPA 10.9.8.1)

Image 4 (Hidden VIPA 10.9.8.1)

Image 5 (Hidden VIPA 10.9.8.1)

Image 2 (Backup VIPA 10.9.8.1)

Image 1 (Primary VIPA 10.9.8.1)

Traditional Network
Wake Up It’s Almost Over!

- Linux on z/Series is the same as every other Linux
  - Must be hardened
  - Vulnerable to network attacks
  - Must be diligent to ensure that all holes covered
- Linux on z/Series is different from every other Linux
  - Simple cloning will allow to quickly clone hardened Linux
  - HiperSockets allow for fast secure communication between images
  - LDAP TDBM and RACF can be used as a back end for Linux authentication