文档分类

- 红盟公告
- 新手上路
- 漏洞分析
- 工具介绍
- 安全配置
- 编程技术
- **安全技术** <=
- 业界动态

本类热门

- 如何利用黑客技术跟踪并分析一名目标人物
- 有关无线安全与黑客的话题
- 让我们回归(by 默难)
- 通用ShellCode深入剖析[转]
- Format String 漏洞介绍/总结(一)
- 结合实例解读ELF文件-阅读笔记(1)

Apache mod_ssl ssl_util_uuencode_binary buffer over 分析和调试方法

提交文章： bkbll
提交日期： 2004-06-03
文章属性： 原创
文档类别： 安全技术
浏览次数： 今4次/总966次

Apache mod_ssl ssl_util_uuencode_binary buffer over 分析和调试方法
ver 1.0
(bkbll#cnhonker.net, http://www.cnhonker.net 2004/06/03)

1. 前言
最近有一些linux厂商说要升级apache的公告,具体问题是出在mod_ssl的ssl_util_uuencode_binary函数
CVE:http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0488
buftraq: http://www.securityfocus.com/bid/10355
漏洞描述(from CVE):
Stack-based buffer overflow in the ssl_util_uuencode_binary function in ssl_util.c for Apache mod_s

configured to trust the issuing CA,
may allow remote attackers to execute arbitrary code via a client certificate with a long subject DN.
当时看到不是默认配置就有的,所以就没怎么看它. 后来看到论坛有人问一些相关信息, 又有邮件来问,
调试平台:Redhat Linux 默认安装,apache 2.0.40
[root@mobilelinux httpd]# rpm -qa|grep mod_ssl
mod_ssl-2.0.40-8

2. 漏洞成因:
这里:http://lists.netsys.com/pipermail/full-disclosure/2004-May/021610.html有比较详细的描述,我cop
+---+
in ssl_util.c there is:

```
void ssl_util_uuencode_binary(
unsigned char *szTo, const unsigned char *szFrom, int nLength, BOOL bPad)
{
const unsigned char *s;
int nPad = 0;

for (s = szFrom; nLength > 0; s += 3) {
*szTo++ = ssl_util_uuencode_six2pr[s[0] >> 2];
/*PROPOSED PATCH: add "if (--nLegth ==0 ) ..." */
*szTo++ = ssl_util_uuencode_six2pr[(s[0] << 4 | s[1] >> 4) & 0x3f];
if (--nLength == 0) {
nPad = 2;
break;
}
*szTo++ = ssl_util_uuencode_six2pr[(s[1] << 2 | s[2] >> 6) & 0x3f];
if (--nLength == 0) {
nPad = 1;
break;
}
```

```
*szTo++ = ssl_util_uuencode_six2pr[s[2] & 0x3f];
--nLength;
}
while(bPad && nPad--)
*szTo++ = NUL;
*szTo = NUL;
return;
}
```
+--+

我们来看一下szTo和szFrom的定义:
In ssl_engine_kernel.c:

```
int ssl_hook_UserCheck(request_rec *r)
{
SSLConnRec *sslconn = myConnConfig(r->connection);
SSLSrvConfigRec *sc = mySrvConfig(r->server);
SSLDirConfigRec *dc = myDirConfig(r);
char buf1[MAX_STRING_LEN], buf2[MAX_STRING_LEN];
char *clientdn;
const char *auth_line, *username, *password;

.....................

apr_snprintf(buf1, sizeof(buf1), "%s:password", clientdn);
ssl_util_uuencode(buf2, buf1, FALSE);

apr_snprintf(buf1, sizeof(buf1), "Basic %s", buf2);

.................

}
```

In ssl_util.c:

```
void ssl_util_uuencode(char *szTo, const char *szFrom, BOOL bPad)
{
ssl_util_uuencode_binary((unsigned char *)szTo,
(const unsigned char *)szFrom,
strlen(szFrom), bPad);
}
```

从调用关系就可以看出来,szTo和szFrom其实就是ssl_hook_UserCheck里面的buf2,buf1
大小为:MAX_STRING_LEN,来看看这个大小:

```
[root@DUMPLOGIN E:\download\linux\httpd-2.0.40]# grep MAX_STRING_LEN include/* -r

include/httpd.h:#define MAX_STRING_LEN HUGE_STRING_LEN
include/mpm_common.h:extern char ap_coredump_dir[MAX_STRING_LEN];

[root@DUMPLOGIN E:\download\linux\httpd-2.0.40]#grep HUGE_STRING_LEN include/* -r

include/httpd.h:#define MAX_STRING_LEN HUGE_STRING_LEN
include/httpd.h:#define HUGE_STRING_LEN 8192
```

大小就是0x2000

那么从ssl_util_uuencode_binary很容易看出,szTo可以大概被覆盖(0x2000/3).
我们来看看调用时候内存结构图:

低地址 --> 高地址

```
|---[buf2]---|---[buf1]---| somedata |pointer | pointer | pointer | saved ebp | saved eip |------
```

buf2既然可以被覆盖大概0x2000/3字节,那么他能覆盖且只能覆盖到buf1,而且还不能完全覆盖buf1.
当调用ssl_util_uuencode_binary时候,buf2和buf1之间同样没有其他变量可以覆盖,所以想在这里覆盖
不大可能,我们只能希望于调用结束后能利用buf2做其他的事情,
但我们来看:

```
apr_snprintf(buf1, sizeof(buf1), "%s:password", clientdn);
ssl_util_uuencode(buf2, buf1, FALSE);

apr_snprintf(buf1, sizeof(buf1), "Basic %s", buf2);
ap_log_error(APLOG_MARK, APLOG_INFO, 0, r->server,"Faking HTTP Basic Auth header: \"Autho
return DECLINED;
```

一直到函数返回,buf2除了用做写到buf1的apr_snprintf函数一个参数外,没其他用处.
结论:在x86系统上基本没什么大作用,连让进程崩溃都不大可能.

3. 调试方法
这里就涉及到很多关于httpd.conf配置和ssl配置方面的东西,本来很简单,但有人问起来就写下吧,其实写
本文章的主要目标.
调试系统:redhat linux 8.0 x86默认安装.
(1). 修改服务端配制,主要要点在于修改ssl.conf打开信任CA等相关开关.
在此之前你需要一个ca.pl文件,该文件在各版本openssl tgz包中都有.
[root@DUMPLOGIN E:\download\linux]#dir "openssl-0.9.7d\apps\CA.pl"
驱动器 E 中的卷是 DOCUMENT
卷的序列号是 F015-E42A

E:\download\linux\openssl-0.9.7d\apps 的目录

2004-03-17 08:08p 5,392 CA.pl
1 个文件 5,392 字节
0 个目录 2,449,305,600 可用字节
[root@mobilelinux ca]# ls
ca.pl
[root@mobilelinux ca]# pwd
/root/ca
[root@mobilelinux ca]# ./ca.pl -newca
CA certificate filename (or enter to create)

Making CA certificate ...
Using configuration from /usr/share/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.....++++++
....++++++
writing new private key to './demoCA/private/cakey.pem'
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [GB]:
State or Province Name (full name) [Berkshire]:guangdong
Locality Name (eg, city) [Newbury]:guangzhou
Organization Name (eg, company) [My Company Ltd]:www.my.com
Organizational Unit Name (eg, section) []:test
Common Name (eg, your name or your server's hostname) []:10.10.10.114
Email Address []:bkbll@cnhonker.net
[root@mobilelinux ca]# ls
ca.pl demoCA
[root@mobilelinux ca]# ./ca.pl -newreq
Using configuration from /usr/share/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.............++++++
..++++++
writing new private key to 'newreq.pem'
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [GB]:
State or Province Name (full name) [Berkshire]:guangdong
Locality Name (eg, city) [Newbury]:guangzhou
Organization Name (eg, company) [My Company Ltd]:www.my.com
Organizational Unit Name (eg, section) []:test
Common Name (eg, your name or your server's hostname) []:homelinux

Email Address []:bkbll@tom.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Request (and private key) is in newreq.pem
[root@mobilelinux ca]# ./ca.pl -sign
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName :PRINTABLE:'GB'
stateOrProvinceName :PRINTABLE:'guangdong'
localityName :PRINTABLE:'guangzhou'
organizationName :PRINTABLE:'www.my.com';
organizationalUnitName:PRINTABLE:'test'
commonName :PRINTABLE:'homelinux'
emailAddress :IA5STRING:'bkbll@tom.com'
Certificate is to be certified until Jun 2 11:17:20 2005 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Signed certificate is in newcert.pem
[root@mobilelinux ca]# ls
ca.pl demoCA newcert.pem newreq.pem
[root@mobilelinux ca]# openssl rsa < newreq.pem > newkey.pem
read RSA key
Enter PEM pass phrase:
writing RSA key
[root@mobilelinux ca]# ls
ca.pl demoCA newcert.pem newkey.pem newreq.pem
[root@mobilelinux ca]# mv newcert.pem server_cert.pem
[root@mobilelinux ca]# mv newkey.pem server_key.pem
[root@mobilelinux ca]# mv newreq.pem server_req.pem
然后将拷贝几个文件(demoCA/cacert.pem, server_cert.pem,server_key.pem,server_req.pem)
到随便那个目录,我的是/etc/httpd/conf/ssl/目录.
然后需要修改/etc/httpd/conf.d/ssl.conf文件:

SSLCertificateFile /etc/httpd/conf/ssl/server_cert.pem
SSLCertificateKeyFile /etc/httpd/conf/ssl/server_key.pem
SSLCACertificatePath /etc/httpd/conf/ssl.crt /* 这个目录本来就有,就指定这个吧 */
SSLCACertificateFile /etc/httpd/conf/ssl/cacert.pem
SSLVerifyClient require
SSLVerifyDepth 10
SSLOptions +FakeBasicAuth /* 这个一定要 */

修改/etc/httpd/conf/httpd.conf:
加这个:

<Directory /var/www/html/usage>
SSLVerifyClient require
SSLVerifyDepth 5
SSLCACertificateFile /etc/httpd/conf/ssl/cacert.pem
SSLCACertificatePath /etc/httpd/conf/ssl.crt
SSLOptions +FakeBasicAuth
SSLRequireSSL
AuthName "test mod_ssl overflow site"
AuthType Basic
AuthUserFile /etc/httpd/conf/httpd.passwd /* 这个文件随便指定 */
require valid-user
</Directory>

修改后,重新启动apche
(2) 触发漏洞的client配置(IE)
在linux上给客户端生成一个证书:
[root@mobilelinux ca]# ./ca.pl -newreq
Using configuration from /usr/share/ssl/openssl.cnf

```
Generating a 1024 bit RSA private key
.....................++++++
..++++++
writing new private key to 'newreq.pem'
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:
State or Province Name (full name) [Berkshire]:guangdong
Locality Name (eg, city) [Newbury]:guangzhou
Organization Name (eg, company) [My Company Ltd]:www.my.com
Organizational Unit Name (eg, section) []:test
Common Name (eg, your name or your server's hostname) []:dumplogin
Email Address []:dumplogin@yahoo.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Request (and private key) is in newreq.pem
[root@mobilelinux ca]# ./ca.pl -sign
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName :PRINTABLE:'GB'
stateOrProvinceName :PRINTABLE:'guangdong'
localityName :PRINTABLE:'guangzhou'
organizationName :PRINTABLE:'www.my.com';
organizationalUnitName:PRINTABLE:'test'
commonName :PRINTABLE:'dumplogin'
emailAddress :IA5STRING:'dumplogin@yahoo.com'
Certificate is to be certified until Jun 2 11:26:58 2005 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Signed certificate is in newcert.pem
[root@mobilelinux ca]# openssl pkcs12 -export -in newcert.pem -inkey newreq.pem -name "MY CEI

demoCA/cacert.pem -out mycert.p12
Enter PEM pass phrase:
Enter Export Password:
Verifying password - Enter Export Password:
[root@mobilelinux ca]# ls
ca.pl mycert.p12 newreq.pem server_key.pem
demoCA newcert.pem server_cert.pem server_req.pem
[root@mobilelinux ca]#

将mycert.p12拷贝到windows下,安装该证书.
将newcert.pem拷贝到/etc/httpd/conf/ssl.crt/(SSLCACertificatePath /etc/httpd/conf/ssl.crt) 重命名为.c
将ssl.crt/目录下的Makefile.crt重命名为Makefile
运行make,会自动生成一个link文件.
再将这个crt的内容添加到cacert.pem文件(SSLCACertificateFile /etc/httpd/conf/ssl/cacert.pem)
cat newcert.crt >> /etc/httpd/conf/ssl/cacert.pem
重新启动apache.

(3). 调试工具.
linux下当然用gdb, windows下我用windbg,以方便server端能attach进程.
(4). 开始调试
启动浏览器,配置一下,将  使用ssl2,ssl3,tls1.0 全部勾上.
然后打开windbg, F6 attach到该浏览器, 设置断点:bp ws2_32!send
```

然后bd 0,暂时禁止断点.
从浏览器上输入linux机器的地址(注意要是刚才在httpd.conf里定义的保护站点,就是usage目录),如果以
出来一个选

择证书的对话框:
选择刚才linux颁发的证书,不要点确定哦,回到windbg,按ctrl+break,be 0,启动断点,g,这个时候回到
IE窗口,点确定,ok,IE挂起来了.
回到linux,在root用户下寻找到处理该请求的进程ID:
[root@mobilelinux ca]# netstat -antp |grep ":443"|grep ESTABLISHED
tcp 0 0 10.10.10.114:443 10.10.10.111:4625 ESTABLISHED 2563/httpd
[root@mobilelinux ca]#
PID 2563就是我们所要的.
[root@mobilelinux ca]# gdb -q -se /usr/sbin/httpd
(no debugging symbols found)...(gdb) attach 2563
Attaching to program: /usr/sbin/httpd, process 2563
Reading symbols from /usr/lib/libz.so.1...done.
Loaded symbols for /usr/lib/libz.so.1
Reading symbols from /lib/libssl.so.2...done.
.....

Loaded symbols for /usr/lib/liblber.so.2
Reading symbols from /usr/lib/libsasl.so.7...done.
Loaded symbols for /usr/lib/libsasl.so.7
Reading symbols from /lib/libnss_nisplus.so.2...done.
Loaded symbols for /lib/libnss_nisplus.so.2
0x420d224b in poll () from /lib/i686/libc.so.6
(gdb)
然后输入断点:
(gdb) b *ssl_hook_UserCheck
Breakpoint 1 at 0x4096a0e0
(gdb) b *ssl_util_uuencode
Breakpoint 2 at 0x40975cc0
(gdb) b *ssl_util_uuencode_binary
Breakpoint 3 at 0x40975d10
这个时候就已经不需要windbg设置的断点了,bd 0,g
再回到刚才的linux窗口:
(gdb) c
Continuing.
[Switching to Thread 8192 (LWP 3331)]

Breakpoint 1, 0x4096a0e0 in ssl_hook_UserCheck ()
from /etc/httpd/modules/mod_ssl.so
(gdb) c
Continuing.

Breakpoint 2, 0x40975cc0 in ssl_util_uuencode ()
from /etc/httpd/modules/mod_ssl.so
(gdb) x/4wx $esp
0xbfffb88c: 0x4096a229 0xbfffb8d0 0xbfffd8d0 0x00000000
(gdb) x/bs 0xbfffd8d0
0xbfffd8d0:
"/C=GB/ST=guangdong/L=guangzhou/O=www.my.com/OU=test/CN=10.10.10.114/Email=dumplogi
(gdb) x/bx 0xbfffd8d0
0xbfffd8d0: 0x2f
(gdb) c
Continuing.

Breakpoint 3, 0x40975d10 in ssl_util_uuencode_binary ()
from /etc/httpd/modules/mod_ssl.so
(gdb)

剩下的事情就不需要我多说了,enjoy it :)

4. 参考:
[1]. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0488
[2]. http://www.modssl.org/docs/2.8/ssl_howto.html
[3]. http://www.drh-consultancy.demon.co.uk/pkcs12faq.html
[4]. http://www.redhat.com/docs/manuals/stronghold/Stronghold-3.0-Manual/admin-guide/chapter2.f
[5]. http://www.freebsddiary.org/openssl-client-authentication.php

返回上页

Copyright © 2000-2004 **HUC** All Rights Reserved.
中国红客网络技术联盟 www.cnhonker.com
webmaster@cnhonker.net