Centralized Backup & Restore of NETWITNESS PLATFORM Version 11.2+ A Wrapper for nw-recovery-tool (NRT)

UPDATED FOR VERSION 12.X (BACKWARD COMPATIBLE TO 11.X)

SCENARIO -

Need to remotely backup your NetWitness hosts to a central location, to satisfy Disaster Recovery Requirements, perform a Tech Refreshes, or to be prepared for RMA replacement of a device.

SOLUTION – A WRAPPER FOR NRT

Building off the framework of the original nw-backup scripts written for 10.x backup/restore and migration to 11.x, a new set of version 11/12 scripts has been written as a "wrapper" to the built in Netwitness Recovery Tool (NRT) functionality of NetWitness Since Version 11.2 was released.

(Please note that this is not an officially supported solution by NetWitness Support, but can be used by customers as a possible backup solution, at their own risk)

OVERVIEW –

The solution consists of 5 scripts (all run from the NW Admin server (node-zero)), supporting files for custom feed backup fix and an example **nw-base.nrt** file (for backing up non-netwitness related OS files):

- **get-all-systems.sh** – generates an inventory file of all hosts in the NW server deployment.
- **ssh-propagate.sh** – Generates an ECDSA key pair on the NW Server and propagates the public key to all hosts.
- **nw-backup.sh** – The main backup script.
- **nw-restore.sh** – Restore Script for RMA/Tech Refresh.
- **cf-fix.sh** – Custom feed backup fix (used for applying fix to warm standby server)
- **nw-base.nrt** – An NRT script to add any custom files to the backup directory as part of the backup process.
- **feedfix folder** – Contains the backup.sh and restore.sh scripts to support the backup of custom feed files on the admin servers.

PROCESS WALKTHROUGH –

Copy the attached zip file to the NW Admin Server host (node-zero) and unzip:

```
mkdir /root/scripts
unzip nw-backup.zip -d /root/scripts
cd /root/scripts
chmod +x *.sh
```

Step 1: Run get-all-systems.sh

```
./get-all-systems.sh
```

```
Usage:

 ./get-all-systems.sh [ -u <USER> [ -g group][ -p <PASSWORD> ][-d /home/path ]][ -b </backup/path> ][-n
<hrs>][-h]

Configuration Options
Note: All command-line options are optional.
      With no options selected, the script will use options set within the file itself.

-b </backup/path> : Path to store the all-systems file and logs. Default:(/var/netwitness/nw-backup)
-u <username>     : User acct to use for non-root SSH access to hosts during backups.  Default:(root)
-p <password>     : Password for <username>   Default: (ask)
-g <group>        : Alternate Group to add user to (or create). Default: (username)
-d </home/path>   : Base home directory path (/home) for <username> Default:(/var/netwitness/nw-backup)
-n <hrs>          : Skip check for new systems if current new-systems is less than <hrs> old.
-h                : Print this help information.
```

Run on the NW Admin Server. Creates the **/var/netwitness/nw-backup** directory (or the directory specified with the -b option), then using a combination of mongo and salt queries, will create the **all-systems** file in that directory. The **all-systems** file is used by the other scripts, with entries that contain the following in comma separated format:

DeviceType,Hostname,IPAddress,MinionID,SerialNumber

```
DeviceType   = NRT Category Type (i.e. AdminServer,Broker,Concentrator,Decoder, etc.)
Hostname     = Short hostname (not FQDN)
IPAddress    = Management Interface IP address of host
MinionID     = Unique Salt MinionID of host
SerialNumber = Device Serial Number for Reference and Support
```

Example:

```
AdminServer,nw-admin,192.168.1.129,70f95dc0-3cb6-4fd4-b9f2-ac923d0ba594,PK10T51
ESAPrimary,nw-esa,192.168.1.131,a598cb6b-4bd2-4ba2-af6a-79df3dab35e6,R9L8LNM
Broker+Search,nw-broker,192.168.1.130, 2a83b597-7970-4872-b76f-109cb591fa90,CSZ77X2
LogHybrid,nw-loghyb,192.168.1.133,87fc872c-68e3-45e3-9108-e30f847dc14e,PK10T0A
Malware,nw-malware,192.168.1.132,2c98e425-57a0-47d2-82d7-15795a6165f5,R90BCFWP
NetworkHybrid,nw-nethyb,192.168.1.134,9a99294e-3889-48b0-9555-11d3c21e2018,R90218K6
```

The script is designed to run from the cron on a regular basis (if you are in a dynamic environment where systems are added/removed on a regular basis), it has a 30 second timeout on the one question it asks.

If changes to the environment are detected, generation of **new-systems** and/or **old-systems** files will occur. The **new-systems** file can be used by other scripts for running specific targeted actions against the newly installed hosts. If a system is "offline" or has been removed from the UI, the old-systems file will have the entries for those hosts, so information about them is not lost.

Step 2: Run ssh-propagate.sh

./ssh-propagate.sh

```
Configuration options
Note: All command-line options are optional.
    Run with no options, script uses the /var/netwitness/nw-backup/all-systems file to target
    all hosts and copy the root users ecdsa-521 bit public ssh key to all hosts, if key does
    not exist, it is generated prior to propagation.

Usage:

./ssh-propagate.sh [ -u <user> ] [-c] [ -b <path> ] [ -t  <target> ]

-b <path>  : path to the location of all-systems file. Default: (/var/netwitness/nw-backup)
-u <user>  : User to propagate keys to on all nodes (must exist on NW Server). Default: (root)
-g <group> : Group associated with the user Default: (username)
-c         : Used with -u <user> [ -g <group>] for non-root user, creates user on remote hosts.
-t <target>: Target hosts (new-systems or anything greppable from all-systems. Default: (ALL)

Run on the NW Admin Server.  Performs the following (depending on options):
```

- Verifies the designated user account already exists on the NW Server
- Updates the /etc/hosts file with host shortnames using entries from all-systems file
- Generates an ecdsa-521bit ssh key for root (or the specified user) if it does not exist
- Iterates through the target list (default is ALL hosts in the all-systems file) and performs the following:
    - Tests ssh connectivity to the host (host responding on port 22)
    - Verifies the user exists on the remote host, if not and the -c option is selected
        - Creates remote user acct
        - Sets password to same as user acct on NW server
    - Tests SSH key authentication to host AS user specified, if auth fails,
    - Copies the user ecdsa public key string to the ~/.ssh/authorized_keys file
    - Adds the target host fingerprint to the user's ~/.ssh/known_hosts file on the NW server
    - Verifies ssh connectivity via ssh-key authentication

Step 3: Modify the nw-base.nrt file

Edit the supplied **nw-base.nrt** file

The default /etc/netwitness/recoverytool/nw-base.nrt file, distributed with the system, only contains the following entries:

```
name nw-base
directory /etc/netwitness/platform/nodeinfo
file /etc/machine-id

# unmanaged files
stash /etc/fstab
stash /etc/hosts
stash /etc/sysconfig/iptables
stash /root/.ssh

# for azure
stash /etc/krb5.conf
stash /etc/logrotate.d/waagent.logrotate
stash /etc/mdadm.conf
```

```
stash /etc/waagent.conf
Post 11.5, this line was added to the bottom of the nw-base.nrt file:

# only for nrt mode, run dnsclient to update the hosts configuration
after-import if [ "${NRT_MODE}" != 'standby' ]; then chef-client --config
/var/lib/netwitness/config-management/client.rb --logfile /var/log/netwitness/config-
management/chef-client.log --log_level info --runlist 'recipe[nw-dns-client]' >
/dev/null 2>&1; fi
```

The "STASH" entries are NOT restored during an NRT import (recovery), but are available for reference in the /var/netwitness/backup/unmanaged folder, after the import. The included nw-base.nrt file has an expanded list of stash files and directories to include files used at several customer installations. To Fully restore functionality, these files need to be available after the restore.

Extended **nw-base.nrt** supplied with the nw-backup scripts:

```
name nw-base
directory /etc/netwitness/platform/nodeinfo
file /etc/machine-id

# unmanaged files
stash /etc/fstab
stash /etc/hosts
stash /etc/resolv.conf
stash /etc/nsswitch.conf
stash /etc/passwd
stash /etc/shadow
stash /etc/group
stash /etc/sudo.conf
stash /etc/sudoers
stash /etc/sudoers.d
stash /etc/exports
stash /etc/krb5.conf
stash /etc/nfs.conf
stash /etc/ntp.conf
stash /etc/rsyslog.conf
stash /etc/logrotate.conf
stash /etc/sysconfig/network
stash /etc/sysconfig/nfs
stash /etc/sysconfig/iptables
stash /etc/sysconfig/iptables.bak
stash /etc/sysconfig/iptables-config
stash /etc/crontab
stash /etc/sysconfig/network-scripts/ifcfg-em1
stash /etc/sysconfig/network-scripts/ifcfg-em2
stash /etc/sysconfig/network-scripts/ifcfg-em3
stash /etc/sysconfig/network-scripts/ifcfg-em4
stash /etc/cron.hourly
stash /etc/cron.daily
stash /etc/cron.weekly
stash /etc/cron.daily
stash /etc/pam.d/netwitness
stash /etc/pam.d/securityanalytics
stash /etc/logrotate.d
stash /etc/logstash
stash /etc/multipath.conf
stash /etc/lvm
stash /etc/raddb
stash /etc/rsyslog.d
stash /etc/snmp
stash /etc/ssh
stash /var/ace
```

```
stash /home
stash /root/ (note: this backs up ALL file/folders in /root including /root/.ssh)
stash /var/netwitness/nw-backup/all-systems

# for azure
stash /etc/krb5.conf
stash /etc/logrotate.d/waagent.logrotate
stash /etc/mdadm.conf
stash /etc/waagent.conf

# only for nrt mode, run dnsclient to update the hosts configuration
after-import if [ "${NRT_MODE}" != 'standby' ]; then chef-client --config
/var/lib/netwitness/config-management/client.rb --logfile /var/log/netwitness/config-
management/chef-client.log --log_level info --runlist 'recipe[nw-dns-client]' >
/dev/null 2>&1; fi
```

Edit the file to include any additional locations (files or directories) that contain customizations in your deployment, then save the file in the same directory as the nw-backup.sh script. The backup script will verify the file on each system matches your modified file and if not, will automatically copy the modified file to each host before running NRT on that host.

Step 4: Run nw-backup.sh

./nw-backup.sh

```
Configuration Options:
Note: All command-line options are optional.
      With no options selected, the script backup ALL devices listed in the
      /var/netwitness/nw-backup/all-systems file (except any commented out with #)
      and copy the backup files to the /var/netwitness/nw-backup/<date>/ directory
      on the NW Server.

Usage:

./nw-backup.sh [-b <NRT path>] [-m <xfer mode>][-l <NFS mount point)>] [-p <NW backup path>]
[-s <remote server IP>][-d <dest path>] [-t <Target>] [-u <SCP user>] [-g <scp group][ -U
<Rmt SCP user>] [-G <Rmt SCP group>] [-M] [-I] [-L] [-R]

Options:
-b <local NRT backup path> : path for NRT backup files. Default: (/var/netwitness/backup)
-m <mode> : remote transfer mode (scp or nfs). Default: (scp)
-p <NetWitness Server backup path> : Path on NW server for logs and location of all-systems
                                     file. Default: (/var/netwitness/nw-backup)
-d <Destination backup path> : Path on destination server to move completed backup files to
                               via nfs or scp. Default: (/var/netwitness/nw-backup)
-s <remote server IP> : Destination server IP address for storing completed backup files,
                        transferred via nfs or scp. Default: (NW Server IP)
-u <SCPUser> : User acct for SCP transfers of completed backups, user must exist on all
               target systems and on Destination server. Default: (root)
-g <SCPGroup>: Group associated with User acct for SCP transfers of completed backups,
               user:group must exist on all target systems. Default (root)
-U <RmtSCPUser> : User acct on Remote system for SCP transfers of completed backups
                  user must exist on remote server and have SSH-key auth configured.
                  Default: (root)
-G <RmtSCPGroup>: Group associated with the User acct for Remote SCP transfers of completed
                  backups, group must exist on remote server. Default (root)

-l <mount_point> : local mount point for NFS share (for -m nfs). Default: (/mnt/backup)
-t <Target> : backup ONLY specific target(s), (can be anything grepable from all-systems
              file). Default: (all)
               special Targets:
```

```
                core (Broker, Concentrator, Decoder, LogDecoder, Archiver,
                      LogCollector(vlc), NetworkHybrid, LogHybrid)
                nonw (all devices except AdminServer)
                nwonly (AdminServer only)
                esaonly (all ESA devices only)
                endpoint (all endpoint devices EndpointHybrid, EndpointLogHybrid, Gateway)

    Exclusions:
    -M : Exclude the Malware Analysis File Store.
    Inclusions:
    -I : Include Broker Index files for RMA processing.
    Service Control:
    -L : Do NOT Stop the Log Collector Service during backup. Default: (Stop service)
    -R : Do NOT Stop the Reporting Engine Service during backup. Default: (Stop service)
          Note: NRT normal operation would stop these services, not stopping will affect:
                LogCollector - Will not have latest tracking data for some logs.
                Reporting Engine - Some live chart data and alert status data may be lost.
```

NOTES –

- If using SCP to a server other than the NW Admin Server, the copy uses the "-3" option and makes the transfer of the backup file via the NW Admin Server, so ONLY the NW Admin server needs to have SSH key authentication configured to the destination server.
- Make sure the modified **nw-base.nrt** file is in the same directory you are running the **nw-backup.sh** script from, the script will hash that file, and verify that hash against the file on each server, if they do not match, it will copy the file in the script directory (where you ran the nw-backup.sh script from) to the remote host, before triggering the NRT backup on that host. If you do not want to make any modifications to the default file, don't include the file in the script run directory.
- Deploy Admin password is programmatically called, so no exposure of password in the scripts
- The **/var/netwitness/nw-backup/all-systems** file can be used for a myriad of other scripting calls, or for targeting a specific type of host, especially when using "salt" commands.

RESTORE: nw-restore.sh

- Follow the published backup/restore document for the version of NW you are running
- The only change will be that the file is archived in a tar file, so after copying the file to the /var/netwitness/backup directory on a NEW host, run tar -xvf <backup-file-name>.tar to extract it there.
- Then follow the remaining instructions for running nwsetup-tui in recovery mode, install services, and upgrade host to same version as backup.
- Then either manually run nw-recovery-tool --import or use the nw-restore.sh script from the NW server to finish the restore process.

nw-restore.sh

```
    Usage: ./nw-restore.sh -t <hostname> [ -b <NRT path> ] [ -p <NW backup path> ] [ -I ]
           [ -h ]

    Configuration Options:
        Note: All command-line options except -t <hostname> are optional.
              With no other options selected, the script will use default paths
              and run against the targeted host.
```

```
General Options:
     -b <NRT backup path>: path to the location of NRT backup files on machine
                           being restored. Default: (/var/netwitness/backup)

     -p <NW backup path>: Base path on NW Server for logs and location of
                          all-systems file. Default:(/var/netwitness/nw-backup)

     -t <Hostname> : Hostname or IP of server being restored. (REQUIRED)

     -I            : Include the Broker Index files for RMA. Default: (Exclude)


Examples:

   #1: ./nw-restore.sh -t nw-decoder-01

   Would run the restore on 'nw-decoder-01' device with default path options:

    o  Backup files on host located @ /var/netwintess/backup
    o  all-systems file located on the NW server @ /var/netwitness/nw-backup/all-
       systems

   #2: ./nw-restore.sh -b /var/netwitness/localbackup -p /var/netwitness/database/nw-
        backup -t nw-broker-02 -I

   Would run a restore with the following options:
    -b : NRT backup files are in /var/netwitness/localbackup on host being restored
    -p : all-systems file in /var/netwitness/database/nw-backup on the NW Server
    -t : get target host info for 'nw-broker-02' from the all-systems file
    -I : If Index data exists in the backup files, restore the broker index files
```

CHANGE NOTES –

Device-Type Changes

Previous versions of the backup script were dependent on the Device-Type, listed in the all-systems file, being of a single type (i.e. **AdminServer, Broker, Decoder, etc.**) and was based off the single "installed services" entry in mongo for each host.

NetWitness 11.4+ introduced the ability to **add** additional services to hosts (i.e. add **New Health & Wellness(Search)** to a **Broker**, or add **Endpoint Server** to a **LogHybrid**), so now multiple "installed services" are listed for these hosts. The **get-all-systems.sh** script, has been updated to handle this scenario by concatenating the installed services together, separated by a '+'. In the above examples the all-systems would show the Device Type listed as **Broker+Search** and **Endpoint+LogHybrid**. This allows continued use of the entries in **all-systems** to be used as a reference by other scripts when looking for specific services.

Non-root User and/or group for SCP

The backup scripts now support using a non-root **user:group** for the SCP copy of the backup files from the host being backed up, to the NW Server (or to an external destination server). If your environment does not allow direct login as root via ssh, or the host you are copying the backups too requires a non-root user, then this will allow for that scenario.

The backup user is required to have full access to "read" the files written by NRT during the backup, and the ability to create directories and files on the destination server.

1. **get-all-systems.sh**

a. Run with the **-u <username>** option, (assumes group name the same as username), if the user account/group does not exist on the NW Server, you will be prompted to have them created.
   Default Settings for the account will be the following:
   Example: using **nwbackup** as the username and not specifying an alternate group or home directory path (-d)

   username: **nwbackup**
   password (you will be prompted to enter)
   group: **nwbackup** (same name as the username)
   home directory: **/var/netwitness/nw-backup**
   (or whatever the **-b <bupath>** option is set to)
   shell: **/bin/bash**

b. Adding the **-g <groupname>** option will create/use a group that is different than the username
c. Using the **-d <homepath>** will create an alternate home directory
d. If preferred, the user account can be created manually before running the script or use an existing account. If ONLY the username option is given, the **Owner:Group** of the backup directories will be set to **username:username** if the group option is also given they will be set to **username:groupname**
e. If copying the backup to a remote (non-netwitness) destination server, either the user MUST exist on that server, have the same primary group, (or use the -U -G options to designate a different user:group on the destination server), have write permissions to the destination path backup files will be stored, and have ssh key authentication pre-configured.

2. **ssh-propagate.sh**
   a. Run with the **-u <username>** option, create and propagate the ecdsa-521 ssh key for this user to all other nodes in the all-systems file.
   b. If user does not exist on a node, adding the **-c** option will allow the script to create the remote user and set the password the same as the local user on the NW server, as well as copying the ssh key to that host.
   c. If an alternate group name was given during the run of get-all-systems.sh, then you should run this with the -g <groupname> option also, so users are created with the same user:group on all systems

3. **nw-backup.sh**
   a. Run with the **-u <username>** uses designated username for all copying of backup files from nw nodes being backed up, to the designated destination (NW Server or external host).
   b. If the **-g <groupname>** option was used during the setup scripts, it should be included along with the username option.
   c. User MUST exist on the nw nodes and have read permissions to the backup files and write/create permission on the destination host.
   d. If the destination host requires a different user:group for SCP then use the -U & -G options to designate the remote user:group for transfers.

This does not change the fact that the actual backup scripts MUST be run as the root user, it just allows the use of a NON-ROOT user to do the file copies between systems.

/etc/hosts updates (ssh-propagate.sh)

1. Pre 11.5 - checks for IP entry in **/etc/hosts** for each host in the all-systems file, and if missing, adds an entry for **<ipaddress> <hostname>.**
2. Post 11.5 - /etc/hosts is controlled by a chef process to handle entries for known nw hosts, the new entries will look like this:
   <ipaddress> <UUID> <UUID.netwitness>
   a. Any entries manually added to the /etc/hosts file will be wiped out by the chef process. To add <hostname> (short hostname) to these entries, a new file **/etc/hosts.user** (format: **<ipaddress> <hostname>**) must be created so the chef recipe can update the **/etc/hosts** file with the hostnames.
   b. **ssh-propagate.sh** will automatically create a **/etc/hosts.user** file from the entries in the all-systems file, and run the chef recipe to update the /etc/hosts, so the entries will read:
   <IPaddress> <UUID> <UUID.netwitness> <hostname>
   c. In addition, if you need to add other hosts to the /etc/hosts that are not part of the Netwitness Deployment (and not resolvable by DNS), you can create a **hosts.custom** file in the directory where the **ssh-propagate.sh** is run from, and it will append those entries to the **/etc/hosts.user** file before running the chef recipe.

nw-backup.sh

1. Previously, **nw-backup.sh** relied on the Device Type entry in all-systems to match an NRT Category, NRT uses the **/etc/netwitness/recovery-tool/category.sequence** file to get a list of components to backup for that Device Type. Post 11.4, with the ability added to install additional services on hosts, this process would no longer work effectively.
   a. The script no longer depends on the DeviceType entry in the all-systems to run the backups, (it will still use it for filtering on specific types if the -t option is used, but is not used to determine what services are running on a particular host). It now queries each host for a list of enabled processes, then parses that list to determine what components are running on the host, and dynamically creates a component list to pass to the NRT script when launched.
   b. Any changes to a host (adding services) will not require a change to the backup process, as the new services will be discovered automatically when the backup is run.
2. NFS use (for copying backup files to another host) has been updated to give better feedback if there are issues.
3. Addition of the -I option to backup Broker Index files for RMA/Tech Refresh, so they can be restored using the **nw-restore.sh** script without losing their aggregation last session information. (Note: this DOES STOP the broker process during the backup)
4. Added extraction of custom feed files (csv/xml) to facilitate restore of custom feed tasks.

cf-fix.sh (and feedfix directory with backup.sh and restore.sh files)

1. NRT excludes excludes the /var/lib/netwitness/uax/scheduler directory in the sa-server NRT backup script, due to that directory holding all the job data from pcap/log extracts as well as the custom feed files (xml/csv). This causes an issue with transfer of the custom feed files to the warm standby server during sync, the jobs still exist, but the supporting files are missing.
2. To fix this issue, the nw-backup.sh script automatically adds the 2 scripts from the feedfix directory (backup.sh and restore.sh) to the /var/netwitness/uax/nwtools directory and modifies the sa-server.nrt file to run them pre-backup and post-restore.

3. The backup.sh script extracts the custom feed files (*.xml and *.csv) to a tar file and places it in the /var/netwitness/uax/backup directory so it is included in the backup of the sa-server files.
4. The restore.sh script extracts those files back to the proper location to support the custom feed jobs.
5. If you have a Warm Standby Admin server, copy the cf-fix.sh and the feedfix directory to the Standby server and run the cf-fix.sh script to configure the sa-server.nrt file with the new commands and add the 2 script files to the proper location. If you have run the nw-backup.sh script on the Primary Admin server these modifications are already made, so when the warm-standby sync job runs (basically an NRT run) and the backup file is copied to the standby server, during fail-over to the standby server, the files will be restored properly.