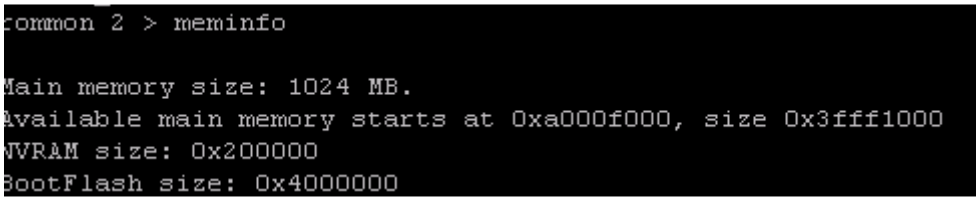
Recovering startup config from Rommon:

We can recover the startup configuration of a device from Rommon prompt as well.

Just came across this. Might be useful in troubleshooting cases related to rommon.

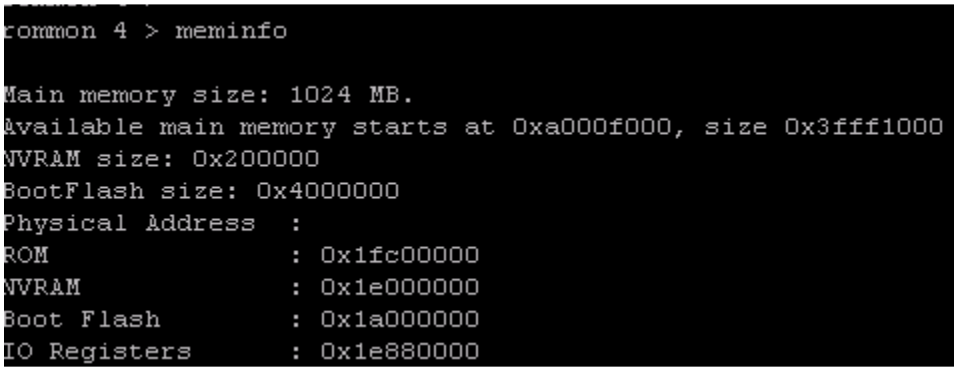
Recovering startup config from sup720 rommon:
==========

++ mem info output before entering priv.



++ meminfo output after entering priv command will give us the physical address details of ROM, NVRAM and IO Registers.

Rommon 3 > Priv



++ For going into the correct NVRAM address we have to make sure that we are in the SP rommon.

```
Software clause at DFARS sec. 252.227-7013.

           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706



Cisco IOS Software, s72033_sp Software (s72033_sp-ADVIPSERVICESK9_WAN-M)
n 12.2(33)SXI6, RELEASE SOFTWARE (fc4)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2011 by Cisco Systems, Inc.
Compiled Mon 28-Mar-11 12:37 by prod_rel_team
Image text-base: 0x40101328, data-base: 0x42321580


*** System received an abort due to Break Key ***
signal= 0x3, code= 0x0, context= 0x50009798
PC = 0x417bc480, Cause = 0x2020, Status Reg = 0x34008002
Exit at the end of BOOT string
```

++ Once we are in the SP rommon Execute dump command to read the NVRAM data.

rommon 6 > dump 0x1e000000 0x1e3000000 à where 0x1e000000 is the starting address of the NVRAM where the startup config would be stored.

After executing the command, you would see something like startup config.

Sample output:
==========

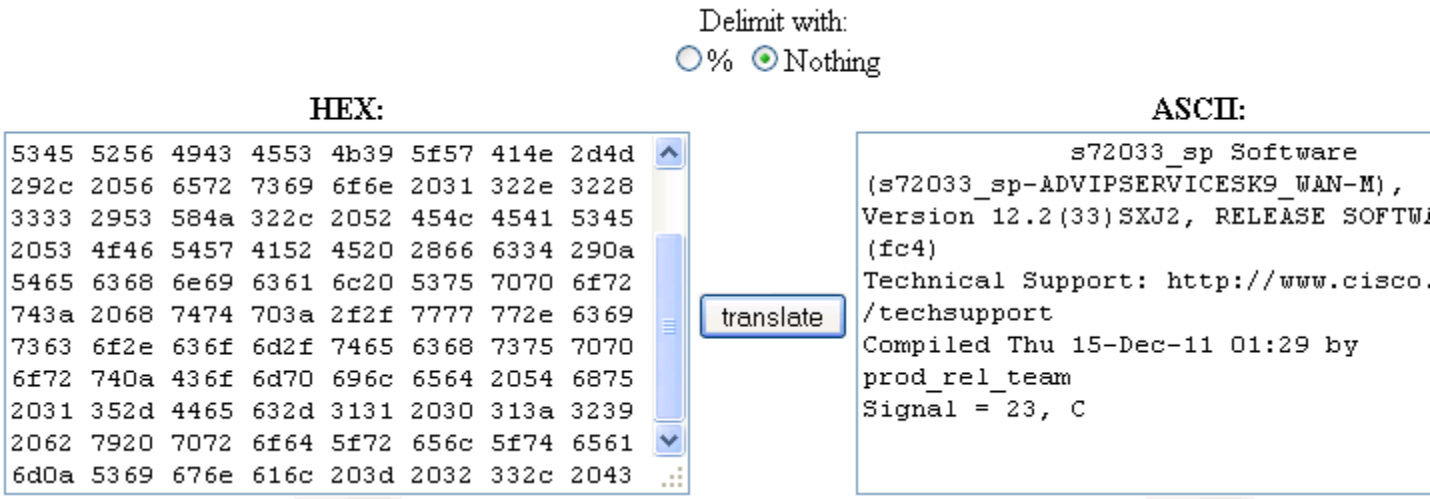
1e0000e0 4c4f 475f 5052 4546 4958 5f56 4552 5349 LOG_PREFIX_VERSI
1e0000f0 4f4e 0031 0052 4554 5f32 5f52 5453 0030 ON.1.RET_2_RTS.0
1e000100 363a 3330 3a32 3720 5554 4320 5475 6520 6:30:27 UTC Tue
1e000110 4465 6320 3620 3230 3131 0042 4f4f 5400 Dec 6 2011.BOOT.

++ Once we know the startup config start address and end address filter the nvram data using the same dump command again.

++ After which we need to remove the Hex address, in this case it is the first column(1e0000e0, 1e0000f0) and convert the rest of the values from hex to ascii.

I had used idea2ic (hex to ascii tool) for decoding this. There are many similar tools available online.

- http://www.idea2ic.com/PlayWithJavascript/hexToAscii.html



The cookie command

This command shows hardware info of the specific Cisco device such as PCB version, product identifier, and RMA. As a note, each Cisco hardware has his own cookie and if they are not the right ones there is a cookie check against hardware that need to be validated by the starting ROM.

Issue of having incorrect cookie info is getting error message of something like bad software or like the following.

Failed Authentication Test. This router may not be a genuine Cisco product.
FAILED: Cookie signature verification failed, status = 540

To illustrate the cookie command, following is the command output comes from 877 router

rommon 1 > cookie

The PRIV password depends from hardware cookie:

password := (i1+...+i5) mod 2^16

where i1...i5 first five words in cookie

Also this features working on 1600,3600,7500

Cisco 3640:

System Bootstrap, Version 11.1(20)AA2, EARLY DEPLOYMENT RELEASE SOFTWARE (fc1)
Copyright (c) 1999 by cisco Systems, Inc.

rommon 1 > cookie

cookie:
00 01 00 03 e3 bd 0d 40 0a ff ...
rommon 2 > priv
Password: fc00
You now have access to the full set of monitor commands.

```
Warning: some commands will allow you to destroy your
configuration and/or system images and could render
the machine unbootable.
rommon 3 >
```

Cisco 7513:

```
System Bootstrap, Version 11.1(2) [nitin 2], RELEASE SOFTWARE (fc1)
Copyright (c) 1994 by cisco Systems, Inc.
SLOT 6 RSP2 is system master
RSP2 processor with 131072 Kbytes of main memory

monitor: command "boot" aborted due to user interrupt
rommon 1 > priv
You now have access to the full set of monitor commands.
Warning: some commands will allow you to destroy your
configuration and/or system images and could render
the machine unbootable.
rommon 2 >
```

This priv command is useful when you need to change the cookie info on the Cisco hardware due to some unexpected change during lightning storm or similar. Note that you need to have sufficient understanding of machine language (Assembler) and lots of leg work such as studying Cisco hardware info samples, checking PCB printed code and serial number labels.