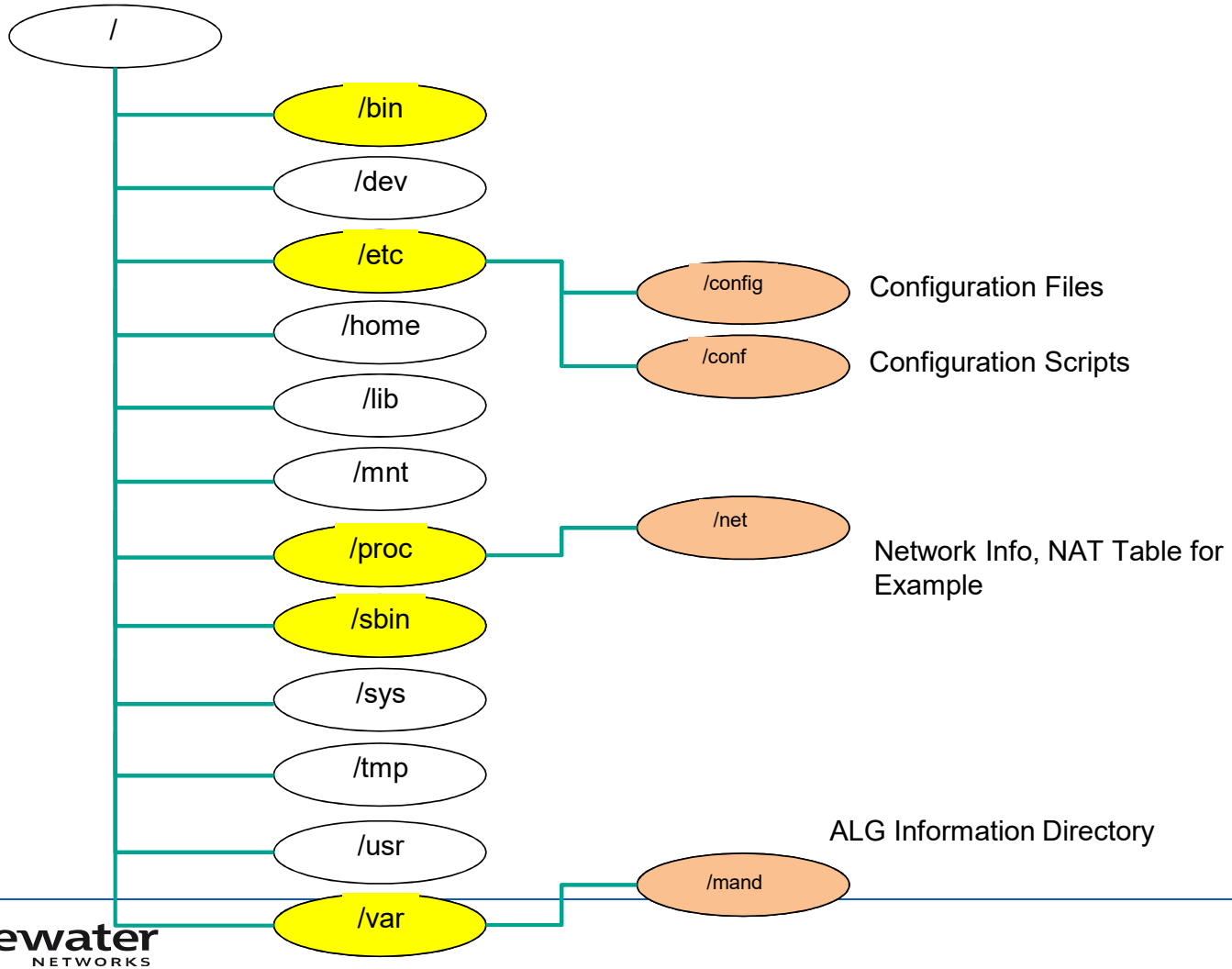
EdgeMarc CLI commands

Notes on Software CLI (cont'd)

- Linux based file system running on flash
- Most of the file system is read only in user mode; this means that shell scripts can only be run under the guidance of EWN Systems Engineers or TAC Engineers
- Bash shell is used when logged in – helpful
- *keeps command history and tab for complete commands similar to Cisco/Adtran*
- A good number of needed unix utilities are available but this is not a redhat system.
- CLI is used mostly for config review, check and also troubleshooting
- CLI can be used for making specific configuration changes, but not all configuration changes
- During login session, history is kept of commands. History is lost when logging out
- TACACS Login or Local username/password will be used

EdgeMarc VOS Directory Structure Tour



Important Directories and Files Cont.

Directory/File	Purpose
/var/	Writeable directory, but not copied to flash.
/var/log/	All EM log files. `messages` and `messages.old` are primary files. Rotate at 32KB.
/var/asterisk/	Survivable mode control files.

/var/asterisk/
extensions.conf
	Contains Local-mode URI aliases. See:
	`[sip-client-set]`
	`exten => 4646,1,Dial(SIP/7202664646@7202664646,15)`
	`exten => 2664646,1,Dial(SIP/7202664646@7202664646,15)`
	`exten => 7202664646,1,Dial(SIP/7202664646@7202664646,15)`

/var/mand/	ALG status files
/var/mand/calls	Number of active (WAN-side) calls. Same as System->System Information **Number of Active Streams:**

Important Directories and Files Cont.

Filename	Purpose
*.conf1 & *.conf2	Input/Output of **ewn save** command. Only one of each filetype allowed. Can `cat` to/from these files.
sipclients.2	SIP Clients Table.
resolv.conf	DNS addresses used by other services. Updated by dynamic wan protocols.

Viewing the Interface Configurations

- Checking the IP addresses of the LAN and WAN ports
  - # **ifconfig**

```
eth0      Link encap:Ethernet  HWaddr 54:39:68:01:23:0E
          inet addr:1.1.1.3  Bcast:1.1.1.255  Mask:255.255.255.0
          inet6 addr: fe80::5639:68ff:fe01:230e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12393803 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12463775 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
              Interrupt:16 Memory:f7d00000-f7d20000

eth1      Link encap:Ethernet  HWaddr 54:39:68:01:23:0F
          inet addr:12.183.130.141  Bcast:12.183.130.143  Mask:255.255.255.240
          inet6 addr: fe80::5639:68ff:fe01:230f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:21168739 errors:0 dropped:0 overruns:0 frame:0
          TX packets:20979794 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
              Interrupt:17 Memory:f7c00000-f7c20000

eth2      Link encap:Ethernet  HWaddr 54:39:68:01:23:10
          inet addr:10.128.1.141  Bcast:10.128.3.255  Mask:255.255.252.0
          inet6 addr: fe80::5639:68ff:fe01:2310/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4193948 errors:0 dropped:0 overruns:0 frame:0
          TX packets:926 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
              Interrupt:18 Memory:f7b00000-f7b20000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1654 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1654 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

Viewing DHCP Leases

- Viewing the status of DHCP leases
  - All times in this file are in UTC (GMT), not your local timezone.
  - This is not a bug. There is no portable way to store leases in the local timezone.

```
# cd /etc/config
# more dhcpd.leases
<< dhcpd.leases >>
lease 192.168.2.15 {
        starts 2 2008/10/14 06:48:18;
        ends 5 2008/10/17 06:48:18;
        hardware ethernet 00:a0:d1:d0:9b:01;
        uid 01:00:a0:d1:d0:9b:01;
        client-hostname "bill";
}
lease 192.168.2.12 {
        starts 1 2008/10/13 14:46:21;
        ends 4 2008/10/16 14:46:21;
        hardware ethernet 00:04:f2:19:ca:ec;
}
lease 192.168.2.11 {
        starts 1 2008/10/13 14:46:16;
        ends 4 2008/10/16 14:46:16;
        hardware ethernet 00:04:f2:18:54:97;
```

DHCP and NAT status

- To see if the DHCP Server is running
  - `# ps -ef | grep dhcpd`

```
24309 root       640 S N /bin/dhcpd -q eth0.1 eth0.100 eth0.200
25151 root       452 S N grep dhcp
```

Verification of dynamic NAT (port/address)

```
# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.2.252:1025      0.0.0.0:*               LISTEN
tcp        0      0 192.168.2.1:5060        0.0.0.0:*               LISTEN
tcp        0      0 99.148.27.5:5060        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 192.168.2.252:7001      0.0.0.0:*               LISTEN
tcp        0      0 192.168.2.252:7002      0.0.0.0:*               LISTEN
tcp        0      0 192.168.2.252:7003      0.0.0.0:*               LISTEN
tcp        0    138 99.148.27.5:23          135.89.202.138:62986    ESTABLISHED
```

ARP Commands

To see a complete listing of ARP filters, type

- `# arp -h`

The following filters can be applied:

```
# arp -a  display (all) hosts in alternative (BSD) style
# arp -s --set            set a new ARP entry
# arp -d --delete         delete a specified entry
# arp -v --verbose        be verbose
# arp -n --numeric        don't resolve names
# arp -i --device         specify network interface (e.g. eth0)
# arp -D --use-device       read <hwaddr> from given device
# arp -A -p, --protocol     specify protocol family
# arp -f --file           read new entries from file or from /etc/ethers
```

To flush the entire ARP cache:

```
# ip -s -s neigh flush all
```

Viewing the SIP clients table

 # cd etc/config
 # cat sipclients.2

```
* 192.168.2.252 1025 6304474938 uri=sip:6304474938@192.168.2.252:1025 ndm=0
user=6304474938 nat=0 trn=0 trp=0 blaport=5060

* 192.168.2.16 5060 6304474942 uri=sip:6304474942@192.168.2.16:5060 ndm=0
user=6304474942 nat=0 trn=1 trp=0 blaport=5060

* 192.168.2.11 5060 6304474941 uri=sip:6304474941@192.168.2.11:5060 ndm=0
user=6304474941 nat=0 trn=1 trp=0 suburi=sip:SD8ts86-
meccfempfbv5t0sqkvm2p4ji24k6hvgi096hfq92fugh9jpfhi0o0ohg6ohg00ho120mqobp82plsffoa
ofndsnk4o4u7e066qv4e50000107@12.194.226.140:5060;att=HATTCENTREX-
5cfhslaerassa;transport=udp subuser=SD8ts86-
meccfempfbv5t0sqkvm2p4ji24k6hvgi096hfq92fugh9jpfhi0o0ohg6ohg00ho120mqobp82plsffoa
ofndsnk4o4u7e066qv4e50000107 blaport=5060

* 192.168.2.14 5060 6304474938-81754940
uri=sip:6304474938@192.168.2.14:5060;user=phone;transport=udp ndm=0
user=6304474938 nat=0 trn=1 trp=0 blaport=5060

* 192.168.2.12 5060 6304474944 uri=sip:6304474944@192.168.2.12:5060 ndm=0
user=6304474944 nat=0 trn=1 trp=0 blaport=5060

* 192.168.2.252 1025 6304474943 uri=sip:6304474943@192.168.2.252:1025 ndm=0
user=6304474943 nat=0 trn=0 trp=0 blaport=5060
```

Debugging… debugged

- When troubleshooting a case, it is optimal to pursue the following debug scenarios:
- Packet capture (tcpdump) using the ‘any’ interface flag
- Output of “sys_report” command
- FXS related issue: All of the above and the fxs debug
- PRI related issue: All of the above and the PRI debug
- Further debug might be required
  - ALG/B2BUA logging
  - Output of system-related commands

CLI Command Reference - VoIP

Command	Purpose
# mandctl log 0x9 / # mandctl log 0	Turns on ALG debugging for VOIP issues / Turns off debugging; note, mand logs are not meant to be perpetual!
# sys_report	Outputs entire running configuration to text on screen
# cat /var/mand/calls	Displays active calls on system
# cat /var/mand/sipservers_status	Displays SIP Server status as reflected on Survivability GUI page - lost messages, received messages, processing state (Active, Idle, Unreachable)
# cat /var/mand/sipservers_data	Displays SIP server status as reflected on GUI (IP Address, Port, Priority)

CLI Command Reference – System Troubleshooting

Command	Purpose
# ifconfig	Displays all configured interfaces
# ewntmsctl –d	Displays T1 status
# ping address | name	icmp testing
# arp	Displays arp status / connected MAC addresses
# traceroute	Conducts icmp traceroute
# cat /proc/net/ip_conntrack | wc –l	Displays count of all NAT'd connections
# cat /var/log/messages # cat /var/log/messages.old	Displays message file up to 32K Displays messages rotated after 32K
# cat /var/log/messages # cat /var/log/messages.old	Displays message file up to 32K Displays messages rotated after 32K

CLI Command Reference – System Management

Command	Purpose
# cat /etc/version	Displays VOS version on system
# cat /etc/platform	Displays hardware type
# cat /var/ewn_model	Displays model of system
# ewn save filename.conf1 or .conf2	Saves running configuration to a file that can be exported off of the system
# cfg_commit	Saves any configuration change made to the CLI to the static configuration
# config_network	Restarts network services
# cfg_restore	Restores system to Comcast default manufacturing configuration

Additional Useful System Debugging command (sys_report) for TAC and capturing info

'sys_report' is a new command that provides useful debug output. Usage is: # **sys_report**

This command combines automates certain scripts and output from a multitude of commands, and provides the following

- Hardware info (/etc/platform)
- Firmware Version info (/etc/version)
- Date/Time stamp
- memory info (/proc/meminfo)
- Process info (ps –ef)
- File System info (df)
- CPU info (top)
- Active/configured interface info (ifconfig)
- Routing table display (route)
- certain mand files (tinfo, xml_info)
- VPN info and logs
- Kernel and OS output info (dmesg)
- message log output (var/log/messages and var/log/messages.old)
- Running configuration (mandctl dbg replay = /var/replay.cfg)

Intro to the packet capture function

- Because the EdgeMarc acts as a passive network probe, it utilizes the native linux 'tcpdump' command to sniff traffic on all of it's interfaces. This command allows the operator to passively monitor packet activity going through the interfaces

- **# tcpdump –i <interface>**
- where <interface> is *eth0* for the LAN side, *eth1* for the WAN side, *hdlc0* for T1, etc. Refer to 'ifconfig' output to better suit your environment.
- Other options for 'tcpdump':

`# tcpdump -h` gives all the options available with flags

this will filter out SSH traffic from the LAN and/or WAN trace by source or destination IP address

```
# tcpdump -ni any -s 0 host xxx.xxx.xxx.xxx
```

This is the best script to capture, as it does not truncate the output and is good for printing to a pcap file:

```
# tcpdump -ni any -s 0 not port 22
```

This capture grabs only the well known SIP signaling port:

```
# tcpdump -ni any -s 0 port 5060
```

This capture grabs SIP signaling and Survivability ports:

```
# tcpdump -ni any -s 0 port 5060 and port 5050
```

This capture grabs SIP signaling and DNS, useful troubleshooting call processing related issues:

```
# tcpdump -ni any -s 0 port 5060 or port 53
```

Saving a packet capture to a viewable Wireshark file

The EdgeMarc has the ability to store packet captures locally on the system, for later FTP to view in Wireshark.

It is important to realize that there is a finite amount of storage space, no more than 16M, so you must exercise caution when running packet captures.

To create a temporary storage file on the EdgeMarc, take the following steps:

```
# mkdir /var/ramdisk
# mount -t tmpfs tmpfs -o size=8M /var/ramdisk
```

After you have created the temporary storage space (in volatile memory), then you can execute any variation of the tcpdump command and save the output to the ramdisk. Conduct this while replicating or tracking the issue/scenario:

```
# cd /var/ramdisk
# tcpdump -ni any -s 0 port 5060 or port 53 -w test.pcap
tcpdump: WARNING: Promiscuous mode not supported on the "any" device
tcpdump: listening on any

34 packets received by filter
```

Enter CTRL c to stop traces on each window

ALG/B2BUA diagnostics script

- The more information on the scenario involved the better the framework is to troubleshoot
- Basic SIP call control and RTP debug:
  - # mandctl log 0x9 (covers most SIP call control and RTP issues)
  - # mandctl log 0x1000009 (SIP + STATE + INFO, used primarily for HA/Stateful Failover debug)
  - # mandctl log 0x05000009 (SIP + Survivability, used when debugging Remote to Local Mode operation)

Debugging the mand (ALG/B2BUA) process

The new command is:

```
# mandctl log <level> [file_%TIME%.txt[size]]
```

- 'level' is the required logging level.
- 'file' if given is the main file to log to.
- 'size' if given, is the maximum size of the log file.
- %TIME, if given, provides a timestamp of the file

Rules of Engagement:

- The rotation will keep one old file (file.old) so it is important to ensure that the 'size' is no more than half the available ramdisk space.
- For example, if you create an 8M ramdisk, you should not give a size larger than 4M, and to be safe, allow some extra space
- Numbers must be used, not abbreviations or suffixes, i.e., 3500000 instead of 4M, 500K, etc).

Debugging the mand (ALG) process - application

```
# mkdir /var/ramdisk
# mount -t tmpfs tmpfs -o size=8M /var/ramdisk
# mandctl log 0x9 /var/ramdisk/test_%TIME%.txt 5000000
```

- This command will create a text file of the mandlog output; additionally it writes useful ALG configuration parameters that will be used for analysis by Engineering
- Because mand logging is taxing, you must always disable mandlogging after you have captured the problem scenario

```
# mandctl log 0
```

Common Cisco to EdgeMarc Command Matrix

Cisco Command	Edgemarc Command
show service-module T1 s0/0/0	Network stats in GUI or **ewntmsctl -d**
show service-module T1 s0/0/0 performance-statistics	Advanced page of stats in GUI **or ewntmsctl -D 0**
wr mem	/etc/conf/bin/**cfg_commit**
copy run start	**ewn save** <file name.conf1 or .conf2>
show version	GUI, System Page or **cat /etc/version**. Uptime can be obtained from GUI or uptime commad
show diag	**cat /var/ewn_model** or cat /**etc/platform**
show run	view file created by the **ewn save** command via the more command **mandctl dbg replay** then retrieve file /**var/replay.cfg**
copy start run	**ewn load <file name>** followed by /etc/conf/bin/cfg_commit
show ip interface brief	**ifconfig -a** For built in switch ports, mii-tool, or use Network page of GUI
show log	**more /var/log/messages**
show clock	**date**
show arp	**arp -a** from CLI
show ip route	**route** or **netstat -rvn** from CLI
copy tftp flash (for upgrade purposes)	Use GUI Upgrade function or **netflash –kiH** (node manager ip) (image name)
copy run tftp	**ewn upload <filename> <ip address>**
show voice port sum	**cat /var/mand/uaHookstate** (FXS Ports) or **cat /var/mand/calls** (total active calls)
show diag	**cat /var/ewn_model**
show mem sum	**cat /proc/meminfo**
show proc cpu	**top**
show ip nat translations	**cat /proc/net/ip_conntrack**
show crypto iskmp | ipsec sa	**/etc/conf/ipsec/ipsec auto --status**