



# Data Center Security Whitepaper

Understanding the Security  
Implications  
of Deploying KVM Over IP

Raritan Computer Europe B.V.  
May 2004

### **Copyright and Trademark Information**

This document contains proprietary information that is protected by copyright. All rights reserved. No part of this document may be photocopied, reproduced, or translated into another language without express prior written consent of Raritan Computer, Inc.

© Copyright 2004 Raritan Computer, Inc., CommandCenter™, Dominion™ and the Raritan company logo are trademarks or registered trademarks of Raritan Computer, Inc. All rights reserved. Java® is a registered trademark of Sun Microsystems, Inc. Internet Explorer® is a registered trademark of Microsoft Corporation. Netscape® and Netscape Navigator® are registered trademarks of Netscape Communication Corporation. RC4® is a registered trademark of RSA Corporation. Other trademarks or registered trademarks are the property of their respective holders.

Revision 1.2



## **Executive Summary**

This paper focuses on the security features of digital KVM (keyboard/video/mouse) solutions.

As enterprise data centers consider deploying a KVM-over-IP approach to server management — in which a network-accessible KVM infrastructure replaces traditional cabled KVM functionality — an entirely new set of security issues and concerns must be addressed by solution providers.

The paper provides an overview of system security in the context of both rackmount and KVM-over-IP systems, and includes a description of the threats that face enterprise systems, as well as the security mechanisms that counter those threats.

The second part of the paper goes on to describe how Raritan's Dominion Series products use leading-edge approaches in engineering, architecture and human factors to provide levels of capability and security that are unmatched in the KVM industry.

# Table of Contents

1	Introduction .....	1
2	KVM Security.....	3
2.1	Security Fundamentals .....	3
2.1.1	The Four "A"s.....	4
2.2	The Human Factor.....	5
2.3	Security Policy .....	5
2.4	Threats and Defenses in KVM-Over-IP Systems .....	6
3	Dominion Product Security.....	7
3.1	Solution Overview .....	7
3.2	Individual Dominion Devices .....	7
3.2.1	Availability .....	8
3.2.2	Encryption .....	8
3.2.3	Authentication .....	9
3.2.4	Network Security .....	10
3.2.5	Configuration and Management .....	10
3.3	CommandCenter .....	11
3.3.1	Authentication Integration and Failover.....	11
3.3.2	Certificate Management.....	11
3.3.3	The Command Center User Interface .....	11
3.3.4	Rule-Based Control .....	14
3.3.5	Auditing/Logging.....	15
3.3.6	Reporting .....	16
4	Conclusions: Facing the Future .....	18

# 1 Introduction

KVM switching solutions enable a single KVM (Keyboard, Video, Mouse) console to connect to the KVM ports or serial ports of multiple computers. KVM solutions provide control over typical Intel-based servers (or any server managed with a keyboard, mouse, and graphical monitor) by connecting to their KVM ports, allowing *BIOS-level* or *console-level* access even when the managed server's operating system (OS) is not working. KVM solutions also provide control over headless servers and many "blade" server arrays by way of their serial (RS-232) ports.

Historically, KVM solutions have predominantly been "rack-based," with one KVM switch and a single KVM console deployed in each server rack. But in the past few years, data center managers have begun to embrace a different approach, in which a single, consolidated KVM switch (or networked array of switches) is used to manage the entire data center.

This approach addresses several problems inherent in rack-based solutions:

- **Physical Issues:** Independent KVM switches require many more KVM consoles than are necessary (which is costly in terms of space and heat dissipation in a data center) and system administrators (sysadmins) have to leave their desks to use the desired KVM console.
- **Operational Issues:** Because troubleshooting a single problem often requires switching between many different servers (e.g., application server, web server and database server), sysadmins must move back and forth between racks to accomplish even simple tasks.
- **Logical Issues:** Because sysadmins identify servers by function, rather than by physical rack location, having to physically locate servers is an ongoing inconvenience; sysadmins don't care where a given server is physically located, as long as they can logically connect to it seamlessly and efficiently.

The solution to these problems lies in adopting a single, multi-user KVM solution to control all servers in the racks; in contrast to multiple KVM switches that can only control a few servers each.

The second part of this paper specifically addresses the Raritan Dominion Series, which adopts a KVM-over-IP approach at the heart of its architecture (also known as *network-based KVM* or *digital KVM*). A KVM-over-IP approach involves digitizing all KVM console information of each managed server, and sending it over network such that sysadmins can interact with the managed server from a networked workstation, just as if they were physically located at a KVM console or switch.

While the advantages of a KVM-over-IP approach are clear, it does represent a significant departure from previous approaches, and results in an entirely new set of security issues and concerns.

For example, security in the rack-based KVM approach typically relies on the physical security of the data center. The KVM-over-IP approach not only increases the size of the security perimeter to encompass the entire enterprise (and perhaps beyond), it also requires that KVM-over-IP adopts the same network security measures used by the network itself.

Digital KVM solutions must address these security concerns — and many others — with a well-defined and comprehensive architectural approach, with security-relevant innovation in both hardware and software system components.

## 2 KVM Security

KVM switches and consoles can make very attractive targets for hackers because they allow direct and immediate console-level access to corporate IT assets. And with the advent of KVM-over-IP, KVM systems become just another target on the network.

Common attacks on the network rely on various approaches (e.g., buffer overflow bugs, worms, viruses, and Trojan Horse programs) to defeat protection mechanisms and eventually take over a system. Hackers would surely agree that gaining control over the KVM system—and all the systems they in turn control—is a convenient way to achieve that malicious goal.

But that is not to say that KVM systems necessarily deserve a different, or somehow “better,” level of protection than other IT assets. As a general security philosophy, protecting any one portion of a system more heavily than another simply adds costs, while adding little true improvement in overall security. It is a bit like putting a steel door on a grass hut—while no one will get through that door, the hut itself is just as vulnerable it was before.

A solid KVM over IP solution should provide security by way of proven network security mechanisms of encryption and certificates. It is recommended to have a thorough and comprehensive architectural approach to hardening or avoiding system vulnerabilities; and leading-edge software capabilities that minimize human error. This architectural approach helps ensure that the valuable functionality provided by the KVM switch does not itself compromise IT infrastructure security in order to provide that functionality.

### 2.1 Security Fundamentals

This section describes the basics of security, including the importance of availability, human factors and security policies in achieving KVM system security.

The term “security” encompasses three broad classes of requirements:

- **Secrecy**, which is the ability to protect information from unauthorized access;
- **Integrity**, which is the ability to protect information from unauthorized alteration or destruction; and
- **Availability**, which is the assurance that authorized access to information can be accomplished in a timely manner (when authorized access is blocked, either due to a malfunction or intentional disruption, the result is called *denial of service*)

In the context of KVM security, the following concepts are also important:

- **Objects**, which are the targets of access requests. Objects include information assets like files and databases, but in a KVM system they also encompass resources such as ports, user profiles and devices.

- **Security Principals, or subjects;** in a KVM system, these are the sysadmins, data center managers and security managers, who make access requests and also manage access rights for all users.
- **Audit information,** which is the record of all security-related events. The audit trail is critically important in assuring individual accountability and the detection of user irresponsibility.
- **Security attributes,** which are the meta-information associated with objects and security principals. Security attributes enable fine-grain control in determining whether an access request is permitted, and also provide the mechanism with which principals and objects can be treated as *groups* in setting security policy.
- **Encryption,** which is simply a mechanism that lets us avoid having to physically protect (or simply trust in the security of) the path between principals and the objects they access. Encryption can be used to prevent disclosure or modification of information, and to certify authenticity of information.
- **Security Perimeter,** which is a boundary between areas of the distributed system in which different levels of security are enforced. Every enterprise has a security perimeter in the form of a “firewall” between corporate systems and the outside world, and many have multiple security perimeters inside the firewall. “Well-known” and registered ports are, in essence, holes in the security perimeter, and may provide opportune avenues of attack.

### 2.1.1 The Four “A”s

The protection of information assets relies on four security functions:

- **Authentication,** which is the verification of the identity of principals. Authentication is based on what you know (e.g., a password), what you have (e.g., a smartcard), or what you are (e.g., biometric information such as fingerprint or retina pattern).
- **Authorization,** which is the representation of the security policy that describes the access permitted by principals to objects. Authorization may depend on other factors such as physical location of the principal, time of day, or the specific software used to access the object.
- **Access Control,** which is the examination/comparison of security attributes that determines whether or not a given access request is permitted.
- **Auditing,** which is the recording of significant security-related events.

## 2.2 The Human Factor

Human factors are critically important to security, for a number of reasons.

First, although underreporting makes it difficult to obtain exact figures about network penetration incidents, it appears that internal attacks are at least as prevalent as external attacks and are probably even more common. So a KVM solution must not only prevent penetration from outside the firewall, but also assure the accountability of internal employees (both sysadmins and general users), as well.

Second, if the user interface to a KVM system is too confusing, clumsy, or tedious, sysadmins will find it difficult to set up and maintain the appropriate degree of system security (and as a result may provide users with far more access than is actually needed). If security features are seen as making day-to-day operations too difficult, they might not even be used—even if failure to use them might result in a security catastrophe. The clearer the UI's representation of security settings, the more likely that appropriate controls will be recognized, understood, and maintained.

Finally, the ability of a system to generate understandable reports from the mass of security audit data is crucial. Good reporting lets sysadmins and datacenter/security managers understand and solve the actual security problems, instead of being overwhelmed by the huge volume of symptoms that may be recorded in the audit files.

## 2.3 Security Policy

Security policy is the sum total of access control over objects by security principals (sysadmins and general users). Thus, every corporate network has a security policy, whether they know it or not. And one of the most common reasons for a security compromise is bad security policy.

If the user interface to security management functions is poorly designed, it will be difficult for an enterprise to implement a good security policy. A good user interface is clearly required to determine whether the security policy is good or bad.

Security experts argue that an enterprise's security policy should be defined explicitly and independently of any security controls that are in place. The defined policy should then be implemented by way of the security mechanisms available. A rules-based user interface that lets you specify security policy helps immensely in this approach as well.

## 2.4 Threats and Defenses in KVM-Over-IP Systems

There are three broad classes of attacks on distributed systems:

- **Probing attacks**, which involve exploitation of security controls that have insufficient power, or are incorrectly applied. A user who probes a system is simply finding “normal” ways to get at information or objects to which he or she should not have access. Obviously, probing is controlled by security policies that have adequate flexibility and are properly applied. However, probing attacks are typically discovered by examination of audit information; the chances of discovering a probing attack are improved by good reporting capabilities.
- **Penetration attacks**, which are the successful compromise of a protected system. Penetration depends on a weakness or flaw in the implementation of security controls. For example, an eavesdropping attack can capture account/password information; a replay attack attempts to re-use an access request that should no longer be considered valid; and a virus, worm, or Trojan Horse attack typically exploits software flaws such as buffer overruns.
- **Denial of service (DOS) attacks**, which involve any reduction in the availability of systems for their intended purpose. DOS attacks are the most difficult to detect and prevent because they involve normal activities, for example flooding a server with unwanted e-mail or service requests. One of the most significant vulnerabilities to DOS attacks is the presence of well-known or registered ports.

As mentioned above, encryption—in the form of secure protocols and certificates—is the mechanism used to protect the integrity and secrecy of network traffic, and in the case of certificates, to provide authorization protection, the assurance of message authenticity and prevent of replay attacks by including timestamp information.

While DOS attacks are difficult to detect and repel, other forms of attack on availability can be countered. For example, an enterprise may be concerned about “monoculture” of its systems, meaning that systems that share a single OS also share its vulnerabilities. No operating system is immune to attack, but some are more vulnerable than others; the greater the popularity and vulnerability of an operating system, the more likely it is that attacks will be tailored to it.

Because KVM systems serve as the enterprise’s last resort in the face of attack, they must be *more* reliable than the servers they manage. If the KVM-over-IP system (or critical parts of it, such as the authorization and authentication databases) use the same OS as the enterprise’s core business systems, then all systems may be at risk from a single threat. This is particularly true for threats such as viruses, which are using increasingly clever methods to deliver their payloads to uninfected machines. And because we live in an age when virus-creation “wizards” are freely available on the Internet, we can expect this threat to increase in the short term.

## 3 Dominion Product Security

### 3.1 Solution Overview

A complete description of the capabilities of the Raritan Dominion Series product family is beyond the scope of this paper (please refer to the Raritan website at [www.raritan.com/kx](http://www.raritan.com/kx) for complete product details). However, while this paper addresses the specific security characteristics of Raritan KVM solutions, some “regular” product attributes — such as ease-of-use and availability — have a serious impact on security and are also covered here.

The Dominion Series includes:

- Dominion KX – a digital KVM switch featuring integrated KVM-over-IP for remote server access, while also providing traditional rack-based, analog KVM capabilities
- Dominion SX – a secure console server for serial (RS-232) access to headless servers, networking gear, many “blade” server arrays and other serially-controlled infrastructure devices
- Dominion KSX – a KVM-over-IP solution that combines the capabilities of the Dominion KX and SX, specifically for remote or branch office infrastructure management
- CommandCenter – an enterprise-level aggregator and management platform for all Raritan KVM-over-IP products and users, providing a centralized entry point to all devices managed with Raritan’s networked KVM products

Raritan provides this full set of solutions in order to flexibly address the needs of different customers. But more importantly, from a security standpoint, the scalability provided by the Dominion Series means that a customer’s security solution and existing investment in Raritan products can be extended almost without limit. Large customers can use CommandCenter to aggregate and manage any of Raritan’s KVM products. And they can combine multiple CommandCenters for even greater scalability.

Small-to-medium business can start small and grow their Raritan KVM solution as their business grows. And because the Dominion product family supports both rack-based and IP-based approaches, they can experience a seamless switch-over to KVM-over-IP if and when that change makes sense.

### 3.2 Individual Dominion Devices

The following sections describe security characteristics and hardening mechanisms that are inherent to each Dominion KX, Dominion SX, and Dominion KSX devices themselves – even when deployed “standalone”, without integration with CommandCenter.

However, note that CommandCenter in turn further exploits many of the security features in each of the Dominion KX, Dominion SX, and Dominion KSX products, to provide an even higher level of overall security, availability, and ease-of-use — at the scope of the entire enterprise system rather than a single KVM device.

### 3.2.1 Availability

As discussed earlier, availability constitutes a security feature because one possible attack on your data center includes *denial of service* attacks – the availability features in the Dominion Series increases the likelihood of your ability to recover from such attacks.

Dominion Series availability features include:

- Modem Connectivity: Dedicated modem ports (DB9 connector) for attaching an external modem to certain Dominion models, or integrated internal modems (RJ11 connector) found in other Dominion models; this allows a sysadmin to connect to a Dominion device even when the network is down.
- Dual Ethernet Ports for failover; should one Ethernet switch or interface card fail, the Dominion device remains available.
- Integrated power control, so that users can connect and control Raritan power strips by way of a single, seamless interface. In an emergency, a sysadmin can not only troubleshoot and reboot the device, but can also power-cycle it – securely, and with a full audit trail.

One hidden yet important availability characteristic of Dominion products is that each Dominion Series appliance runs a hardened operating system based on a Linux kernel; and CommandCenter runs the JBoss J2EE environment. While none of these facets of the Dominion Series are ever exposed to the customer (nobody has access to the root-level), the fact that Linux and JBoss are Open Source Software allows Raritan to modify them and strip out unneeded functions to remove areas of potential vulnerabilities.

Open Source Software has other security advantages as well. The consensus in the security community is that an implementation that relies on secret, proprietary algorithms for security is not truly secure (this is called “security by obscurity”). A *truly* robust implementation remains secure even when its protocols and algorithms are published and well-known.

### 3.2.2 Encryption

#### Protocol Encryption

The Dominion Series family incorporates 128-bit SSL/TLS (Secure Sockets Layer / Transport Layer Security) encryption for all communication on the network. When using the KVM-over-IP features of these devices (even without benefit of CommandCenter), all web transactions are automatically switched from the HTTP to HTTPS (SSL-secured) transport protocol.

Dominion Series products use X.509 security certificates out-of-the-box, to provide encryption-based assurance of data secrecy and integrity in KVM communications. Custom and third-party X.509 certificates are also supported on certain models as of press time, and will eventually be supported on all Dominion Series models as well.

### **KVM Encryption**

Unlike competing KVM products, the Dominion Series completely encrypts all KVM information – this includes video information, as well as low-bandwidth keyboard and mouse events.

Two common misconceptions regarding this subject exist:

- (1) “KVM-over-IP video traffic need not be encrypted because each vendor’s video compression algorithm is proprietary”;
- (2) “In KVM-over-IP applications, it is the keystrokes that must be protected, not the video.”

From a security standpoint, these constitute faulty assumptions because:

- (1) Any scheme that relies upon the secrecy of a given algorithm is not truly secure – the algorithm can be easily reverse-engineered, and is for certain already known by those who designed the algorithms themselves;
- (2) A great deal of sensitive information may appear on a sysadmin’s screen, for example, server names, user names, and IP addresses. Therefore, full encryption of video traffic is a very important security feature for digital KVM solutions.

Some alternative solutions fail to encrypt video traffic because the sheer volume of video data (which constitutes the majority of KVM-over-IP traffic), requires robust enough hardware to encrypt video traffic without negatively affecting performance. Dominion Series products implement an innovative approach to address this issue: after converting the analog video signal to digital, Dominion reduces the amount of video data by capturing only the changes to the screen (rather than re-transmitting the entire screen over and over). The video data is compressed, then encrypted for transmission. Finally, the maximum amount of bandwidth used by the KVM system can be configured, which ensures that the KVM solution does not become a bandwidth burden.

### **3.2.3 Authentication**

Each Dominion KX, SX, or KSX device contains an internal authentication and authorization database that applies to the servers it controls. Authorization information is stored using a one-way hash function so that not even sysadmins can see the actual password. In addition, Dominion KX, SX, and KSX devices support external RADIUS authentication/authorization databases.

#### **Dominion KX Local Port Authentication**

Dominion KX offers both remote access to servers over the network, as well as local access to servers from the server rack itself. One important characteristic of

security for a digital KVM switch is whether the security perimeter includes this local rack access port – otherwise, this would constitute a security backdoor to disturbing operations. Many customers overlook this important aspect of digital KVM security until it is too late.

While it is true that one must have access to the data center in order to access a digital KVM switch’s local ports at all, you should always avoid relying solely upon *physical* security mechanisms to protect the server racks. Remember that many people usually have access to the data center, including but not limited to contractors, union installers, and potentially malicious employees.

Three major protections exist for the Dominion KX local port implementation. First, users must login with a username and password to access the local port at all. For convenience, this username and password is the same as that used for remote access over the network – and will authenticate against a remote directory server if your Dominion KX is so configured. Second, all access attempts – whether remote or from the local rack – are fully logged and auditable. Third, users that login to the Dominion KX local port will not see every server connected to the digital KVM switch, but only those servers to which they have access rights.

### **3.2.4 Network Security**

Besides employing encryption, the Dominion Series also restricts the degree to which your digital KVM implementation is vulnerable at all, by limiting the security perimeter through two network protocol best-practices.

First, Dominion Series limits the number of vulnerable TCP Ports possible for attack. Dominion KX, SX, and KSX each utilize only a single, customizable TCP port for its proprietary communications. This is important because the fewer TCP ports exposed, the less likely it is for a hacker can exploit that port without detection; and by allowing customized TCP ports, a hacker cannot simply consult Raritan’s website to know which port to begin exploiting.

Second, Dominion Series employs flexible access control lists to limit the scope of client IP addresses to which Dominion will even respond. This best-practice severely limits the degree to which your security perimeter extends.

### **3.2.5 Configuration and Management**

Dominion KX, SX, and KSX products all provide easy-to-use graphical user interfaces — in both rack-based and remote KVM-over-IP use cases — for remotely configuring network characteristics, ports, devices, user profiles, IP-based access control lists (ACLs), and authentication servers.

These devices also support firmware upgrades over the network, and integrated auditing via a centralized syslog server, allowing them to be remotely monitored and audited by any number of third-party tools..

Of course, in its role as an overall aggregator and manager of Raritan KVM devices, CommandCenter integrates all these products and provides central control of their capabilities, as well as providing more sophisticated tools that can be applied across all the Dominion Series products in the enterprise.

## 3.3 CommandCenter

CommandCenter provides centralized access and management for all Dominion Series devices. It is an enterprise-class control point that provides sophisticated tools for the management of authorization, authentication, and auditing functions across the entire distributed system, including database backup/restore functions for the KVM devices themselves. CommandCenter is accessible via both browser (Java applet) and Secure Shell (SSH2).

The following sections describe CommandCenter's enhanced security-related capabilities, above and beyond those described in the previous section.

### 3.3.1 Authentication Integration and Failover

CommandCenter provides the ability to manage the local authorization/authentication databases in Dominion KX, SX, and KSX products. In addition, CommandCenter supports but does not require, the use of external authorization/authentication servers, including LDAP, LDAPS, RADIUS, Active Directory, and TACACS+.

Moreover, CommandCenter provides order-specified failover support; in other words, should the first external authorization server in the list be unavailable, the next one is used (which can be of the same or a different type), and so on through the list of specified authorization servers.

### 3.3.2 Certificate Management

All connections to the CommandCenter use HTTPS. CommandCenter is installed with an X.509 certificate from Raritan, and allows customers to replace this with their certificates.

CommandCenter allows customers to use the Raritan certificate, to install a different certificate (including a third-party certificate), and to generate a certificate from the root certificate. An installed certificate can be uploaded as a file or modified from an existing certificate (into which the user can cut and paste certificate and private key information).

CommandCenter also supports export of certificates for installation on other machines, by allowing the certificate and private key to be cut and pasted, or saved as a file. CommandCenter can manage certificates used by other components, such as LDAPS (if a customer uses LDAPS for remote authentication, the CommandCenter acts as an LDAPS client and installs the LDAPS root certificate).

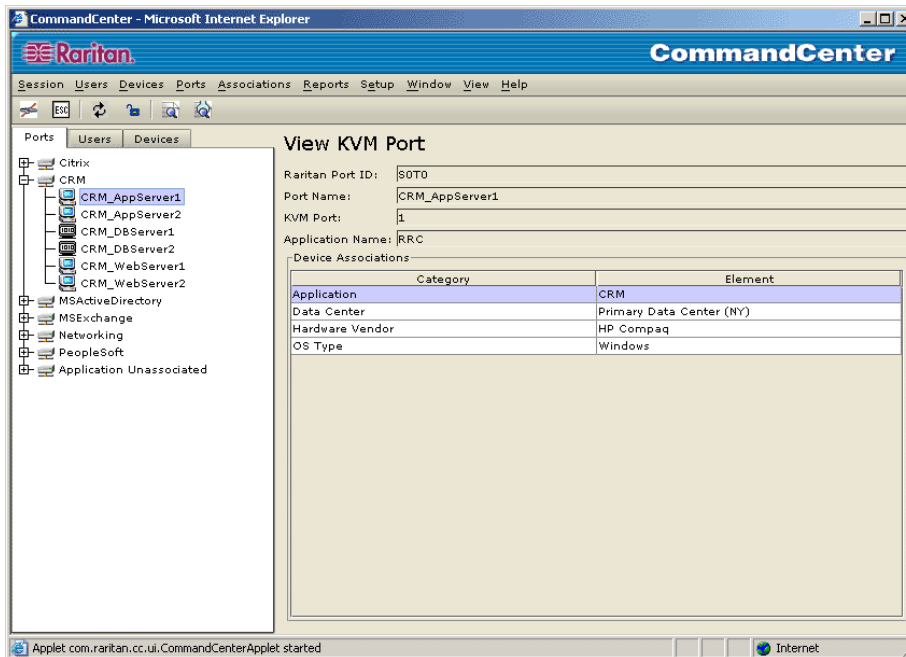
### 3.3.3 The Command Center User Interface

The CommandCenter graphical user interface (GUI) provides sophisticated tools to control all aspects of user and administrative permissions, device and port access, auditing and logging, power control, certificate management and reporting. Its ease-of-use features include online help, context sensitive help, and tooltips.

The true power of the CommandCenter GUI comes from four fundamental capabilities: grouping, attributes, rule specification, and customization. CommandCenter provides useful built-in groups and attributes, and also allows CommandCenter users to define new categories based on attributes. For example, consider this sample of subjects, objects, and attributes under CommandCenter control (*not* an exhaustive list):

- **Role** — which represents a system security role, for example administrator, operator or observer.
- **User** — which represents a sysadmin in one of the roles mentioned above. The user authorization is defined as: 1) the overall system capabilities, available or restricted, based on the roles assigned to the user, and 2) access to ports, available or restricted based on the user groups to which the user belongs; for example, access can be restricted by time of day (e.g., "ANY" or "9AM to 5PM") and/or maximum session duration.
- **UserGroup** — which represents a logical group of users. A user can belong to more than one group, in which case when access rights are evaluated for a particular access request, the most permissive are used.
- **Port** — which represents an edge port.
- **PortGroup** — which represents a logical group of ports.
- **Device** — which is the representation of a Raritan KVM device.
- **DeviceType** — which represents a Raritan KVM device type.
- **ACL** — which is an Access Control List that represents the user (or user group) when enforcing device access rules. The ACL defines the accessible devices, and the permissions (e.g., DENY, VIEW ONLY, CONTROL) for every user who is assigned to a user group.
- **Attribute** — which is a pre-defined, built-in characteristic (like those described above in this list); additional attributes include device name, system name, IP address, serial number (or unique ID) and so on.
- **Category** — represents a user-defined category, for example, location, rack, country and so on.
- **CategoryValue** — which represents a value for a category, and can be a text string or an integer.

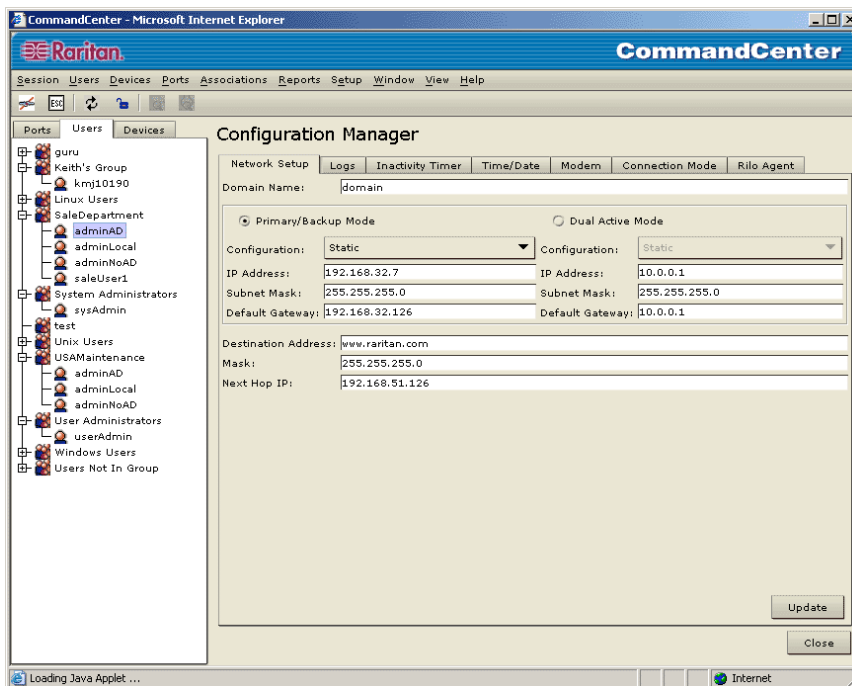
Thus, the CommandCenter GUI allows customized groupings and attributes, which lets you permit or deny access with both bulk operations (including the use of scripts), and with extremely fine-grain control. For example, Figure 1 illustrates the CommandCenter view of a particular device.



**Figure 1: Command Center Device View**

At the lower right of this figure, you see the pre-defined (out-of-the-box) categories that apply to the selected device, and the custom attributes (elements) that have been defined and specified for this device.

Figure 2 illustrates the CommandCenter view of a particular user's configuration information:



**Figure 2: Command Center Configuration View**

### 3.3.4 Rule-Based Control

Administrators specify the rules that control access using Boolean expressions that can contain groups, categories, and attributes. Boolean operators for integer values are "<", "<=", "=", "<>", ">", ">=", and ">". Boolean operators for strings are "=", "<>" (not equal), and "prefix=" to allow matching. A "wildcard" indicator ("\*") is supported for both integer and string expressions.

One can specify access by one or a combination of enumeration and expressions, and these operations can apply to attributes and categories. For example:

```
Rack = 5, 12, 17  
Rack > 1  
Location = NJ-Morristown  
Location = NY*  
OS = Windows*  
OS<>Redhat-9
```

In addition, AND, OR, and NOT operator are provided for use in expressions that have multiple attribute. For example:

- PortGroup1: (**Location = "NY" AND OS = "Windows NT"**)
- PortGroup2: (**Application = "CustomApplet" AND NOT DeviceName = "RestrictedServer"**)
- DeviceGroup4: ((**AdminRights = "Users" AND "Groups"**) **AND DeviceType = "Dominion SX \***)

The rules so specified are processed in order, which provides great convenience. For example, you can restrict access to an entire category (e.g., "**(NOT Location = "New York")**"), then go on to specify rules that allow access to certain servers in the "New York" category.

Figure 3 Illustrates the use of Boolean expressions on custom categories to create a user-specified group of ports (in this case, those running Windows or Linux, and located in the United Kingdom).

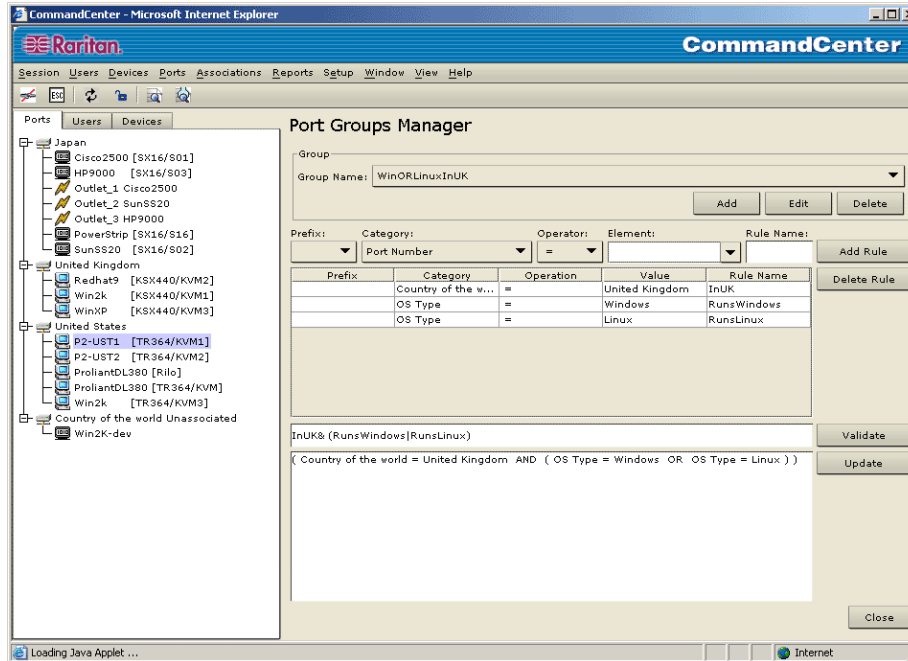


Figure 3: Custom Group Specification

### Flexible Policies Effect High Security

In short, this implementation of security policies (based on custom-defined attributes) allows users to create authorization rules as simple or complex as they truly need them. More importantly, though, these policies are easy to maintain and comprehensively applied. For instance, in the above example, if a new server were added to the system that is located in New York and running Windows NT, it automatically falls under "PortGroup1" – the security policy is effective immediately, with essentially no effort, and sysadmins need not worry about any loopholes.

### 3.3.5 Auditing/Logging

CommandCenter controls the information that is captured in both the audit trail (which includes application errors) and the system log (which includes system errors). This provides a single point of access to the normal system audit trail, system accounting, troubleshooting/debugging aids, and forensic analysis to document both direct and circumstantial evidence in the event of a penetration.

Log filtering includes by-date and time-of-day constraints. For example, a customer might want detailed logging during hours when no changes are expected, but less rigorous logging during normal maintenance windows.

While CommandCenter allows great flexibility in controlling audit records, useful defaults are provided and some functions are permanently present. For example, no one can turn off the logging of changes to logging functions.

Log entries include time of day, source IP address, user ID (if available), group(s) and device(s) affected and any other items relevant to the logged event. Log views are filtered such that the viewer does not see any entries he or she is not authorized to see.

A partial list of supported audit log events includes:

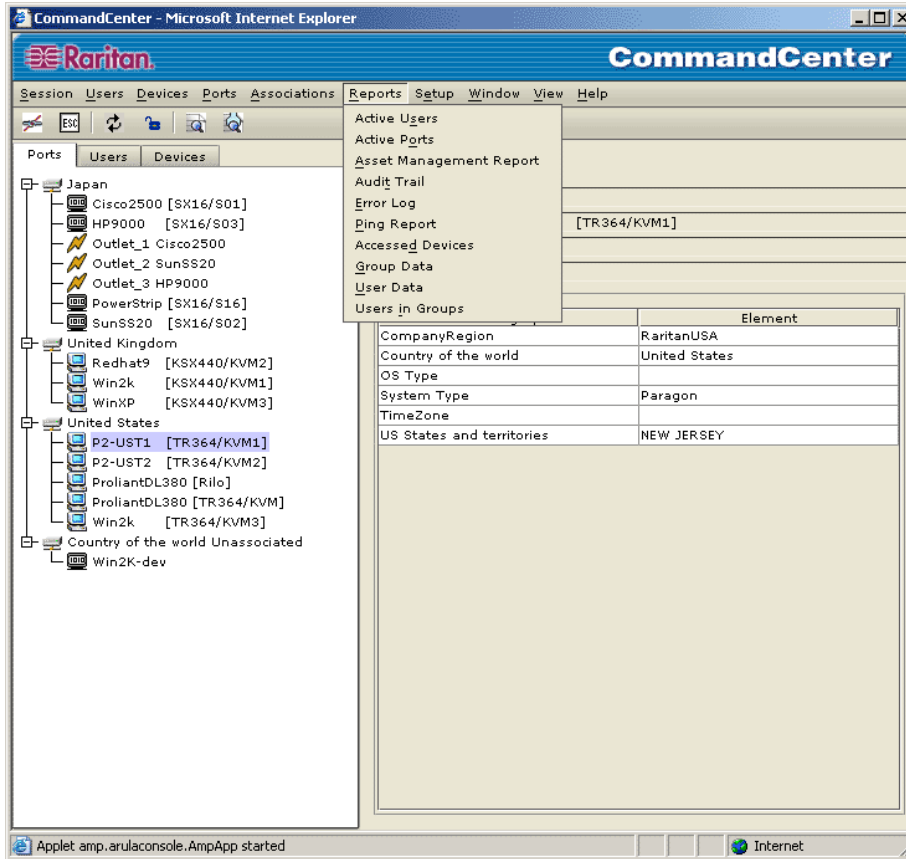
- All CommandCenter administrator transactions, which log before-and-after entries for parameters that have been modified. For example: adding a new device, changes to an NTP server (new and old IP addresses are logged), changes to user permissions and download/installation of firmware.
- All logins and logouts from CommandCenter itself, including reason (e.g., user logoff, inactivity timeout, force off by admin, network disconnect, etc.)
- All device connects and disconnects are logged, including device name, IP address, and the reason for the connect/disconnect
- All connects and disconnects from target systems
- All commands to Raritan-supported power strips (through a target system or directly to the strip)
- All failed attempts for any of the above events, including as much as is known about the cause of the failure

Appropriate severity levels (Notice, Informational, Error, Debug and Critical) are assigned to events and can be used to filter audit reports.

### **3.3.6 Reporting**

CommandCenter provides sophisticated reporting capabilities. (Note that the report and customization descriptions below assume Admin View access to the audit/logging database).

Because reporting encompasses various areas of the product family, it is impossible to include all the reports that every user might want or need. Raritan's strategy is to provide a useful set of built-in reports, and to allow customers to create custom reports from the same log data. The built-in reports available for ports are shown in Figure 4 .



**Figure 4: CommandCenter Built-In Reports**

In addition, built-in reporting covers:

**Users and Groups**—including user data (list of all users and the data associated with them), No Password (list of all user IDs that do not yet have passwords), Group Data (list of all groups and the data associated with them), Users In Groups (list of all groups and the users in each of them), Notifications (report of all methods of notifying users, emphasizing users with no notification yet) and Active Users (list of all active users and the ports and target names to which they are logged in).

**Ports, Applications, and Plugins**—Ping report (ping status of all or a subset of machines), Edge Port (list of all edge ports and their status, Device Profiles, Firmware Report (hardware, firmware, and software versions; model and serial number, optional historical list of firmware updates), Application Report (list of applications, import dates, and feature descriptions), Accessed Ports (edge ports accessed in the last day, week, month, or user-specified period), Active Ports (ports and logged-in users), Accessed Ports By User/Group (edge ports accessed in the last day, week, month, or user-specified period, by user or group), Asset Management (list of all Raritan devices accessible via this CommandCenter, with name, ID, IP address, serial number and any additional available information).

All these reports have a “Save As” option, which allows users to save the report out put in either the format as viewed or CSV (comma separated value).

## 4 Conclusions: Facing the Future

Protecting enterprise computer systems against changing security threats is a never-ending task. Raritan is the best choice for partnership in pursuing this goal because it is a forward-looking, security-aware company with unmatched experience in providing KVM solutions. Raritan addresses current and potential security problems with a disciplined approach to architecture and an innovative approach to engineering.

Throughout its 19-year history, Raritan has led the industry in inventing new approaches to KVM capabilities and security. For example, Raritan pioneered the use of Cat5 cabling for KVM solutions, which allows a much higher concentration of cables while retaining extremely high-quality video.

Raritan also invented the concept of Computer Interface Modules (CIM) that prevent servers from “locking up” when a KVM connection is removed, by providing a keyboard/mouse emulation function that acts as a “heartbeat” to keep the server working. Furthermore, Raritan’s unique approach to encrypting video information, as well as low-bandwidth keyboard and mouse events, provides the only available solution to protecting the information that appears on a sysadmin’s screen.

Most important, Raritan provides innovation and a “best practices” mind-set at the architectural level, from the beginning of every product’s design phase. The use of formal modeling techniques in designing KVM solutions (UML), the adoption of Open Source components in implementing those solutions, and the level of importance assigned to human factors, together provide much greater assurance that Raritan’s distributed system security mechanisms will work as intended.

The choice of Raritan as a partner in distributed systems security means that your enterprise has the highest level of KVM security available in the industry today. And Raritan’s long history of innovation, best practices, and forward-looking approach to security means that you will continue to stay several steps ahead of the continually evolving threats to system security.

