Yuichi Nakamura, Hitachi Software Engineering Co., Ltd.
ynakam@hitachisofṭ.jp
1. What is SELinux?
2. Difficulties in Embedded SELinux
3. Development of Embedded SELinux
4. Application to various devices
5. Related works
Statistics of vulnerabilities found in embedded devices

N=77

* This data is corrected by us from the Internet.

Often reported: In a mailing list, 30 vulnerabilities are reported a year.
Linux based embedded devices are increasing
   - TV, DVD recorder, Cell phone, Home gateway, STB

Connected to the Internet

Exposed to attacks

Once exploited:
   - System is destroyed, used as spring board.
   - Device-makers have to recall to fix vulnerabilities.

Security technology suitable for embedded devices is needed.
• Update and Virus scan are common in PCs.

• Update
  – PC: OS vendors take care of all updates
  – Embedded devices
    • Device-makers have to prepare update
    • Heavy task
      – Watch all bugfix
      – Backport patch
      – Provide update software
    • Update will be delayed, or not prepared…

• Virus scan
  – Heavy (Pattern file: 30Mbyte in PC)

• Security technology effective even with no update is required.
• -> SELinux
• Security-Enhanced Linux
  – Developed by NSA (http://www.nsa.gov/selinux)
  – Implemented in kernel
  – Merged to 2.6

• Access Control Feature
  – Least privilege (Type Enforcement)
  – Mandatory Access Control (MAC)
    • No one (including root) can avoid

• SELinux can confine behavior of attackers
  • Very difficult to do harm
    – Effective before update

• Widely used for PCs
  – Enabled on Redhat, Fedora by default
DVD recorder

**Server**
- can listen
- can read (Deny)

**Media files**
- can read

**TCP 80**
- can listen

**Server config files**

**Media player**

**SELinux**

**Linux kernel**

**Security policy**

**Access control configuration**

- (1) load
  - Assign security domain to process
  - Even root cannot avoid

- (2) Assign security domain to process

**Attackers cannot do harm**

**If virus is included in media file, it cannot do harm**
Process | Permission | Resource
--- | --- | ---
| | | 

- Label based access control
- *Domain* label is assigned to processes
- *Type* label is assigned to resources
- Domain is not allowed nothing by default
- Allow necessary access permissions
Domain is allowed nothing by default

- Need to allow necessary accesses
- Configuration for access control rules
- Allow domains to access types

```allow httpd_t web_contents_t file:{ read };
```

<table>
<thead>
<tr>
<th>Domain</th>
<th>Type</th>
<th>Permission</th>
</tr>
</thead>
</table>
の関連 II は 設計 に関して

© 2023

11
#1. Extended attribute(xattr)

#2. Difficulties in security policy

#3. Performance
• Xattr
  – Data structure in file system used to stores labels and other attributes

• In SELinux, filesystem must support xattr!

• Xattr support in filesystem
  – Ext3, ext2 : OK
  – Jffs2 : OK
    • Merged to 2.6.18 by KaiGai
  – LogFS, yaffs: Not yet

• We have to use jffs2 for flash ROM
3 Steps to configure policy for embedded devices
- 1) Obtain sample policy (called refpolicy)
- 2) Tuning: remove unnecessary rules
- 3) Add necessary rules

For PC servers, refpolicy is good
- refpolicy is well-written for PC distros.

Difficult to write small, precise policy for embedded devices
- Have to remove too many rules
- Dependencies in policy
- Tons of macros
• **Refpolicy is intended for PC usage**
  – Included configuration for Fedora, Debian, SUSE
  – Large
    • File size: more than 2M, memory consumption more than 5M
  - To use for embedded need tuning
    - remove unnecessary rules

• **Example: To configure simple Apache server**
  – We removed more than 400 rules
  – For each rules,
    • You have to understand what you are removing,...
  – It is only a part
    • Base system is not included
• Dependency within policy
  – After removing part of policy, error appears because of dependency.
    • Have to declare label when using label.
    • If only declaration is removed, error appears.
    • Sometimes labels are declared in macro, declaration is hidden..
  – Example:
    • After removing policy related to sendmail, error appears in policy of apache
• Macros are traditionally used to write policy
• Macros are increasing:
  – More than 1000
  – Difficult to understand

• Also a lot of labels
• Overhead on system call
• Memory usage
• File usage

• Ported SELinux to SH based device and measured
  – Target board
    • Renesas R0P751RLC0011RL (R2DPlus)
      – SH 7751R(SH4 240Mhz)
      – RAM 64 Mbyte
  – SELinux before tuning
    • kernel 2.6.22
    • File system
      – ext3 on CF card
      – jffs2 on FLASH ROM
    • Policy : refpolicy in Fedora 6 without tuning
    • Userland: Userland as of Mar 2007
- Imbench
- The SELinux overhead

### Overhead is bigger in embedded environment

<table>
<thead>
<tr>
<th></th>
<th>SELinux Overhead</th>
<th>Overhead in Embedded Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Security policy, SELinux itself consumes memory

Memory usage by SELinux: B-A
  
  A = SELinux disabled kernel, output of free command
  
  B = SELinux enabled, output of free command
    
    Policy is taken from Fedora Core 6

Result: 5365 kbyte

For embedded, it is big.
Big for Flash ROM system
보도 자료는 이 앱의 기능을 위해 제공됩니다.
• Issues to port SELinux to embedded devices
  – #1. Extended attribute(xattr)
  – #2. Difficulties in security policy
  – #3. Performance

Our work
• Used SELinux Policy Editor instead of refpolicy.

• Refpolicy
  – Removing rules from existing policy file, to write small policy
  – Difficult

• SELinux Policy Editor
  – Write only rules that is necessary.
  – Easy to write small policy.
• Tool to configure SELinux policy

• Main feature: SPDL(Simplified Policy Description Language)
  – Hide labels and dependency internally

• Developed by Hitachi Software

• GPL
  – http://seedit.sourceforge.net/
Hides detail of SELinux (SPDL)

Simplified Policy → seedit-converter → SELinux Policy

GUI, Command line tools
Policy generator, template maker
* Example of SPDL: policy for Web server program

```plaintext
{
    domain httpd_t;
    program /usr/sbin/httpd;

    ... 
    allow /etc/httpd/** r,s;
    allow /var/log/httpd/** r,a,s;
    allow /var/www/** r,s;
    allownet -protocol tcp -port 80,443 server;
}
```

- Hide labels
  - Name based configuration: Can use file name, port number
  - Resolve dependency internally

- Simplified permissions
• Do not have to use sample policy
  – no macros, dependencies

• Can write custom policy for embedded devices
  – SPDL is easy to understand

• Can write small policy
  – can describe only what you need
  – Policy size : about 60k byte for 10 apps

• http://seedit.sourceforge.net/
• Overhead
• Memory footprint
• File size
• Mainly reduced read/write overhead
  – It was big (about 150%)

• Other tuning
  – Hand optimization
  – Removed logics about unused permission
    • such as NIC, IP address
• Duplicated permission checks in file read/write
  – In open and read/write system call

• Permission check can be removed at read/write
  – Need check only policy is changed after open

• Made patch, merged in 2.6.24
<table>
<thead>
<tr>
<th>Column A</th>
<th>Column B</th>
<th>Column C</th>
</tr>
</thead>
<tbody>
<tr>
<td>例 例例</td>
<td>例 例例</td>
<td>例 例例</td>
</tr>
<tr>
<td>例 例例</td>
<td>例 例例</td>
<td>例 例例</td>
</tr>
<tr>
<td>例 例例</td>
<td>例 例例</td>
<td>例 例例</td>
</tr>
<tr>
<td>例 例例</td>
<td>例 例例</td>
<td>例 例例</td>
</tr>
<tr>
<td>例 例例</td>
<td>例 例例</td>
<td>例 例例</td>
</tr>
<tr>
<td>例 例例</td>
<td>例 例例</td>
<td>例 例例</td>
</tr>
<tr>
<td>例 例例</td>
<td>例 例例</td>
<td>例 例例</td>
</tr>
<tr>
<td>例 例例</td>
<td>例 例例</td>
<td>例 例例</td>
</tr>
<tr>
<td>例 例例</td>
<td>例 例例</td>
<td>例 例例</td>
</tr>
</tbody>
</table>

- Good in read/write!
- Need work in “create”
• Development of policy by SELinux Policy Editor
  – can write small policy easily
  – Wrote policy for 10 apps

• Removing big buffers in kernel
  – Buffers for 32768 policy rules : 252K byte
  – Modified to allocate dynamically depending of policy size
  – Only 1kbyte is allocated when small policy is loaded
    • Merged to 2.6.24
      – http://marc.info/?t=118767097300001&r=1&w=2
- Small policy contributed a lot (about 4.6M)
• (1) Writing small policy by SELinux Policy Editor

• (2) Reducing size of library: small libselinux
  – separate libselinux and libsepol
  – Remove unnecessary functions from libselinux
  – Merged to SELinux community
    • http://marc.info/?l=selinux&m=118064545200576&w=2
    • `make EMBEDDED=y` build option
  – 482k -> 66k

• (3) Reducing size of commands
  – a) Integrated commands to BusyBox
    • With Japanese community
    • Merged to BusyBox
  – b) Choose least set of commands
    • `load_policy, setfiles, restorecon, ls -Z, ps -Z, setenforce, getenforce, is_selinux_enabled`
  – 375k -> 11k
<table>
<thead>
<tr>
<th></th>
<th>Before tuning (kbyte)</th>
<th>After tuning (kbyte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel (zImage)</td>
<td>73.7</td>
<td>73.7</td>
</tr>
<tr>
<td>Library</td>
<td>482.1</td>
<td>66.3</td>
</tr>
<tr>
<td>Command</td>
<td>374.6</td>
<td>10.8</td>
</tr>
<tr>
<td>Policy</td>
<td>1,356.2</td>
<td>60.4</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>2,286.6</strong></td>
<td><strong>211.2</strong></td>
</tr>
</tbody>
</table>
THE BEGINNING OF GENESIS STORIES
- SH based
  - L-Box (NTT Comware)
    - SH7751R
    - Originally 2.4 based
    - Without modifying userland
  - CAT 760 (Silicon Linux)
    - SH7760 based small board
    - Rootfs on 16Mbyte Flash ROM

- ARM based
  - Zaurus(Angstrom)
  - Android on Zaurus
• Not yet to Android on QEMU
  – Yaffs2 does not support xattr

• We ported SELinux to Android on Zaurus
  – File system is ext3
• Two domains can be assigned
  – Android_init_t: Programs run from init
  – Android_java_t: Programs run from “app_process”

• Can not assign domains for separate java apps
  – All run as “android_java_t”
    • They are launched from “app_process”
- Multi-mode phone
- Private mode/Business mode in one phone
- Security policy switches between mode

Private Mode

```
android_java_t
```

Java Apps

- Business Data
- Private Data

Business Mode

```
android_java_t
```

Java Apps

- Business Data
- Private Data

Demo is from 17:00!
• Assigning domains to each java apps
  – We should be able to do..
  – We want source of “app_process”!!!

• Xattr for yaffs
  – Planning to do
• What is Audit?
  – Framework to obtain system call logs

• Can obtain logs useful to develop SELinux policy
  – Full path name

• Not mandatory, but useful

• CPU dependent because entry.S has to be modified.
  – Supports x86, Power PC, MIPS
  – SH not supported

• Submitted audit for SH patch, merged to 2.6.25
• xattr support for jffs2
  – By KaiGai merged to 2.6.18

• Improvement of latency in security check
  – By KaiGai merged to 2.6.24
    • http://marc.info/?t=119078657600002&r=1&w=2

• BusyBox for SELinux
  – SELinux Applets
  – Assigning domains to BusyBox applet
    • By Shinji: Merged to 1.8.2
• Strict policy

• More tuning
  – We can reduce more
    • Example
      – we can remove MLS support, booleans from kernel

• xattr for yaffs, logfs
• Difficulties in Embedded SELinux
  – Difficulty in policy
  – Performance problem

• Development of Embedded SELinux
  – Policy by SELinux Policy Editor
  – Tuning

• Application to some devices

• SELinux is suitable security technology for embedded!
  – Effective without update
  – Architecture independent
  – Lightweight
• People in seBusyBox project
  – KaiGai
    • General advices, hosted project site, ml

• SELinux community
  – Stephen Smalley
    • Advices/ideas about implementation of tuning SELinux

• BusyBox community
  – Denis Vlasenko
    • Advices about BusyBox

• Renesas solutions
  – Yusuke Goda: flash ROM boot support for evaluation board
See http://elinux.org/SELinux

SELinux Policy Editor
  - http://seedit.sourceforge.net/

Reducing read/write overhead
  - Merged to 2.6.24

Removing big fixed size buffer
  - Merged to 2.6.24
  - http://marc.info/?t=118767097300001&r=1&w=2

Reducing size of library
  - Merged to libselinux 2.0.35
  - http://marc.info/?l=selinux&m=118064545200576&w=2

SELinux’ed BusyBox
  - Many applets merged
  - Assigning domain to applets

Improving latency in permission check
  - Merged to 2.6.24
  - http://marc.info/?t=119078657600002&r=1&w=2

Audit for SH
  - Merged to 2.6.25
Linux is a registered trademark of Linus Torvalds in the U.S. and other countries.

All other trademarks or registered trademarks are the property of their respective owners.