
INIT.D EN OPSTARTPROCESSEN

SOFTWARESECURITY

DOOR MICK BEER

ONDERZOEK NAAR DE WERKING VAN INIT.D

Introductie

In dit rapport onderzoek ik de werking van het init.d bestand en wat deze in gang zet met betrekking tot het opstartproces van het apparaat waar de firmware op draait.

Mick Beer Kampen, 16 februari 2022

Inhoudsopgave

| | | |
|----------|--|-----------|
| 1 | Introductie | 1 |
| 2 | Statische analyse | 3 |
| 2.1 | Bestandstype | 3 |
| 2.2 | Inhoud van rcS | 3 |
| 2.3 | Opvallende programma 's | 4 |
| 2.4 | Rechten van scripts | 5 |
| 2.5 | Telnet | 5 |
| 3 | Het boot proces | 6 |
| 3.1 | verband met init.d | 6 |
| 3.2 | Kernel executie | 6 |
| 4 | Eigenschappen en vereisten van init | 7 |
| 4.1 | Aanwezigheid | 7 |
| 4.2 | Verbinding rcS en boot | 7 |
| 4.3 | \$1 en \$2 | 8 |
| 4.4 | Samengevat | 8 |
| 5 | Webservers en init.d in de praktijk | 9 |
| 5.1 | linux bestandsmachtigingen | 9 |
| 5.2 | webserver | 9 |
| 6 | Bibliografie | 10 |
| 7 | Bibliografie | 10 |

Over dit bestand

Dit bestand is enkel voor educatieve doeleinden.

Statische analyse

2.1 Bestandstype

Door het commando `binwalk` werd duidelijk dat `rcS` in `init.d` een executable script is met decimal 0 wat starting positie is, initialisatie. Het bestand is geschreven dmv Bash scripting wat te herkennen is aan de `#!`. Dit wordt shebang genoemd en verteld het operatingsysteem welke interpreter gebruikt moet worden om de rest van het bestand te verwerken.

```
(kali㉿kali)-[~/../_DIR890LA1_FW110b07.bin.extracted/squashfs-root/etc/init.d]
$ file rcS
rcS: POSIX shell script, ASCII text executable

(kali㉿kali)-[~/../_DIR890LA1_FW110b07.bin.extracted/squashfs-root/etc/init.d]
$ binwalk rcS
```

| DECIMAL | HEXADECIMAL | DESCRIPTION |
|---------|-------------|---------------------------------------|
| 0 | 0x0 | Executable script, shebang: "/bin/sh" |
| 19 | 0x13 | Unix path: /etc/init.d/S??* ;do |
| 164 | 0xA4 | Unix path: /etc/init0.d/rcS |

Figuur 1: Binwalk output van `rcS`

2.2 Inhoud van `rcS`

Ook wordt er verwezen naar het bestand `/unit.d/S` en `/unit0.d/rcS`. Door het commando `binwalk` werd duidelijk dat `rcS` in `init.d` een executable script is met decimaal 0 wat starting positie is, waar verder in het onderzoek dieper op ingegaan wordt. Het bestand is geschreven dmv Bash

```
(kali㉿kali)-[~/../_DIR890LA1_FW110b07.bin.extracted/squashfs-root/etc/init.d]
$ cat rcS
#!/bin/sh
for i in /etc/init.d/S??* ;do
    # Ignore dangling symlinks (if any).
    [ ! -f "$i" ] && continue
    # Run the script.
    echo "$i"
    $i
done
echo "$0] done!"
/etc/init.d/rcS
```

Figuur 2: Inhoud van rcS

scripting wat te herkennen is aan de “!”. Dit wordt shebang genoemd en verteld het operating-systeem welke interpreter gebruikt moet worden om de rest van het bestand te verwerken.

2.3 Opvallende programma's

Door de bestanden uit te lezen die in dezelfde directory staan met opvallende namen zoals mydlink (de naam van het apparaat) kwam de output van figuur 3 naar voren. Ook de rechten van de bestanden zijn onderzocht. De directory init.d staat vol met bash-scripts die aangeroepen worden tijdens het booting proces

```
(kali㉿kali)-[~/../_DIR890LA1_FW110b07.bin.extracted/squashfs-root/etc/init.d]
$ cat rcS
#!/bin/sh
for i in /etc/init.d/S??* ;do
    # Ignore dangling symlinks (if any).
    [ ! -f "$i" ] && continue
    # Run the script.
    echo "$i"
    $i
done
echo "$0] done!"
/etc/init.d/rcS

(kali㉿kali)-[~/../_DIR890LA1_FW110b07.bin.extracted/squashfs-root/etc/init.d]
$ cat S10init.sh
#!/bin/sh
mount -t proc none /proc
mount -t ramfs ramfs /var
mount -t sysfs sysfs /sys
mount -t usbfs usbfs /proc/bus/usb
echo 7 > /proc/sys/kernel/printk
echo 1 > /proc/sys/vm/panic_on_oom

(kali㉿kali)-[~/../_DIR890LA1_FW110b07.bin.extracted/squashfs-root/etc/init.d]
$ cat S22mydlink.sh
#!/bin/sh
MYDLINK=`cat /etc/config/mydlinkmtd`
#domount=`xmldb -g /mydlink/mtdagent`
domount=`mfc mount_mydlink state`
if [ "$domount" = "on" ]; then
    mount -t squashfs $MYDLINK /mydlink
fi
```

Figuur 3: Mydlink programma

2.4 Rechten van scripts

De rechten van de scripts die aangeroepen worden door het `init.rcS` proces zijn onderzocht en worden weergegeven in figuur 4.

```
(kali㉿kali)-[~/../_DIR890LA1_FW110b07.bin.extracted~/squashfs-root/etc/init.d]
$ ls -al
total 52
drwxrwxr-x  2 kali kali 4096 May 24 2016 .
drwxrwxr-x 15 kali kali 4096 May 24 2016 ..
-rwxr-xr-x  1 kali kali  181 May 24 2016 rcS
-rwxr-xr-x  1 kali kali  190 May 24 2016 S10init.sh
-rwxr-xr-x  1 kali kali  101 May 24 2016 S12ubs_storage.sh
-rwxr-xr-x  1 kali kali  129 May 24 2016 S15udevd.sh
-rwxr-xr-x  1 kali kali  688 May 24 2016 S16ipv6.sh
-rwxr-xr-x  1 kali kali  225 May 24 2016 S19init.sh
-rwxr-xr-x  1 kali kali  764 May 24 2016 S20init.sh
-rwxr-xr-x  1 kali kali   36 May 24 2016 S21usbmount.sh
-rwxr-xr-x  1 kali kali  194 May 24 2016 S22mydlink.sh
-rwxr-xr-x  1 kali kali   20 May 24 2016 S23udevd.sh
-rwxr-xr-x  1 kali kali  121 May 24 2016 S45gpiod.sh
```

Figuur 4: Bestandsrechten

2.5 Telnet

De server service Telnet heeft de volgende inhoudt;

[illegible]

Figuur 5: Telnet inhoud

3

SECTION

Het boot proces

3.1 verband met init.d

/etc/init.d is een essentieel onderdeel van het bootproces. Wanneer de bootloader geexecute wordt, de bootloader een POST uitvoert, de kernel image decomprest naar de main ram en de kernel execute met `init= ...` default is `etcpreinit/sbin/init`.

3.2 Kernel executie

Hierna start de kernel de scan naar mtd partitie rootfs voor een superblock en mount deze squashFS partitie (waar etc inzit). Etc/preinit doet pre-initializatie setup, creëert directories, mounts fs /proc /sys..Daarna mount de kernel andere partities onder rootfs. Als INITRAMFS niet gedefinieerd is wordt sbinit (mother of all processes) geroepen. Als laatst wordt de kernel thread de userspace init proces. De userspace start wanneer de kernel de rootFS mount en het eerste programma wordt gestart wat normaliter sbinit is. De link tussen applicatie en kernel is de clib en syscalls die hierbij horen. Init leest de inhoud van etc/init/tab voor de sysinit entry, dit is normaliter `::sysinit:/etc/init.d/rcS S boot`). Hierna roept init etcinit.drcS S boot. RcS voert de symlinks om de daadwerkelijke startupscripts die in `etc/rcd/S##` met de optie "start". Nadat rcS klaar is met de processen moet het systeem volledig draaiend zijn.

Eigenschappen en vereisten van init

Er zijn een aantal essentiële eigenschappen en vereisten voor het init proces zodat dit goed zijn werking kan doen.

4.1 Aanwezigheid

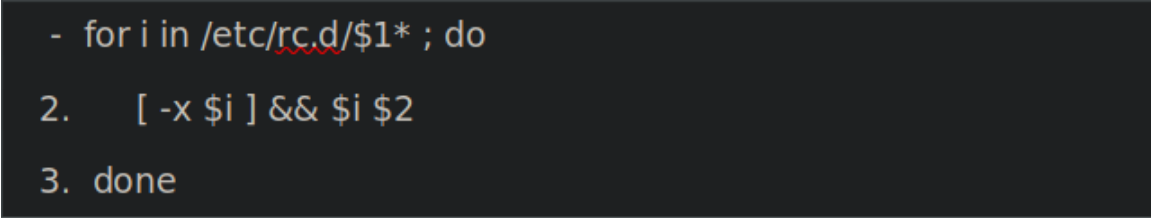
`/bin/init` moet in het filesystem aanwezig zijn om te kunnen booten omdat deze essentiële processen in gang zet om het besturingssysteem te laten draaien. `/etc/init.d/rcS` dient de juiste rechten om executable te zijn te hebben en `#!/bin/sh` als eerste lijn in het bestand. Nadat het filesystem is gemount door de kernel is `rcS` nodig.

4.2 Verbinding `rcS` en boot

`/etc/init.d/rcS` zorgt dat programmas tijdens het booten gestart en gedraait worden. Dit wordt normaliter gebruikt om filesystems te mounten (enkel de root is gemount op dit punt). Ook wordt het gebruikt om daemons te starten voor bijvoorbeeld usb ports, netports of bluetooth. Meestal is een `rcS` een shellscript dat gemakkelijk bewerkt kan worden. Het is gewoonlijk dat `rcS` scripts geschreven worden dat `rcS` scripts andere `/etc/rcS.d` scripts aanroepen en executen als een domino effect. Hierdoor is het goed mogelijk dat elke daemon zijn eigen init script heeft. Dit heeft twee kanten, aan de ene kant kan dit het isoleren van errors mogelijk maken. Als 1 script faalt zullen de andere scripts blijven draaien. Maar er zijn meerdere files die beheert moeten worden op kwetsbaarheden waardoor het kwetsbaarheid oppervlak groter is. De file `/etc/rc.local` wordt uitgevoerd door `rcS` als dit aanwezig is. Dit is bedoeld zodat de systeem-administrator de commands handmatig kan beheren.

4.3 \$1 en \$2

Sinds computers op zichzelf niet slim zijn, maar door grote rekenkracht slimme krachtige berekeningen uit kunnen voeren, is dat het zelfde met rcS script. Het is op zichzelf een simpel script qua functie. Het doel van het script is om alle scripts in de etcrc.d directory te starten met de juiste opties/configuraties. Sinds rcS wordt geroepen met de "S" en "boot" optie zal het rcS script 1 met "S" *vervangen* 2 met "boot". Hier de representatie



```
- for i in /etc/rc.d/$1* ; do
2. [ -x $i ] && $i $2
3. done
```

Figuur 6: Representatie

4.4 Samengevat

- Voer de volgende eenmaal voor elke entry (file/ling) in /etc/rc.d/ directory die begint met een "S".
- Als de file executable is, execute de file met de optie "boot"
- Herhaal stap 1, vervang \$i met de volgende bestandsnaam tot er geen bestanden meer zijn om te controleren.

Webservers en init.d in de praktijk

5.1 linux bestandsmachtigingen

In tegenstelling tot Microsoft-programma's gebruikt Linux bestandsmachtigingen in plaats van bestandsnaamextensies om aan te geven of deze vermelding uitvoerbaar is of niet. Voor een uitleg van bestandsmachtigingen, zie "man chmod" op een Linux/Unix machine over uitleg voor machtigingen en uitvoerbare bestanden. Kijk naar de map `/etc/rc.d` dan zie je dat sommige scripts koppelingen hebben die essentieel zijn voor het opstarten, maar geen afsluiting (d.w.z. `/etc/init.d/httpd`), terwijl sommige andere geen opstartscript hebben, maar wel een afsluitscript (d.w.z. `/etc/init.d/umount`).

5.2 webserver

In het geval van `httpd` (de webserver) maakt het niet uit of het script stopt of niet, er is niets op te ruimen voordat u stopt. Aan de andere kant MOET het `umount`-script worden uitgevoerd voordat het wordt afgesloten om ervoor te zorgen dat alle gegevens naar de media worden gespoeld voordat alle relevante opslagmedia worden ontkoppeld, anders kan gegevensbeschadiging optreden. Het is niet nodig om `umount` aan te roepen bij het opstarten, omdat de montage van opslagmedia ergens anders wordt afgehandeld (zoals `/etc/preinit`), dus er is geen opstartscript voor deze. Nadat het laatste opstartscript is uitgevoerd, moet u een volledig operationeel OpenWrt-systeem hebben.

Referenties

- [1] NOVI Hogeschool, en Wiersma, A. (2022, januari). Handout-opgaven. A. Wiersma.
- [2] . (z.d.). Understanding the rc Scripts in Linux. Thegeekdiary. Geraadpleegd op 15 februari 2022, van <https://www.thegeekdiary.com/understanding-the-rc-scripts-in-linux/>
- [3] Why is rcS required after file system is mounted by the kernel? (2012, 19 november). Unix and Linux Stack Exchange. Geraadpleegd op 15 februari 2022, van <https://unix.stackexchange.com/questions/56075/why-is-rcs-required-after-file-system-is-mounted-by-the-kernel>
- [4] eißhaupt, M. (2021, 10 december). The Boot Process. OpenWrt Wiki. Geraadpleegd op 15 februari 2022, van <https://openwrt.org/docs/techref/process.boot>