六、BOA 服务器的移植

1、源码下载

```
http://www.boa.org/
News! (last updated 23 February 2005)
Latest Released Version (0.94.13) here (signature here)  --- 下载地址
```

1.1 boa 简介：

其可执行代码只有大约 60KB 左右，Boa 是一个单任务的 HTTP 服务器，Boa 只能依次完成用户的请求，而不会 fork 出新的进程来处理并发连接请求。Boa 支持 CGI。

Boa 的设计目标是速度和安全。(CGI 只是一个进程，用来提供接口)，自动目录生成和自动文件枪支进行拼接。

Boa 的主要设计目标是速度和安全性。安全性在“不能被恶意用户破坏”的意义上，不是“细粒度访问控制和加密通信”。

```
特点：可靠性和可移植性，Boa 不是作为功能强大的服务器。
```

开发平台：GNU / Linux 是目前的开发平台。

```
推荐：GoAhead   Apache2
```

跨平台移植三部曲：

```
1、./configure 生成一个 Makefile      ./configure  --prefix   --host
2、make
3、make install
```

2、解压源码：

```
tar -xvf boa-0.94.13.tar.gz
```

3、进入源码目录下的 src 目录：

```
cd boa-0.94.13/src/
```

在源码目录下配置与编译。执行

```
./configure
```

这是根据 configure.in 文件进行一系列的配置，生成 config.status,configure,和 Makefile 文件。

如果无法执行，考虑权限，修改该目录下的 configure 的属性为可执行 chmod 755 configure

4、make 编译源码：

问题 1：

```
目录：  boa-0.94.13/src$ make
    yacc  -d boa_grammar.y
    make: yacc: Command not found
    make: *** [y.tab.c] Error 127
解决方法：
    /boa-0.94.13/src$ sudo apt-get install bison
```

问题 2：

```
    yacc  -d boa_grammar.y
    gcc  -g -O2 -pipe -Wall -I.   -c -o y.tab.o y.tab.c
    ......
    lex  boa_lexer.l
    make: lex: Command not found
    make: *** [lex.yy.c] Error 127
解决方法:
    /boa-0.94.13/src$ sudo  apt-get install flex
WARNING: The following packages cannot be authenticated!
  libfl-dev flex
Install these packages without verification? [y/N]
E: Some packages could not be authenticated
注意加参数-f 自动搜索依赖，下面选择 y  :  sudo  apt-get install flex -f
Do you want to continue? [Y/n] y
WARNING: The following packages cannot be authenticated!
  libfl-dev flex
Install these packages without verification? [y/N] y
```

问题 3：

```
util.c: In function ‘get_commonlog_time’:
util.c:100:39: error: pasting "t" and "->" does not give a valid preprocessing token
         time_offset = TIMEZONE_OFFSET(t);                                      ^
compat.h:120:30: note: in definition of macro ‘TIMEZONE_OFFSET’
 #define TIMEZONE_OFFSET(foo) foo##->tm_gmtoff                          ^
make: *** [util.o] Error 1
```

问题描述：在 compat.h 文件中的宏定义问题： ‘TIMEZONE_OFFSET’

```
解决方法:
    修改 boa-0.94.13/src$ vi compat.h  +120
    找到
    #define TIMEZONE_OFFSET(foo) foo##->tm_gmtoff
    修改成
    #define TIMEZONE_OFFSET(foo) (foo)->tm_gmtoff

    将 boa.c 中的 vi   boa.c
    225     #if 0     //注释掉
    226         if (setuid(0) != -1) {
    227             DIE("icky Linux kernel bug!");
    228         }
    229     #endif

这三行注释掉，否则 boa 启动时会出现“boa.c:226 - icky linux kernel bug!: No suchfile or directory 错误”
```

```
清除之前编译的内容 make clean （防止某些依赖的信息出现问题）
  重新 make
  再次 boa-0.94.13/src$ make
  make: Nothing to be done for `all'.
```

make 主要检查的是时间戳， 只要 target 比依赖的文件时间靠后它就认为不需要编译

5.建立安装目录

```
  sudo mkdir -p  /boa    /boa/www    /boa/cgi-bin   /boa/log
  将 src/defines.h 中的
       #define SERVER_ROOT "/etc/boa"
修改为
  #define SERVER_ROOT "/boa"
```

（这样 boa 程序启动时会在/boa 目录下寻找 boa.conf 配置文件 ,并且将/boa 文件夹作为服务器的根目录）。

将需要的文件复制到安装目录中

```
  将 boa-0.94.13/src 目录下生成的 boa、boa_indexer 二进制文件复制到/boa 下
      sudo  cp  boa boa_indexer  /boa
  将 boa-0.94.13 目录下的 boa.conf 文件复制到/boa 下
      sudo  cp  boa.conf  /boa
      可选项： arm-linux-strip boa（去掉 boa 中的调试信息，只是减小文件的大小可以执行也可以不执行）
  将/etc/mime.types 复制到/boa 目录下
      sudo cp  /etc/mime.types  /boa
```

7、返回 boa 的顶层目录 --- 修改配置文件

```
  boa-0.94.13$ ls
      boa.conf   contrib  docs      extras      README
      ChangeLog  CREDITS  examples  Gnu_License  src
```

vi boa.conf， //几乎全部指定到 boa 的目录下，这样方便我们进行管理

重新指定一些文件的生成路径，因为重新指定这些路径后会帮助我们深刻的理解关于 boa 服务器的工作机制

```
  Port  80
  User  0
  Group 0
```

bind 调用的 IP 地址，一般注释掉，表明绑定到 INADDR_ANY，通配于服务器的所有 IP 地址
#Listen 192.68.0.5
error_log 和 access_log 会自动生成，只要指定生成路径就可以了。

```
  ErrorLog /boa/log/error_log
  AccessLog /boa/log/access_log
```

存放 HTML 文件的根路径

```
  DocumentRoot /boa/www
```

```
UserDir public_html
```

默认页面，若之输入 http://127.0.0.1/则会自动返回给浏览器默认页面 index.html

```
DirectoryIndex index.html
##### 保持默认
DirectoryMaker /boa/boa_indexer   //被修改
KeepAliveMax 1000
KeepAliveTimeout 10
MimeTypes /boa/mime.types   //被修改
DefaultType text/plain
#####指定传给 cgi 程序的 PATH 环境变量
CGIPath /bin:/usr/bin:/usr/local/bin
#####保持默认
Alias /doc /usr/doc
#####如果输入 http://127.0.0.1/cgi-bin/test.cgi，则 boa 服务器会到/boa/cgi-bin 中寻找 test.cgi 程序。
ScriptAlias /cgi-bin/ /boa/cgi-bin/
```

8.建立测试页面

（1）index.html ，将 index.html 放在/boa/www 目录下

```
sudo cp index.html image.jpg /boa/www
<html>
    <body>
        <h3>this is a test!</h3><br/>
        <img src="image.jpg"/>
        <h3>tree picture</h3><br/>
        <a href="/cgi-bin/test.cgi">to cgi page</a>    //指定了 cgi 可执行文件存放的路径，默认从/boa 的根目录开始查找
    </body>
</html>
```

注意： 笔记本打开，另存为 utf-8 格式，自己添加一张图片到当前的目录下

（2）test.c， 使用 gcc -o test.cgi test.c，将 test.c 编译生成 test.cgi，后缀为 cgi 的类型编译后得到的 test.cgi 放在/boa/cgi-bin 目录下，

```
sudo cp test.cgi  /boa/cgi-bin/

#include <stdio.h>
int main()
{
  printf("Content-type:text/html\n\n"); //这句一定要加上
  printf("<html><body>");
  printf("<font style=\"color:red; font-size:30px;\">Hello, CGI!</font><br/>");
  printf("<a href=\"/index.html\">return index.html</a>");
  printf("</body></html>");
  return 0;
}
```

9、查看/boa 目录下所有的文件：

```
fengjunhui@ubuntu:/boa$ tree -a
.
├── boa
├── boa.conf
├── boa_indexer
├── cgi-bin
│   ├── pass.cgi
│   └── test.cgi
├── log
│   ├── access_log
│   └── error_log
├── mime.types
└── www
    ├── image.jpg
    ├── index.html
    └── pass.html

3 directories, 11 files
fengjunhui@ubuntu:/boa$
```

10、测试效果

进入/boa 目录，使用./boa 来运行 boa 服务器（当然也可将/boa 路径加入系统 PATH 环境变量，这样不用进入/boa 目录，直接输入 boa 就可以了）

在浏览器中输入 http://127.0.0.1/便可访问到默认的页面 index.html,

```
或：    http://192.168.1.200:80
        http://localhostip:port
```

点击 index.html 页面中的超链接便可访问到 cgi 测试页面，点击 test.cgi 中的超链接又可返回 index.html 页面。

（1）index.html 页面

```
fengjunhui@ubuntu:/boa$ ./boa
[19/Jul/2017:08:44:28 +0000] log.c:73 - unable to dup2 the error log: Bad file descriptor
```

解决方法：

```
这个问题其实就是你没有创建 log 文件夹，以及文件夹下的 log_error 和 access_error 文件引起的。只要在 boa.conf 中找到这两个文件的信息，将其路径改成自己需要的路劲，并在相应的位置创建对应文件就 OK。
  62 ErrorLog /boa/log/error_log
  74 AccessLog /boa/log/access_log
```

fengjunhui@ubuntu:/boa$ ls　发现有 log 目录

boa　boa.conf　boa_indexer　cgi-bin　log　mime.types　www

问题描述

```
log.c:73   unable to dup2 the error log: Bad file descriptor
```

问题解决：

```
Vi  log.c  +73
     扩展：
     函数名：  dup2
     功能：  复制文件描述符
     用法：  int dup2(int oldfd,int newfd);
  修改 src/log.c
  注释掉
     if (dup2(error_log, STDERR_FILENO) == -1) {
         DIE("unable to dup2 the error log");
     }
     为：
   71 #if 0
   72         /* redirect stderr to error_log */
   73         if (dup2(error_log, STDERR_FILENO) == -1) {
   74             DIE("unable to dup2 the error log");
   75         }
   76 #endif
```

/boa 目录下重新执行 ./boa 执行

```
  Cannot open /boa/log/access_log for logging: logfile open: Permission denied
  出错，因为不是 root 用户（虚拟机），权限不够。（A9 开发板默认登录为 root 用户，不需要加 sudo）
  sudo ./boa

  然后再./boa  此时 boa 服务器就已经启动，打开 ubuntu 浏览器，输入 ubuntu 的 ip 地址，就可以看到你放在/var/www 里的网页了。ok
```

测试结果：

```
  http://127.0.0.1/index.html
  this is a test!
```

```
    return to cgi page
```

```
    点击跳转:
    Hello, CGI!
    return index.html
```

以上是在 PC 上部署 BOA 的大致步骤，仅供参考。

问题描述：

fengjunhui@ubuntu:/boa$ sudo ./boa
[19/Jul/2017:13:44:11 +0000] boa.c:194 - unable to bind: Address already in use

如何关闭 boa 服务器：

```
fengjunhui@ubuntu:~$ ps -axj | grep "boa"
 2102 24862 24860 24255 pts/18   24255 S    65534   0:00 ./boa
 2102 25735 25733 24255 pts/18   24255 S    65534   0:00 ./boa
25753 25793 25792 25753 pts/4    25792 S+    1000   0:00 grep --color=auto boa

kill -9 pid（boa）
fengjunhui@ubuntu:~$ sudo kill -9 24862
fengjunhui@ubuntu:~$ sudo kill -9 25733
```

重新启动 boa，再次通过浏览器访问 cgi，访问成功！

```
fengjunhui@ubuntu:~/Boa-0.94/boa-0.94.13/src$ sudo cp boa_indexer boa /boa
cp: cannot create regular file ‘/boa/boa’: Text file busy
cp 并不改变目标文件的 inode，事实上它的实现是这样的：
# strace cp test2 test 2&1 | grep open.*test
open("test2", O_RDONLY|O_LARGEFILE) = 3
Linux 由于 Demand Paging 机制的关系，必须确保正在运行中的程序镜像（注意，并非文件本身）不被意外修改，
因此内核在启动程序后会锁定这个程序镜像的 inode。这就是为什么 cp 在用“O_WRONLYO_TRUNC”模式 open 目标文件时会失败。
```

```
而先 rm 再 cp 的话，新文件的 inode 其实已经改变了，原 inode 并没有被真正删除，直到内核释放对它的引用。
 fengjunhui@ubuntu:/boa$ sudo rm boa
 sudo cp boa_indexer boa /boa
```

扩展部分：

11、移植到 A9 开发板上：

（注意修改源码，具体的现象可以参考视频，遇到问题推荐自己通过百度解决问题，会查资料也是一种能力）

（1）移植 boa 到嵌入式 linux 上的方法和上面几乎一样，具体做法是\在

```
./configure
 生成 Makefile 后将 Makefile 中的
CC = gcc   CPP=gcc -E
改为
CC = arm-linux-gcc,
CPP = arm-linux-gcc -E,
```

然后 make 就可以了。

（2）重新拷贝修改后的文件：

```
 将 boa-0.94.13/src 目录下生成的 boa、boa_indexer 二进制文件复制到/boa 下
 将 boa-0.04.13 目录下的 boa.conf 文件复制到/boa 下
 将/etc/mime.types 复制到/boa 目录下
 将 test.cgi 放在/boa/cgi-bin 目录下，
 将 index.html image.jpg 放在/boa/www 目录下
```

（3）拷贝文件到 rootfs 文件系统目录下：

```
    cd boa 进入 boa 文件目录
```

移植到开发板后：

```
    [root@fengjunhui boa]:~$ ./boa
    No such user: nobody
    解决方法:
        sudo vi boa.conf
        (1)Group 的修改
            修改 Group nogroup  为 Group 0
        (2)user 的修改
            修改 User nobody 为 User 0
```

[root@fengjunhui boa]:~$ ./boa

问题描述：

```
 [01/Jan/1970:00:23:10 +0000] boa.c:211 - getpwuid: Success
[01/Jan/1970:00:00:39 +0000] boa.c:211 - getpwuid: No such file or directory
```

解决：修改 src/boa.c

```
注释掉下面两句话：
#if 0
      if (passwdbuf == NULL) {
              DIE(”getpwuid”);
              }
              if (initgroups(passwdbuf->pw_name, passwdbuf->pw_gid) == -1) {
              DIE(”initgroups”);
              }
#endif
```

（2）移植 boa 到嵌入式 linux 后，在启动时若出现“gethostbyname:: Success "然后程序退出，则需在原 boa.conf 文件中增加一行：

ServerName www.your.org.here

然后重新运行 boa,然后在主机浏览器输入开发板网址（http://开发板 ip:port）

```
[root@fengjunhui ]:~$ cd boa
[root@fengjunhui boa]:~$ ls
boa          boa_indexer  log          www
boa.conf     cgi-bin      mime.types
[root@fengjunhui boa]:~$ ./boa
[01/Jan/1970:00:00:24 +0000] boa: server version Boa/0.94.13
[01/Jan/1970:00:00:24 +0000] boa: server built Jul 19 2017 at 23:07:48.
[01/Jan/1970:00:00:24 +0000] boa: starting server pid=1185, port 80

[root@fengjunhui boa]:~$ [01/Jan/1970:00:01:30 +0000] request from 192.168.8.223
"GET /favicon.ico HTTP/1.1" ("/boa/www/favicon.ico"): document open: No such file
or directory
[01/Jan/1970:00:01:30 +0000] request from 192.168.8.223 "GET /favicon.ico
HTTP/1.1" ("/boa/www/favicon.ico"): document open: No such file or directory
这个问题不影响使用：
```

问题描述：

```
[01/Jan/1970:00:01:34 +0000] cgi_header: unable to find LFLF
出错位置
    cgi_procee_header()
对于错误 cgi_header: unable to find LFLF 的补充

浏览器中显示 502 Bad Gateway
The CGI was not CGI/1.1 compliant.
解决方法：在 Ubuntu 端和 A9 端的交叉编译环境不一样，最好链接为静态库，这样不用单独提供
库的支持
<3>编译方式:arm-linux-gcc -static -o xxx.cgi  xxx.c
```

可能的原因:

<1>输出 MIME 格式错误，结尾一定要有两个换行符

```
    printf("Content-Type:text/html\n\n")
```

<2>权限错误

```
chmod 777 var/www/cgi-bin/xxx.cgi
```

<3>编译方式:arm-linux-gcc -static -o xxx.cgi xxx.c

<4>备用方案：代码本身错误也会引起这个问题。实践中这个问题困扰了很久，一个字符串数组的处理有了问题,提示的错误信息却是这个，开始一直郁闷于格式和权限，也可能是调用的一个小函数有些问题而导致的。所以在遇到这个问题时，如果输出格式和权限都正确的情况下，检查代码是个很好的选择

测试：

```
在 Ubuntu 上输入你板子的 ip: http://192.168.1.100/
```

OK ，实现了和直接在 Ubuntu 上操作一样的结果

输入指定的路径去访问：

```
http://192.168.1.100/index.html
http://192.168.1.100/pass.html    跳转访问到指定的 cgi
```

指定是 OK 的。

可选项： ***----->然后给 boa 瘦身

```
arm-none-linux-gnueabi-strip boa
fengjunhui@ubuntu:~/smartstorage/boa-0.94/boa-0.94.13/src$ arm-none-linux-gnue
abi-strip boa
arm-none-linux-gnueabi-strip: Unable to recognise the format of the input file `boa'
我们并没有修改 Makefile，那么这个时候编译生成的 x86 格式的 boa
fengjunhui@ubuntu:~/smartstorage/boa-0.94/boa-0.94.13/src$ file boa
boa: ELF 32-bit LSB  executable, Intel 80386, version 1 (SYSV), dynamically linked
(uses shared libs), for GNU/Linux 2.6.24,
BuildID[sha1]=95bed4fc9481ca5ac84616c6cfb2578f4e4d1a2f, not stripped
F
```

错误记录：

```
line: 1: Syntax error: word unexpected (expecting ")")
```

开发板上运行可执行程序报出错误：

line1: 1: Syntax error: word unexpected (expecting ")")

解决思路：

1.编译器的问题

```
用 arm-linux-gcc 编译，可能原来是用 gcc 编译的。
假如是脚本，#!/bin/sh 改 #!/bin/bash 试试。
```

2.文件完整性

重新烧写或上传一遍。

3.编译命令问题

比如我的一个测试程序 test.c ：

```
arm-linux-gcc -o test.o -c test.c    //编译为目标文件
arm-linux-gcc -o test.o test.c        //编译为可执行文件
```

莫听竹林打叶声，何妨吟啸且前行。竹杖芒鞋轻胜马，谁怕？一蓑烟雨任平生。

（----------完----------）

boa.conf 内容详解：

25 Port 80

监听的端口号，缺省都是 80，一般无需修改。注意，如果你的 httpd 服务器已经开启，那么你必须关掉或者重新设定端口号，

当你修改了模式 http 端口那么访问的时候要加上端口号，如 http://localhostip:port

作为哪个用户运行，即它拥有该用户的权限，一般都是 nobody，需 要 /etc/passwd 中有 nobody 用户

User nobody

查看： vi /etc/passwd 18 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin

作为哪个用户组运行,即它拥有该用户组的权限,一般都是 nogroup,需要在/etc/group 文件中有 nogroup 组

Group nogroup

查看： vi /etc/group 39 nogroup:x:65534:

是否使用本地时间。如果没有注释掉，则使用本地时间。注释掉则使用 UTC 时间

UseLocaltime

是否记录 CGI 运行信息，如果没有注释掉，则记录，注释掉则不记录

VerboseCGILogs

服务器名字

ServerName www.chad.com

是否启动虚拟主机功能，即设备可以有多个网络接口，每个接口都可以拥有一个虚拟的 Web 服务器。一般注释掉，即不需要启动

VirtualHost

非常重要，HTML 文档的主目录。如果没有以/开始，则表示从服务器的根路径开始。注意，如果此处设置不正确，将不能打开网页

DocumentRoot /var/www

如果收到一个用户请求的话，在用户主目录后再增加的目录名

UserDir public_html

HTML 目录索引的文件名，也是没有用户只指明访问目录时返回的文件名

DirectoryIndex index.html

当 HTML 目录没有索引文件时，用户只指明访问目录时，boa 会调用该程序生成索引文件然后返回给用户，因为该过程比较慢最好不执行，

可以注释掉或者给每个 HTML 目录加上 DirectoryIndex 指明的文件

DirectoryMaker /usr/lib/boa/boa_indexer

如果 DirectoryIndex 不存在，并且 DirectoryMaker 被注释，那么就用 Boa 自带的索引生成程序来生成目录的索引文件并输出到下面目录，

该目录必须是 Boa 能读写

DirectoryCache /var/spool/boa/dircache

一个连接所允许的 HTTP 持续作用请求最大数目，注释或设为 0 都将关闭 HTTP 持续作用

KeepAliveMax 1000

HTTP 持续作用中服务器在两次请求之间等待的时间数，以秒为单位，超时将关闭连接

KeepAliveTimeout 10
指明 mime.types 文件位置。如果没有以/开始，则表示从服务器的根路径开始。可以注释掉避免使用 mime.types 文件，此时需要用 AddType 在本文件里指明
MimeTypes /etc/mime.types
文件扩展名没有或未知的话，使用的缺省 MIME 类型
DefaultType text/plain
提供 CGI 程序的 PATH 环境变量值
CGIPath /bin:/usr/bin:/usr/local/bin

将文件扩展名和 MIME 类型关联起来，和 mime.types 文件作用一样。如果用 mime.types 文件，则注释掉，如果不使用 mime.types 文件，则必须使用
AddType application/x-httpd-cgi cgi
指明文档重定向路径
Redirect /bar http://elsewhere/feh/bar
为路径加上别名
Alias /doc /usr/doc
非常重要，指明 CGI 脚本的虚拟路径对应的实际路径。一般所有的 CGI 脚本都要放在实际路径里，用户访问执行时输入站点+虚拟路径+CGI 脚本名
ScriptAlias /cgi-bin/ /var/www/cgi-bin/
用户可以根据自己需要，对 boa.conf 进行修改，但必须要保证其他的辅助文件和设置必须和 boaconf 里的配置相符，不然 Boa 就不能正常工作。
最后将修改好的 boa.conf 放在开发板上的/etc/boa 目录下。
在上面的例子中，我们还需要创建日志文件所在目录/var/log/boa，
创建 HTML 文档的主目录/var/www，将 mime.types 文件拷贝到/etc 目录，
创建 CGI 脚本所在目录/var/www/cgi-bin/。
mime.types 文件用来指明不同文件扩展名对应的 MIME 类型，一般可以直接从 Linux 主机上拷贝一个，大部分也都是在主机的/etc 目录下，同样，将主机的 mime.types 文件拷贝到开发板上的/etc 目录下。