Design and Implementation of a Guard Installation and Administration Framework (GIAF)

15 March 2007
Doc Rev 1.0

Boyd Fletcher
USJFCOM J9 & SPAWAR SC SD
boyd.fletcher@je.jfcom.mil
boyd@spawar.navy.mil
757.535.8190

Chris Roberts
General Dynamics
christopher.roberts@je.jfcom.mil
757.203.3294

Motivation

- Traditional guards are complex to install
  - Hundreds of steps
  - Lots of typing at command line
  - Complex ST&E* at installation sites
- Traditional guards are complex to maintain
  - Command line tools
  - Inconsistent User Interfaces
  - Numerous unrelated tools
  - User environment designed for engineers not operators
- Guards frequently fair poorly in CT&E** because of complex or poorly written installation processes

*Security Test & Evaluation
**Certification, Test & Evaluation

Project Goals

- Fast install
  - 30 mins from bare hardware to software installed and basic configuration complete
- No command line
- No typing of commands during install or operation
- Wizard based installation (next/previous)
- “Windows-like” user interface
- Integrated administration tools
- “Install Secure”
- Consistent, repeatable Installations
- Install from DVD
- Increase the success rate for guard CT&E’s and reduce the time and cost of conducting them.
- Reduce time to conduct an ST&E

GIAF Features

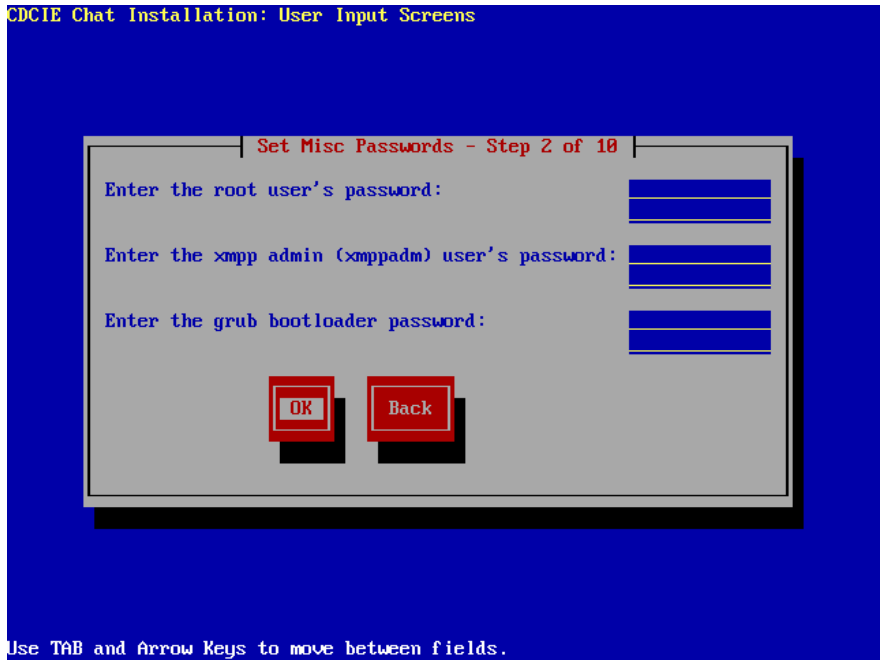
- Perl/Newt based installer framework
- Bourne Shell based DISA STIG compliant lockdown scripts
- Extensible Guard Administration Console (GMC)
- System monitoring
- System integrity checking
- System restoration
- Locked down X Windows with “Windows-like” look-n-feel
- Numerous small tools for assisting in the software installation and configuration

Installation Process

- Automates installation of o/s, o/s patches, and application packages
- Automates configuration of applications
- Automates security lockdown including activation of the SELinux policy
- Provides a common log file for all installation messages
- Runs the guard installer to gather system and application configuration information that is used by the configuration scripts
- Most work done in ks.cfg and several shell and Perl scripts

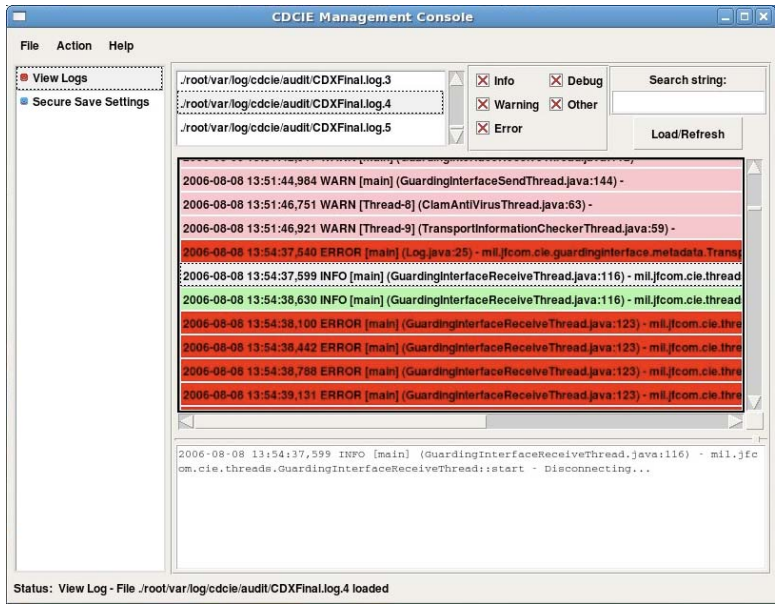
Installer

- Perl/Newt based
- Runs immediately after operating system packages are installed
- Uses common Next/Previous Wizard metaphor
- Does strong password validation against o/s policy for user accounts
- Easily extensible
- Verifies Data Format (I.e. IPs)
- Has ability to write configuration information to a key/value file that is used by the configuration scripts



Guard Management Console

- Perl/Tk based
- Supports dynamically loaded modules based on role (user)
  - Module/Role associations are configured in an XML file
- Uses SELinux policy to enforce which users can access which modules
- GMC is constrained via SELinux Policy
- Interface works similar to MS Management Console (MMC)
- Included Modules:
  - Log Viewer
  - Backup
  - Process Control
  - Password Changing
  - Network Configuration



Skeleton of a Module

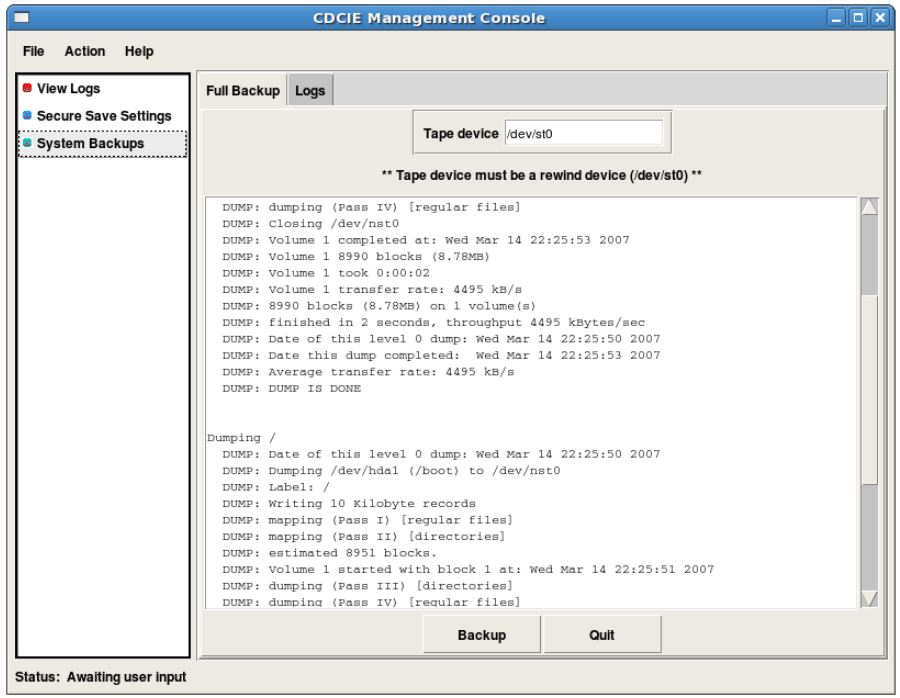
```
package "ModuleName";
sub new {

   # This registers the module with the GMC.  When called the
module will become available on the left side module selection
screen.
}
sub load {
   # This loads the module into the GMC and displays it in the
right side of the GMC window.
}
sub unload {
   # This will unload the module from the GMC.
}
sub help {
   # This will return a help dialog for display to the user
}
-1
```

System Monitoring (local)

- Provides a quick view into the current status of critical system including:
  - Critical Processes
  - AntiVirus
  - TCP/IP ports
  - Memory usage
  - CPU usage
  - Disk space usage
  - User Password Expiration
  - Displays SELinux booleans
- Easy to modify to monitor multiple TCP/IP ports, processes, and booleans.
- Displays system classification, logged in user, and system version number.

System Backup and Recovery

- Backup
  - Supports backup to tape.
    - Backup to disk in near future
  - Only does full system backups
  - Incrementals/Differentials not allowed
- Recovery
  - Accessed via a boot option on the distribution media
  - Only Restores entire system
  - SELinux file system relabeled during this process



Centralized Logging Daemon (CLD)

- Leverages the Apache Log4J Project
  - Log4C may be supported in the future.
- Clients (i.e. filters in an assured pipeline) use a custom appender for Log4J
- Clients talk to CLD over a one way System V Message Queue that is constrained by SELinux
- The CLD support uses Log4J and supports all its normal logging capabilities.

An example of logging a simple text message:

```
Logger log = (Logger)
  Logger.getLogger(App.class);

log.info(“Application initialized
successfully.”);
```

An example of logging a document object and an associated message:

```
Logger log = (Logger)
  Logger.getLogger(App.class);

TransportInformation ti =
  new TransportInformation();

ti.setStatusMessage("This
TransportInformation object includes a secure
document as DATA.");

ti.setData(CDGLogHelper.readFileIntoByteArray
(“confidential.doc”));

log.warn(ti);
```

Lessons Learned

- GMC has reduced the time to create administration tools from weeks to days
- Early development of the installers allowed us to do full system testing earlier in the development cycle and conduct more accurate unit level testing since the applications are being deployed into a realistic (very close to end state) environment.
- Portions of the GIAF have been used successfully in our NSA CT&E'd cross domain solution CDCIE Chat.

Future

- Support Remote Monitoring of Guard
- Support Remote Logging via the CLD
- Integration with CLIP
- Replace Secure JTuX with Tresys' Secure IPC library