

# The Secure Console Whitepaper

Browser-based, Command Line  
Interface, or Both?

Raritan Computer Europe B.V.  
November 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

Secure console management has become an essential requirement for operating today's complex data centers and managing assets across distributed environments. IT organizations need to give their core team of skilled sysadmins ready access to critical business computing resources on an anywhere/anytime basis. If that access is not adequately provisioned, core systems may under-perform or fail – with potentially disastrous consequences for the business.

Vendors have generally offered IT a choice between two different approaches to console management: terminal servers or browser-based solutions. Each of these approaches has its own appeal. Terminal servers use a familiar command line interface (CLI) and can be effectively secured using a secure shell (SSH) protocol. Browser-based solutions offer the advantages of ubiquitous desktop/mobile Web access and standardized protection under the Secure Sockets Layer (SSL) protocol.

Other capabilities further differentiate these two approaches. Depending on the vendor, each approach may support different terminal emulations – and therefore different platforms and devices. A browser-based product may not provide for out-of-band access in the event of a denial-of-service attack. An SSH-enabled terminal server may offer little or no port buffer caching, rendering it unable to deliver the console messages and alerts sysadmins need to effectively administer a remote system.

With pros and cons accompanying both approaches, IT decision-makers may face a difficult choice in selecting a strategic secure console architecture for the enterprise.

Fortunately, they no longer have to make such a choice. Raritan's Dominion SX combines terminal server capabilities and browser-based device access in a single, integrated package – allowing IT organizations to dynamically apply the best approach for any given task or situation. By leveraging this unique combined solution, IT can optimize sysadmin productivity, ensure the security of the enterprise environment, and deliver the high service levels so essential to the bottom-line performance of the business.

# Table of Contents

1	The secure console .....	1
2	CLI vs. the browser .....	3
3	The best of both worlds .....	5
4	About Raritan .....	7

# 1 The secure console

Today's enterprise computing environments are large, complex and geographically distributed. They're also critical for the moment-by-moment operation of the business. That's why it's essential to provide sysadmins with effective remote access to enterprise computing resources. Sysadmins must be able to apply their skill, expertise and experience wherever and whenever it's needed – especially as enterprise infrastructure continues to grow faster than sysadmin headcount.

It is also essential to provide this access in a secure manner. Gone are the days when an insecure telnet session would be considered an acceptable practice. Corporate data and systems resources are just too valuable to be put at risk by having sysadmin logins and passwords traveling over the network “in the clear.” Regulations such as the Health Insurance Portability & Accountability Act of 1996 (HIPAA) also mandate that proper measures be taken to protect sensitive data. So security has become an integral component of any remote console management solution.

To make the remote terminal server secure, IT gave up the ubiquity of plain telnet (which was universally available on Unix boxes as well as some Wintel platforms) to implement the SSHv1 protocol. Most organizations used pioneering shareware/freeware solutions such as Tera Term and Putty. Some IT organizations did this with a commercial SSH client. These SSH clients communicated in a secure manner with the secure console server (SCS), allowing sysadmins to manage systems remotely without compromising corporate assets. Strict separation was therefore maintained between the “managed entity” (i.e. the server) and the SCS. At the same time, the SCS user appears to the managed entity as a local, serially connected terminal.

However, in addition to any licensing cost for commercial SSH software, this approach generated overhead costs associated with administering and managing distributed SSH clients. In other words, remote terminal capabilities were no longer ubiquitous or free.

This technology continued to evolve. In June 2000, SSHv2 was introduced to the market and rapidly accepted, since it resolved many of the security vulnerabilities of the SSHv1 protocol.

Another technology – one that did not require a licensed client, with all of its associated costs and administration headaches – then emerged. This was the new ubiquitous Web browser.

Arula Systems, a spin-off from Hewlett-Packard, was the first to market with a browser-based secure console server that used the SSL protocol, SSL certificates, and RC4 encryption. Arula also leveraged a then-emerging technology called Java to provide “signed” applets, which could be dynamically downloaded from the secure console server to the client browser in order to enable a secure remote management session. This protected the security of the session and ensured that the client was communicating with the right secure console server.

Early adopters quickly validated this new model. Microsoft's Java Virtual Machine (JVM) provided the requisite functionality on the dominant client operating system, Windows. Clients could thus communicate with the SCS without the need for client software and all of its associated costs and administration issues. The Java applet could also include resizable CLI console windows and support useful features such as copy-and-paste and client-based logging.

The encrypted, SSL browser session has now become the de facto standard security model for all types of Web interactions, including online banking and retailing. IT organizations thus have a choice between two secured approaches to remote console management: the SSH-secured terminal server and the SSL-secured browser.

## 2 CLI vs. the browser

Both SSH-secured terminal server solutions and SSL-secured browser solutions have their strong points and shortcomings. Many IT organizations already have secure terminal server solutions in place and therefore have a high comfort level with the technology. SSH clients may also support a broader range of terminal emulations than browser-based solutions. Most servers and networking equipment use VT-100 emulation, which originated with DEC and implements the ISO646 character set (US ASCII). For European applications, however, the operative character set may be ISO8859-1 or ISO8859-15. Some special applications may even dictate use of other terminal emulations, such as VT320, that are not generally available on the server side from SCS vendors. In these situations, the broad emulation support of the SSH client are important.

In addition, because server-side SCS applications may not natively support all the languages and character sets required by a global IT organization, the ability of SSH clients to support foreign languages – especially double-byte languages such as Japanese, Chinese and Korean – is sometimes necessary for effective communications with managed devices.

On the downside, as noted above, SSH clients require licensing, installation and administration. This not only adds cost and labor – it also limits where and when SSH clients can be used. Sysadmins who are away from their desks and don't have their personal laptop or other SSH-enabled mobile device with them can't do their jobs if called upon to do so.

SSH client solutions also typically lack the caching capabilities and intelligence necessary to deliver critical alerts (including TCL scripts and SNMP traps) to sysadmins. Events such as “processor utilization greater than 99% for more than 3 minutes” may occur on a system when the sysadmin is not logged on to a system port. When the sysadmin does connect, having that information is critical to proper system administration and/or troubleshooting. With a low- or no-cache SSH-enabled terminal server, sysadmins will not receive those alerts. The severely limited caching offered by SSH clients also makes it impossible to browse console log histories of any reasonable size.

The administration of SSH-based clients may also not integrate with existing enterprise directories and/or Authentication, Authorization and Accounting (AAA) mechanisms, such as RADIUS, Lightweight Directory Access Protocol (LDAP), Active Directory and Cisco Systems' TACACS+. This can prevent IT organizations from realizing the operational efficiencies and implementing the security best practices that require common security administration across all IT functions.

SSL-secured browser solutions, on the other hand, provide sysadmins with access to managed systems from any Java-enabled device with Internet access. So they can potentially address an emergency situation on a moment's notice from a hotel business center, an airport lounge, a borrowed PDA – or even a sufficiently sophisticated cell phone. This client-free solution also eliminates the costs and administration associated with SSH solutions.

Browser-based solutions offer many other advantages as well. Because they can fully leverage the intelligence of the operating system, these SCS solutions can provide the necessary data caching to store complete port buffer histories. This provides sysadmins with full visibility into alerts, traps, scripts and logs. Such information is invaluable for ensuring the health of critical systems and/or quickly performing urgent troubleshooting tasks.

Browser-based solutions also provide a graphical user interface, which can make sysadmin tasks easier and more intuitive. They offer flexible encryption options, since encryption engines can be dynamically downloaded to client machines in the form of Java applets. And they can allow integration into existing directories and AAA mechanisms – resulting in simplified administration and improved compliance with security best practices.

There are, admittedly, some limitations to current browser-based SCS solutions. These include limitations in the range of terminal emulations and languages supported natively on the client side. A browser-based solution may also not be as appropriate for use on a local data center “crash cart” as a basic CLI terminal server.

Thus, if forced to choose between a secure terminal server and a secure browser-based solution, IT organizations might find themselves with an unacceptable gap in their SCS functionality. Under today’s extreme operational pressures, such a gap could limit the ability of IT to make optimal use of its sysadmins’ skills – exposing the business to the risks associated with underperforming critical services and downtime.

## 3 The best of both worlds

Actually, IT organizations do not have to make such a difficult choice. SCS solutions are available today that offer both SSH- and SSL-based security – allowing sysadmins to use either secure terminal server or secure browser approaches as appropriate. By combining these two complementary access modes, this new breed of SCS solution delivers the full range of capabilities IT organizations require to effectively and efficiently keep their computing environments in peak condition.

Specifically, the ability to interchangeably employ both secure CLI and secure browser access provides IT with the following benefits:

### **Anytime/anywhere access to managed resources**

By enabling sysadmins to manage critical systems regardless of where they are and what type of device they can access at the moment, hybrid SCS solutions ensure that critical services won't be jeopardized by contingencies of time and location.

### **Ironclad security**

Regardless of which mode is implemented for any given remote session, IT managers can be sure that only authorized sysadmins are given access to enterprise systems – and that all session data will be properly encrypted as it traverses network connections.

### **Reduced client administration costs**

By leveraging the ubiquity of the Java-enabled browser, IT is no longer forced to purchase and manage client software for every end-point from which it wants to enable sysadmins to manage enterprise resources.

### **Universal device and language support**

A hybrid SCS solution can support virtually any type of managed device by enabling special terminal emulations and/or foreign languages to be installed on specific clients as required.

### **In-band and out-of-band connectivity**

Out-of-band access via the console port (as opposed to in-band access via the server's Ethernet connection) allows sysadmins to use out-of-band connectivity as a failover option in the event of a network failure or DoS attack, ensuring their ability to manage critical resources under any conditions. A built-in modem – another out-of-band solution – should be standard for data center and remote office applications.

### **Full alerting and log availability**

With sufficient cache size, sysadmins can get all the information they need to accurately assess the status of managed resources and to quickly troubleshoot system problems.

### **Integration with enterprise directory and AAA standards**

This integration streamlines administration of access rights and eliminates the potential security lapses that can occur when such rights are managed in multiple separate directories.

It is important to note that not all SCS solutions that claim to be “browser-enabled” actually provide all these benefits. Some vendors use that term merely to mean that the SCS itself can be configured from a browser interface. That is not the same thing as enabling all the systems to which the SCS is providing access to be administered and managed from a browser interface. IT buyers must pay close attention to this critical distinction.

Also, some SCS solutions only offer narrow browser support – typically Microsoft Internet Explorer running on Microsoft Windows XP. This limited browser support is insufficient to deliver the benefit of anytime/anywhere access, especially as changes in the market make the use of non-Microsoft solutions more commonplace.

However, beyond the features and capabilities that differentiate one SCS solution from another, the most important concept for IT decision-makers is this: No systems administration team should find it necessary to choose between terminal server and browser-based SCS solutions. The optimal strategy is to deploy both. That way, IT can best ensure the availability of critical services to the business – while making the best possible use of its own finite resources of people, time and money.

## 4 About Raritan

Raritan Computer Inc., based in Somerset, N.J., is a leading supplier of IT infrastructure management solutions for secure access, monitoring, and management of servers and other IT devices in data centers and remote offices. Raritan's products are used to control and manage millions of servers at more than 50,000 data centers, computer test labs, and other sites around the world. From the small business to the enterprise, Raritan's complete line of compatible and scalable digital and analog KVM, serial console, and remote connectivity products offers IT professionals the most reliable, flexible, and secure solutions to manage IT equipment, while improving operational productivity. Raritan's Peppercon OEM division develops manufactures and markets advanced, hardware-based, remote-management products based on digital KVM-over-IP and IPMI technology. Founded in 1985, Raritan has experienced 19 consecutive years of profitable growth and technical innovation. Raritan has 31 offices and it's products are distributed in 76 countries. More information on the company is available at [Raritan.com](http://Raritan.com).

# # #

All marks are the property of their respective owners.