Next Generation Security for the 10,240 Processor Columbia System

Thomas Hinke, Paul Kolano, Derek Shaw, Chris Keller, Dave Tweten, Todd Welch, Wen (Betty) Liu

June 2005
Outline

- Columbia Overview
- System Requirements
- Related Work
- Columbia Security Design Decisions
- Columbia Security Architecture
- Secure Front End
- Secure Unattended File Transfer
- Secure Unattended Proxy
- Perimeter Enforcer/Controller
- Security Monitoring and Intrusion Detection
- Conclusions
Columbia Overview

• A 10,240-processor system
  – Located at the NASA Advanced Supercomputing (NAS) division at the NASA Ames Research Center
  – Supports each of NASA’s four missions: Science, Exploration Systems, Aeronautics and Space Operations
• Comprised of 20 Silicon Graphics nodes, each consisting of 512 Itanium II processors
  – Most users access the nodes through the PBS batch scheduling system
  – Some users have direct access
• A 64 processor Columbia front-end system supports users as they prepare their jobs and then submit them to the PBS system
• Columbia nodes and front-end system use the Linux OS
• Prior to SC04, the Columbia system was used to attain a processing speed of 51.87 TeraFlops,
  – Number two on the Top 500 list of the world’s supercomputers
  – World’s fastest “operational” supercomputer since it was fully engaged in supporting NASA users
System Requirements

- Columbia’s operational requirements that impact security
  - The system must be Internet accessible from NASA Centers, companies and universities
  - The system will be connected to the 10 Gigabit/second National Lambda Rail
  - The system should support script-based file transfer, where the user is not necessarily present at the time of the transfer
  - The system should be able to support grid computing and other network-oriented software that require the opening a number of ports to external sites

- The security requirements included the following:
  - Allow external access and interaction with Columbia, while preventing and detecting malicious behavior to the extent possible
  - Support two-factor authentication using RSA’s SecurID
  - Support security monitoring and intrusion detection
Related Work: Access Control Architectures

- Results of site visits to six of the nation's leading computer centers and an interview of personnel from a seventh
- Following provides taxonomy of the architectures that are being used

Fully Open Type

- High End Computer (HEC) ➔ HEC-based ID & Authentication ➔ Internet

Static ACL Type

- High End Computer (HEC) ➔ HEC-based ID & Authentication ➔ Router with Static ACLs ➔ Internet

Dynamic ACL Type

- High End Computer (HEC) ➔ HEC-based ID & Authentication ➔ Router with Dynamically set ACLs ➔ ID/Auth Website ➔ Internet

VPN Type

- High End Computer (HEC) ➔ HEC-based ID & Authentication ➔ VPN Server ➔ Router with Static ACLs ➔ Internet

SGIUG June 2005
One Extreme of Access Control Architecture

- Fully Open Type
  - Advantages:
    - Simple since it involves no additional security components
    - Running of grid software or other network software not impacted
  - Disadvantages:
    - High-end computer is open to direct attack from the Internet
Another Extreme of Access Control Architectures

- VPN Type
  - Advantages:
    - High security
  - Disadvantages:
    - User must install necessary VPN clients

VPN Type

- High End Computer (HEC)
- HEC-based ID & Authentication
- VPN Server
- Router with Static ACLs
- Internet
Access Control Architectures Middle Ground: Static

- Static ACL Type
  - Advantages:
    - Provides an additional layer of security over fully open type
    - Does not require any user client as in VPN type
    - Open to attack by only those systems that are permitted to send packets through perimeter
    - A variant that pushes toward dynamic ACL type,
      - Can use ACLs to block IP addresses that are caught scanning
      - Release blocks based on cessation of activity after some time period
  - Disadvantages:
    - Must leave ports open to support grids and other network software
Access Control Architectures Middle Ground: Dynamic

- **Dynamic ACL Type**
  - **Advantages:**
    - Default is deny all access, so does not leave unnecessary ports open
    - Upon successful user authentication, allow access from user’s IP address
    - For Columbia, taking this to the port level
  - **Disadvantages:**
    - More complex
    - Must know when to open and to/from which host
    - Must know when to close access
Related Work: Two-Factor Authentication

- Technology being used included Cryptocard and RSA’s SecurID
- At the extremes
  - Some sites did not use any form of two-factor authentication
  - Some sites required all users to use two-factor authentication
- In the middle
  - Some sites required two-factor authentication only for privileged users
- A number of sites where considering the adoption of two-factor authentication
Columbia Security Based on Enclave

- Columbia, its associated mass storage, visualization and other high-end systems placed within Columbia enclave
  - High level of security control is enforced on all externally initiated access to any system within the Columbia enclave
  - Access between systems within the enclave is allowed to proceed with little security-induced restrictions
- All user workstations are considered to be outside of the Columbia enclave
  - Including those that are in the local organization
  - Including those that are located at some remote location
  - Hence, access by all user to systems within the Columbia enclave is subjected to a high level of security mediation
Columbia Security Design Decisions

- Security system must satisfy the requirements of a security reference monitor, which requires that
  - It must be invoked on every access by a subject to an object
  - It must be tamper proof
  - It must correctly enforce the desired security policy
- Security system must support the unattended transfer of files into and out of Columbia, where a user is not present at the time of the transfer to perform two-factor authentication
- Security system must be able to handle network traffic up to a rate of 10 Gigabits/second
- Security system must be able to support complex protocols that may need to use many different ports for remote system communication
  - bbFTP for large file transfers
  - Grid software, such as that provided by the Globus project
- Security system must be as unobtrusive as possible and place a minimum burden on the user
- Security system must not require Columbia users to install any special software

SGIUG June 2005
Columbia Security Architecture Areas

• The Columbia’s security protection system can be partitioned into three basic areas
  – The perimeter protection system, which controls access through the perimeter of the enclave
    • Main focus of presentation
  – The monitoring and intrusion detection system, which monitors for unauthorized activity that is initiated from either inside or outside of the enclave
    • Minor focus of presentation
  – The security systems associated with internal operation of the systems within the enclave
    • Will not be considered further in this presentation
Secure Front End Supports Strong Authentication and Shields Columbia

- Columbia’s SSH daemon will not be exposed directly to an attack from the internet
- SFE’s SSH daemon will be exposed to direct internet attack
- SFE will be locked down to minimize ability to attack it
- Supports only SSH, which includes SCP
- Supports strong authentication for Columbia enclave
  - RSA’s SecurID authentication
    - PIN + time-varying, pseudo-random number called a token code
    - Also requires either password or public key authentication
  - Since SecurID does not satisfy NASA’s password regulation
    - Requires three out of the four possible types of characters and SecurID does not distinguish between upper and lower case letters
SFE Eases User Access

- Supports an SSH pass-through capability to make the SFE as unobtrusive as possible
  - User places public keys on SSH and on Columbia (future plans call for key generation and deployment infrastructure)
  - User uses an ssh-agent on his workstation that securely holds his private key,
  - User can access Columbia (through the Secure Front End) by having to personally enter only his SecurID PIN and token code
    - ssh-agent performs public key authentication first with the Secure Front End and then Columbia (or other system within the enclave)
SFE Supports Reference Monitor

- Satisfies "always invoked" requirement since it is placed in a position to mediate all interactive access
  - Network ACLs ensure that SFE is only device that can support interactive access within Columbia enclave
- Satisfies tamper proof requirement in a number of ways
  - Implemented as a separate device from the other systems within the Columbia enclave that support user processing
  - Design of the SFE minimizes the opportunity for users or code operating on behalf of the user to attack the systems
    - Uses Linux distribution that allows one to start with a minimal capability system and then add only those additional capabilities that are needed
    - Minimizes the possibility of including unneeded capabilities with potential security vulnerabilities
  - Uses chrooted (or jailed) environment for all users which limits
    - Their access to system directories
    - Their access to only the minimum Linux functions necessary to perform a limited number of tasks: log in, pass-through, scp file transfer or public key deployment

SGIUG June 2005
Secure Unattended File Transfer

• Supports user’s need to transfer files
  – When user is not present to perform SecurID authentication
    • E.g., where file transfer is under script control
  – While still satisfying NASA requirements to have processing on Columbia tied to two-factor authentication

• Initial approach: DMZ-resident file server
  – Accessible to users that need to transfer files into Columbia enclave
  – Does not require two-factor authentication
  – File transfer is two-step process
    • Users can transfer files into DMZ file server from outside of Columbia Enclave
    • Columbia users then pull files into enclave
      – which is allowed by Columbia security policy
    • Analogous approach can be used in the opposite direction

• Final approach: Secure Unattended Proxy
  – Allows users to acquire ticket (based on public key technology) to transfer files for a limited period of time
Secure Unattended Proxy (SUP)

- Allows a user to pre-authorize future file transfers
  - SecurID and Password authentication
  - Through a web site provided by the Secure Unattended Proxy component
- Authorization results in the creation of a public/private key pair
  - Used to perform authentication at the time of the transfer
  - Public key automatically stored in the Secure Unattended Proxy and on the target system
  - Private key used by script at time of transfer to perform public key authentication
    - With Secure Unattended Proxy component
    - With Columbia (or other target system)
- Capabilities of the private/public key pair limited
  - Valid for only one week
  - Can be used for only the transfer of files to a specified set of pre-approved directories on the target system
SUP Role in Security Reference Monitor

- Secure Unattended Proxy serves as major component of reference monitor for secure unattended file transfer
  - It mediates all commands given by users to ensure that they
    - Use only authorized protocols (e.g., SCP or bbFTP)
    - Use only arguments consistent with security concerns
  - It performs any rewriting of the command
    - To bring it into compliance with Columbia security policy
    - Prior to passing the command on to the SSH daemon on the target systems
  - To support "tamper proof" requirement
    - Secure Unattended Proxy is located in component that is separate from user processing
    - Access is even more restricted than for SFE since users are not even permitted to log into Secure Unattended Proxy component

SGIUG June 2005
Columbia Role in Security Reference Monitor

- SSH daemon on Columbia (or other enclave system) also performs part of the reference monitor function
  - Must be protected from tampering
  - Ensures that the command invoked from the Secure Unattended Proxy will actually be invoked on the Columbia
- If the Secure Unattended Proxy says
  - "ssh Columbia-front-end /usr/bin/bbftpd -f -s",
  - Then code associated with the daemon will ensure that Columbia actually invokes "/usr/bin/bbftpd -f -s" and not something else due to user configurable ssh behavior
Secure Unattended Proxy in Use

- All unattended file transfers are initiated through an SSH connection
  - Between the user’s computer and Columbia (or other system within the Columbia enclave),
  - Secure Unattended Proxy serves as the reference monitor intermediary
- SSH connection supports the control channel for the transfer
  - For SCP transfer it also supports the data channel
- For high performance file transfer protocols such as bbFTP
  - A separate data channel is created to directly connect Columbia with the remote computer
  - Connection is still under the mediation control of the Perimeter Enforcer
Perimeter Enforcer/Controller Design Constraints

- Objective is to have a security reference monitor to mediate all network packet traffic into the Columbia enclave
- Design driven by need to support grids and other network-oriented software
  - Operating behind perimeter protection system
  - Operating without having to keep a large number of ports open in anticipation of future grid activity
- Design is a "least privilege" approach
  - Ports are opened through perimeter only when needed
  - Ports are kept closed at all other times
- This required the solution of three problems:
  - How to dynamically control access at the 10 Gigabit/sec speed
  - How to determine when the ports are to be opened
  - How to determine when the ports are be closed
Perimeter Enforcer/Controller Design

- Support 10 Gigabit/sec networks
  - Must use network device such as router switch
    - Firewalls or general purpose computers cannot handle data rate
    - Need to use dynamic ACLs since ACLs are the access control technology of switches/routers
  - Need device to support dynamic ACLs that operates atomically so that during the ACL change period
    - Packets to not drop on the floor or
    - Pass through in violation of previously established ACLs
Perimeter Enforcer/Controller Implementation

- Perimeter Enforcer, based on network router/switch, is what provides the actual enforcement
- Perimeter Controller needed since network devices such as routers/switches have limited intelligence
  - Based on general purpose computer
  - Used to control the opening and closing of ports
  - Performs this control by sending ACL changes to the Perimeter Enforcer
- How to determine when to open and close ports
  - Solution is to instrument authorized applications
  - Application itself can inform Perimeter Controller when
    - Desired ports need to be opened
    - Ports can be closed since they are no longer needed
  - Prototypes show that this is relatively easy to do
Security Monitoring and Intrusion Detection

- Objective of the NAS Security Monitoring System is to protect our network from threats such as
  - policy violations,
  - vulnerabilities and
  - intrusions
- This design can be separated into the following four functional areas:
  - Active Monitoring
  - Passive Monitoring
  - Analysis
  - Reporting
Multifaceted Monitoring Capabilities

- Active Monitoring
  - Periodic scans for possible vulnerabilities and remote exploits on port services across our network
  - Verify scan results with probe packets
  - Use packets to obtain current profiles (OS, ports availability, services, packages, versions etc) of all systems

- Passive Monitoring
  - Analyze network packet traffic to:
    - Match known attack signatures
    - Discover deviations from normal IP protocols and normal traffic patterns
    - Detect NAS policy violations
  - Use honeypots to learn and guard against new intrusion techniques not yet documented by security community
  - Centralized logging to detect suspicious user/account activities
Multisource Analysis and Reporting

• Analysis
  – Correlate data from resources across monitoring network to gather a comprehensive security status
  – Eliminate false-positives during packet analysis and scan verifications by referencing system profiles
  – Examine logs for user/account tracking and anomalies
  – Connection tracking and traffic trending

• Reporting
  – All monitoring resources will send alerts and/or reports to a central correlation engine
  – Higher-level analysis and forensics can be manually conducted based on reports from autonomous analysis
  – Security analyst views only processed and prioritized alerts
Conclusions

- Reference monitor concept has been shown to be a valuable concept in designing security systems for high-end computer systems.
- Tamper proof requirements led to the use of a separate component, the Secure Front End, to mediate all interactive access to Columbia.
  - Decision had important ramifications for integrating two-factor authentication into the design.
    - Concentrated SecurID authentication for interactive access into the two Secure Front Ends.
    - Greatly simplified the design, since SecurID did not have to be engineered into Columbia and the other systems located within the Columbia.
- Always invoked concept led to
  - All interactive sessions flowing into the Columbia enclave through the Secure Front Ends.
  - All unattended file transfers flowing through the Secure Unattended Proxy.
  - This makes it a relatively easy task to identify connections that enter the enclave without going through these components, indicating a problem with network security configuration.