ctys-uc-WoL(7)
Wake on Lan

November 29, 2010

Contents

List of Figures

.

1 General

The Wake-on-LAN - WoL - feature enables the remote wake-up of machines by sending of a **MagicPacket(TM)**.
The PM plugin supports the most - probably all - practical cases of WoL .
In order to function the NIC on the target machine has to be setup properly before shutdown(of the wake-up target).
This enables a volatile register on the NIC, which is - as far as known - not stored persistently, thus requires a continued stand-by power.
The packets are processed on Ethernet-Layer, therefore require in the most common case a segment specific broadcast. This requires some specific cases for remote-wake-up.
Some specific requirements arise from the execution structure of **CREATE** for the **PM** plugin. This is inherent to the circumstance, that the target of PM as a runtime-container is not available before the successful execution - at least the first step - of **CREATE** of the WoL-target. The specific case here is the fact, that the PM requires it's managed target as a runtime container for itself. In companion with the restricted permission to some tools like ether-wake, a **gateway-host** for execution of the CREATE method is required, which could be the localhost itself.
The basic instances involved into processing of WoL are:

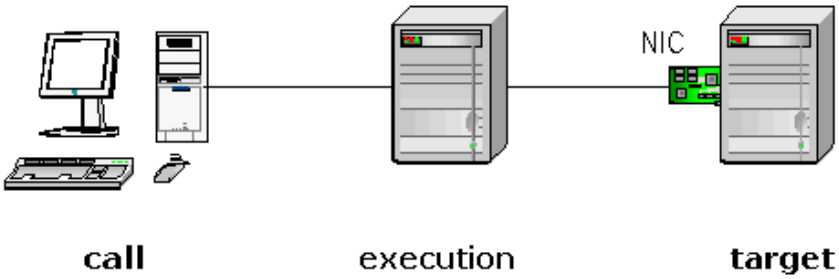


Figure 1: Structure of WoL configuration

The NIC on <target> requires the following pre-initialization, which is (seems) to be stored in a volatile RAM powered by stand-by power, thus has to be reinitialized when <target> is powered-off.

```
ethtool -s <ifX> wol umbg

with Wake On:

  u - unicast messages
  m - multicast messages
  b - broadcast messages
  g - MAgicPacket(tm)
```

REMINDER: The interface to be addressed is the physical interface receiving the required packets. This could vary from the standard naming schema. For example in case of standard Xen the name of the physical interface name of original "eth0" is altered to "peth0". Anyhow, some implementations might bypass the register settings to physical interfaces, this has to be verified for each execution platform.
The resulting usage of WoL with ctys calls only requires the following calls for wake-up of the off-line, but stand-by powered <WoL-target>.

1. Switch the hostX into stand-by mode - executed on the WoL target.

   ```
   ctys -t pm -a cancel=wol,poweroff hostX
   ```

2. Activate the hostX from stand-by mode - executed on the WoL initiator.

   ```
   ctys -t pm -a create=wol,t:hostX,broadcast:192.168.3.255
   ```

- t:hostX
  Could be any valid <machine-address>, e.g. "m:macX" or "l:labelX".
- broadcast:192.168.3.255
  Send a broadcast UDP to port 9 on given address. This has to be supported by the router. When not present a broadcast to the local subnet only is sent. Alternatively the final router of target subnet could be configured for redirection of a specific address/port to a local brooadcast. For example for "192.168.3.251" on port "4711", the following could be used: "broadcast:192.168.3.251:4711".

The following Use-Cases comprise some supported and verified scenarios for WoL application. These scenarios show up one of the main pitfalls within a SSO environment, where e.g. a kerberos based "ksu" does not work for any case.

2 PREREQUISITES

The required prerequisites due to the required system tools and access permissions should be validated with the utility **ctys-plugins**.
The WoL feature is based on the following pre-requirements:

- Support of the NIC for WoL, an "active sleep mode"
- Support of the Motherboard, namely the BIOS, and if present the OnBoard-NIC for WoL.
  Depending from the bus design and/or on-board LAN, e.g. the "PME" feature for the PCI bus is required for NIC cards, and additionaly a WoL feature. Both options are usually located within the Power Management settings of the BIOS.
  The flags frequently found in APM mask of BIOS are:
  - PME Events
  - Wake On LAN - Event
  - Wake On Modem/Ring - Event
- A machine record for ENUMERATE with full scope of data should be generated on the WoL-target as root user by usage of the tool **ctys-genmconf**.
- The caching database with at least a MAC mapping generated from dhcpd.conf by tool **ctys-extractMAClst** or **ctys-extractARPlst** has to be present.
  Alternatively a manual entry for the destination machine could be added manually. For the format refer to **ctys-extractMAClst**.
  Preferably the full database in addition should be generated by usage of **ctys-vdbgen**, which is a collector for running systems. Therefore for bootstrap MAC mapping is required and sufficient.
- The pre-shutdown setting of a volatile flag on the NIC, which activates the autonomous "active sleep mode" of the card/controller-chip(-set).
  This could be established persistently by adding it to an init script. Alternatively ctys CANCEL action could be used only for "WOL" option, simulating persistency by refreshing the setting on each CANCEL.
  The setting is only stored as long as the NIC is in stand-by mode, e.g. the S5 state of ACPI, thus lost when the Powersupply of the machine is actually switched off for standard elements. Anyhow, solutions may exist, which store the attribute in a non-volatile storage.

The tools required for ctys are:

- ethtool, for setting the NIC to WoL.
- netcat for sending the WoL packet to remote segments.
- ether-wake for sending the WoL packet to local segment.
- halt, poweroff, shutdown, and reboot for cancel action.

The permissions required are:

- root permission for ethtool for write access.
- root permission for halt, poweroff, shutdown, and reboot, when no wrapper like the consolehelper of CentOS/RHEL is present.

The configuration of ksu and/or sudo should be preferred for granting access permissions, the change of file permissions should be avoided.

2.0.1 Configuration of Access Permissions

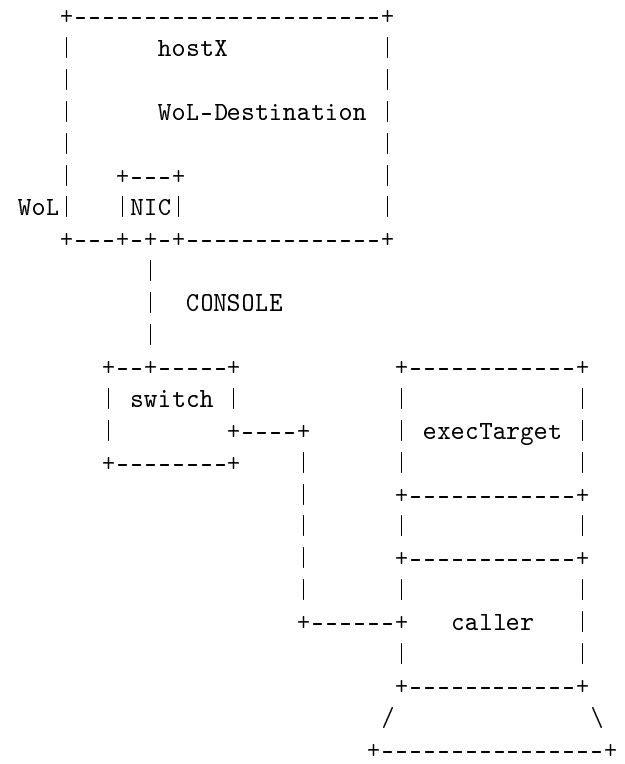
The plugin PM has the key role for the application of WoL, thus is required to be setup properly.
Two locations of execution are accessed for a complete boot, thus both required to be prepared.

- caller site: Access to network resources for sending MAC packets is required.
- wake-up target:

The NIC has to be prepared - obviously before shutdown. This requires permissions for the CANCEL action of PM including the required system resources.
.

3 WoL with Bridge-less Configuration

A single NIC as unique remote access port in a production environment. No bridges are present.
Anything works quite straight forward. Due to the required root-permissions any shutdown-utility could be used with local and network based users.



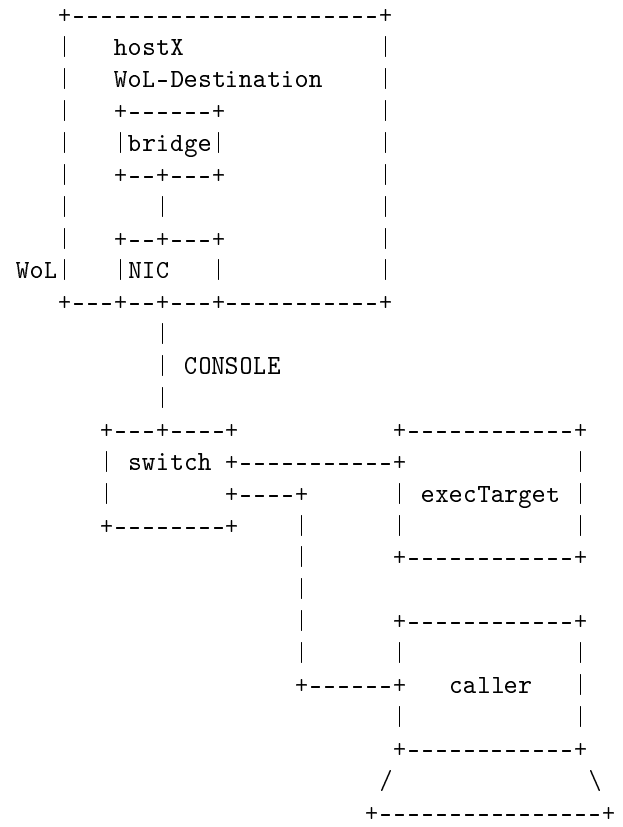
Bridge-less WoL configuration within same Ethernet Segment

Figure 2: "Bridge-less" WoL configuration within same Ethernet Segment

```
ctys -t pm -a create=wol,t:<hostX> -Z NOSUDO '(-d 99 -Z KSU)'

ctys -t pm -a cancel=wol,self,stack,poweroff:0 -Z KSU <hostX>'(-d 99 -w -Z SUDO)'
```

4 WoL with a Virtual-Bridge

A single bridged NIC as unique remote access port in a production environment.
In this scenario things become somewhat complicated. The main reason for now is, that in case of a bridging a physical NIC, the "wol g" option of ethtool is in some cases not correctly set on the physical nor the logical NICs beeing a member of a bridge.
Thus a "stop" of the bridge is required previously to setting WoL, which - at least temporarily - disconnects "<hostX>" from the network. And therefore from the authentication services as well as from the required runtime-utilities. More severe - from NFS too!
Thus for safety this is supported for local users only.



WoL configuration with a virtual bridge

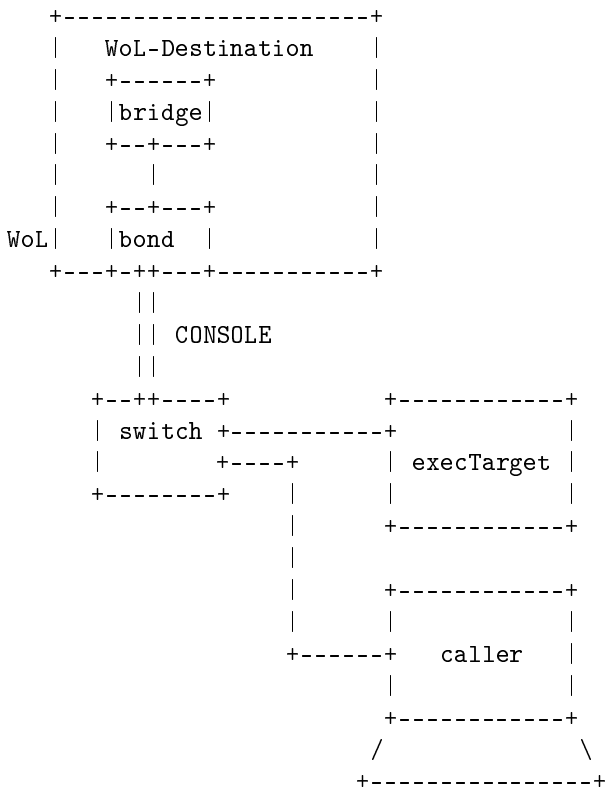
Figure 3: WoL configuration with a virtual bridge

```
ctys -t pm -a create=wol,t:<hostX> -Z NOSUDO

ctys -t pm -a cancel=wol,self,stack,poweroff:0 -Z NOKSU <hostX>'(-Z SUDO)'
```

5 WoL with NIC-Bonding on the Target

A bridged bonded NIC group as unique remote access port in a production environment.
The same restrictions as for 2.).

```
ctys -t pm -a create=wol,t:<hostX> -Z KSU <execTarget>'(-Z SUDO)'

ctys -t pm -a cancel=wol,self,stack,poweroff:0 -Z NOKSU  <hostX>'(-Z SUDO)'
```



WoL configuration with a virtual bridge on a bond device

Figure 4: WoL configuration with a virtual bridge on a bond device

6 WoL for a Target Behind a Router

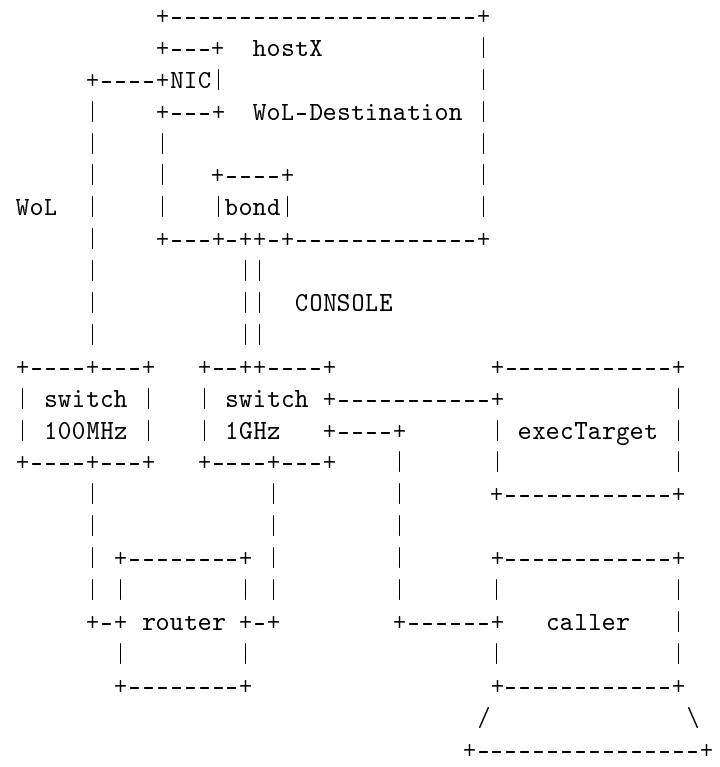
A bridged bonded NIC group as non-WoL remote access port in a production environment. Additionally a specific management NIC with WoL support.
In this case restrictions to local-only users apply only, when the WoL-NIC is used as access port. In any case no local restrictions apply to network-users with mounted HOMEs.
But instead some configuration of network equipment is required in order to pass the required messages through involved routers.
A typical scenario where a remote broadcast is required even in the local LAN.

```
ctys -t pm -a create=wol,t:<hostX>,broadcast:192.168.3.255 -Z KSU <execTarget>'(-Z KSU)'

ctys -t pm -a cancel=wol,if:eth2,self,stack,poweroff:0 -Z KSU  <hostX>'(-Z KSU)'
```

The reason for this scenario is the restriction of the NIC as beeing a recipient for WoL packets. This is defined so by the Motherboard/BIOS. But using a Gigabit port for multiple of those machines was not welcome. Thus the good-old 100MHz device, which is actually a c1538m, was used for WoL purposes only. It is used for a printerserver anyway, and had enough unused ports. The router in this case is a OpenBSD custom-made router, offering pass-through of directed-broadcasts, thus anything works fine.

7 CLI - Cycle for WoL

This halts the machine with preparation of WoL.



WoL configuration for a NIC behind a router

Figure 5: WoL configuration for a NIC behind a router

In this case sudo is used for raising permissions on the target machine where the sudoers is configured with **requiretty**, thus the **-z 2** parameter is required.

```
ctys -t pm -a cancel=self,force,wol,poweroff:0,timeout:0 -z 2 wol1@lab01'(-Z sudo)'
```

The following call switches the machine "t:192.168.1.72" on and opens a CLI console within the callers window. Therefore the option **-z 2** is required twice and **-b 0** once. This is due to the requirement of a pty for the reasons:

- **local:** The remote call requires a pty for the sudo call, which has to be set locally when preparing the ssh-call. This is the default configuration for 'sudoers'.
- **remote:** The remote call requires a pty for chained login to the target machine, once this is up and operational. The CLI of ssh requires in the test environment a pty.

The CLI requires in each case a synchronous forground mode by **-b 0**.

```
ctys -t pm -a create=wol,t:192.168.1.72,sshping:wol1,console:cli -z 2 -b 0 lab00'(-Z sudo -z 2)'
```

8 XTERM - Cycle for WoL

This halts the machine with preparation of WoL.

In this case sudo is used for raising permissions on the target machine where the sudoers is configured with **requiretty**, thus the **-z 2** parameter is required.

```
ctys -t pm -a cancel=self,force,wol,poweroff:0,timeout:0 -z 2 wol1@lab01'(-Z sudo)'
```

The following call switches the machine "t:192.168.1.72" on and opens a **gnome-terminal** window with a **bash**. Therefore the option **-z 2** is required for the execution of **ethtool** by sudoers, which is configured with **requiretty**.

```
ctys -t pm -a create=wol,t:192.168.1.72,sshping:wol1,console:xterm -z 2 lab00'(-Z sudo)'
```

9 GTERM - Cycle for WoL

This halts the machine with preparation of WoL.
In this case sudo is used for raising permissions on the target machine where the sudoers is configured with **requiretty**, thus the **-z 2** parameter is required.

```
ctys -t pm -a cancel=self,force,wol,poweroff:0,timeout:0 -z 2 wol1@lab01'(-Z sudo)'
```

The following call switches the machine "t:192.168.1.72" on and opens a **gnome-terminal** window with a **bash**. Therefore the option **-z 2** is required for the execution of **ethtool** by sudoers, which is configured with **requiretty**.

```
ctys -t pm -a create=wol,t:192.168.1.72,sshping:wol1,console:gterm -z 2 lab00'(-Z sudo)'
```

10 EMACS -Cycle for WoL

This halts the machine with preparation of WoL.
In this case sudo is used for raising permissions on the target machine where the sudoers is configured with **requiretty**, thus the **-z 2** parameter is required.

```
ctys -t pm -a cancel=self,force,wol,poweroff:0,timeout:0 -z 2 wol1@lab01'(-Z sudo)'
```

The following call switches the machine "t:192.168.1.72" on and opens a **gnome-terminal** window with a **bash**. Therefore the option **-z 2** is required for the execution of **ethtool** by sudoers, which is configured with **requiretty**.

```
ctys -t pm -a create=wol,t:192.168.1.72,sshping:wol1,console:emacs -z 2 lab00'(-Z sudo)'
```

If Emacs does no start with a shell check your emacs version, your environment and your configuration first. Try the X11 plugin with **CONSOLE:EMACS** next.

11 VNC - Cycle for WoL

This halts the machine with preparation of WoL.

In this case sudo is used for raising permissions on the target machine where the sudoers is configured with **requiretty**, thus the **-z 2** parameter is required.

```
ctys -t pm -a cancel=self,force,wol,poweroff:0,timeout:0 -z 2 wol1@lab01'(-Z sudo)'
```

The following call switches the machine "t:192.168.1.72" on and opens a **gnome-terminal** window with a **bash**. Therefore the option **-z 2** is required for the execution of **ethtool** by sudoers, which is configured with **requiretty**.

```
ctys -t pm -a create=wol,t:192.168.1.72,sshping:wol1,console:vnc -z 2 lab00'(-Z sudo)'
```

12 SEE ALSO

ctys-IPMI(7), *ctys-PM(7)*, *ctys-uc-PM(7)*, *ctys-wakeup(1)*, *ether-tool(8)*, *ether-wake(8)*, *nc(1)<a.k.a. netcat>*, *ethers(5)*

13 AUTHOR

Maintenance:	<acue_sf1@sourceforge.net>
Homepage:	<http://www.UnifiedSessionsManager.org>
Sourceforge.net:	<http://sourceforge.net/projects/ctys>
Berlios.de:	<http://ctys.berlios.de>
Commercial:	<http://www.i4p.com>

14 COPYRIGHT