

RICHARD W. PICKETT, JR.
440 Millwood Court
Bowling Green, KY 42104
Richard.Pickett@CSRTechnologies.com
(270) 303-9154

SUMMARY:

Twenty+ years experience in Information Technologies including Information Security, Clustering and HA implementations, UNIX and Linux Systems Administration, Programming, Networking, Data Center Operations, Application Design and Development, and Windows Domain Management.

SKILLS SUMMARY:

Operating Systems: Linux, HP-UX, Sun Solaris, Windows Servers and Desktop versions.

Linux Distros: Red Hat, Centos, Debian, Ubuntu, Slackware.

Major Applications: VMWare ESX, Apache, Mysql, Tomcat, ClamAV, NAILS, ActiveMQ, Postfix, Sendmail, Exim, BIND, keepalived, PEN, DRBD, GNBD, OCFS2, GFS.

Programming: C/C++, Bash, PHP, Javascript.

Certifications: Security+ and CISSP training.

EXPERIENCE:

Sr Linux Server Administrator

L1 Identity Solutions: November 2010 to Present

Open Network - Hardened server installations for security accreditation according to DoD STIGs ♦ Implemented a 3-tier, 12-instance Linux cluster on ESX Server behind two LB F5s ♦ Configured 6 tomcat-based JSP instances ♦ Configured 6 PEN instances for small-footprint FOLB ♦ Configured two ActiveMQ pairs for FOLB message brokering ♦ Configured remote Mysql slave with table-based triggers to import activity from DSP into Open Network ♦ Configured 6 postfix instances for security separation and both in- and out-bound virus scanning ♦ Configured 3 PRDMs shared between Linux VMs for OCFS2 cluster file-sharing ♦ Integrated Linux login with Likewise Open to give SSO for all accounts ♦ Configured nagios and monit for alerting and automated incident-response ♦ Coded deployment scripts in bash to stagger-deploy new application code without introducing downtime ♦ Coded scripts in bash to allow central control of all application functionality and eliminate errors introduced by manual commands ♦ Performed daily maintenance and operations procedures as needed.

DSP - Managed a LB-pair and 4-node Red Hat cluster serving apache, PHP, mysql, and SMB ♦ Replaced DRBD-based mysql redundancy with a mysql multi-

master instance for increased performance, failover, and reduced support issues without needing additional hardware ◆ Replaced complex GFS-over-GNBD-over-DRBD clustered file-system with a smaller OCFS2-over-DRBD instance for live-updated files and a distributed local copy of all static files on each cluster node to increase performance and reduce support issues without needing additional hardware ◆ Optimized existing PHP code, often reducing nightly processing runtimes from hours to minutes ◆ Optimized mysql configuration as well as table definitions, increasing performance and system stability ◆ Streamlined End-of-Month processing, increasing accuracy, automating manual processes, and reducing run time.

Linux Server Administrator / PHP Developer **IO Studio: September 2008 to November 2010**

Path To Honor .mil Migration - Managed migration from .com environment to .mil environment colocated in the Pentagon, allowing storage of sensitive PII and direct data feeds with military systems ◆ Configured and maintained RHEL LAMP stack to serve the .mil domain in a IBM blade server VM environment ◆ Configured and maintained LinuxShield to scan the system nightly and report any security threats ◆ Audited code and procedures to ensure all applicable DoD STIGs were met ◆ Implemented DoD-compliant methods of updating code and maintaining the servers ◆ Implemented auditing procedures to evaluate errors in real time ◆ Designed automated system to synchronize constantly changing .com data files (20Gig) through the Pentagon's multiple layers of security during the month-long migration process ◆ Implemented CaC-based authentication mechanism to meet DoD STIG requirements for PII data accessed via https ◆ Coordinated fail-over procedures between Pentagon servers and COOP, the .mil off-site fail-over facility ◆ Configured mysql database to use a mounted SAN for failover, the SAN was always kept in sync with the COOP, on failover the COOP mysql instance instantly started up using the same SAN ◆ Researched system packages for security-related suitability ◆ Modified .com production procedures to meet DoD compliance ◆ Managed and performed maintenance procedures as needed.

Path To Honor Server Administration - Ensured 99.99% uptime with no unscheduled downtime on a Debian-based LAMP stack load-balanced configuration ◆ Facilitated and administered two separate complex code and database separations that resulted in the detachment of several different applications that used many overlapping sections of code and database tables - 630,000+ lines of code and 400+ database tables spread over 3 databases ◆ Managed weekly life-cycle code releases ◆ Implemented a patch system that updated live servers with new code within seconds, replacing a 20-minute manual process ◆ Designed and implemented release-cycle procedures to manage SDLC, reducing management procedures to minutes instead of hours ◆ Developed supplemental scripts and procedures to speed up repetitive administrative tasks ◆ Configured and maintained all web-server essential and supporting applications.

Path To Honor Lead Programmer - Researched and coded in PHP a class that provides key statistics to the Path To Honor, the AFQT assistance program designed for the National Guard ♦ The PHP code uses complex MYSQL queries and produces output that is imported directly into charts and graphs by Microsoft Excel ♦ Researched and coded bug fixes as needed.

WAP / WML - Researched, designed and coded in PHP a class that detects the current browser type using WURFL and sends html and css designed specifically for that browser's capabilities ♦ Analyzed the data of over 381,000,000 log entries to determine browser capabilities and recommend specific capability sets based on browser popularity.

G-RAP - Designed and coded a tool in PHP to interface with the Guard Recruiting Assistance Program MYSQL database allowing staff to quickly make date and status changes ♦ Coded in PHP double-checking methods to ensure the accuracy of the data and prevention of inconsistent and erroneous entries.

CRT - Researched and coded in PHP on the Zend framework a queue system that imports National Guard candidates and separates them according to call center and the candidate's activity ♦ Many features of Zend were utilized including Zend_Acl, Zend_Auth, Zend_Form (including validation), Zend_Layout, Zend_Mail, Zend_Navigation, Zend_Paginator, Zend_Registry, Zend_Session, and Zend_Soap ♦ Implemented Doctrine as the ORM instead of Zend_Db.

Apache Request Routing - Researched, designed, and coded in PHP a class that utilizes a MYSQL database table to quickly route incoming requests to their appropriate PHP handler scripts ♦ Designed and coded in PHP an interface for non-technical staff to add, delete, and modify the priority of routing entries without adversely affecting the website.

Information Systems Security Consultant Nortel Networks: November 2000 to February 2007

Achilles - Designed and coded in C++ a multi-threaded Windows LSA filter to integrate the Windows PDC password change mechanism with NorPASS, Nortel's industry leading Single-Sign-On infrastructure, this enforces Nortel's NorPASS password rules which are much more stringent than Window's ♦ Implemented the Windows IO Completion Port (IOCP) to scale worker threads to handle the varying load of incoming password change requests ♦ Used Critical Sections and Events to implement optimized multi-threaded synchronization ♦ The LSA filter was installed on 20+ PDCs and handled all of Nortel's Windows password change requests (100,000+ users around the world).

Cerberus - Designed and coded in C++ a UNIX-based Apache web server module (plug-in) that integrates Apache 1.3 and 2.0 servers to use NorPASS for SSO infrastructure for Authentication, Authorization, and Access controls on hundreds of Nortel's internal web servers and applications. The Cerberus plug-in implemented multi-threaded caching of user data to speed up the re-authentication of returning users. The Cerberus plug-in checked for the existence

of PKI-signed cookies and verified the signature of the cookies to authenticate user requests.

Project ADS - Designed and coded in C++ the Active Directory Syncer (ADS) which runs as a multi-threaded service on Windows Domain Controllers, receiving password synchronization commands from NorPASS to set Windows account passwords when users change their password in the NorPASS account management mechanism ♦ Used Critical Sections and Events to implement optimized multi-threaded synchronization ♦ Interfaced with Active Directory via LDAP to lookup accounts and synchronize passwords ♦ Used Windows impersonation to assume the roll of an account in order to change it's password.

Issachar - Designed and Coded in C/C++ a multi-threaded ISAPI DLL bulk password-change tool that allows certain administrators to change passwords on a list of users to a known value for PC upgrades and back to the original passwords after the upgrade has been completed ♦ Implemented Critical Sections and Events to pass data from IIS threads to the worker threads that performed the password changes.

Performed day-to-day operational duties as needed including monitoring, troubleshooting, and maintaining Nortel's Intranet security mechanisms; worked with departments and users on security issues; provided level 3 (final) support for security issues.

Windows NT/2000 Server System Engineer

Nortel Networks: July 1998 to November 2000

Helmsman - The Helmsman Server is a back-end product that performs text queries on collections of documents and returns the resulting document list as well as the documents themselves as they are selected ♦ The Helmsman Server is the de-facto document delivery tool used Nortel-wide including the Nortel Networks Internet site ♦ The Windows NT Helmsman server was wrote to replace the existing UNIX-based server to save thousands of dollars per year in hardware, maintenance, and support ♦ Researched and documented, using UML, an existing document server written for UNIX ♦ Designed and implemented a Windows NT-based multi-threaded server to replace the existing UNIX server ♦ Designed the Windows NT Helmsman server to run as a NT Service and uses the SCM to communicate with the user for setup/configuration via a MFC GUI ♦ Coded the Helmsman server to communicate with multi-OS based clients via TCP/IP ♦ Implemented asynchronous (overlapped) IO Completion Ports (IOCPs) with the Winsock 2.0 API to avoid costly buffer manipulation and expensive thread context switching ♦ Implemented IOCP mechanisms to send transactions internally from communication threads to worker threads; IOCP is highly optimized with the NT thread scheduler to minimize CPU usage, maintaining a drop-proof transaction transfer mechanism from the communication threads to the worker threads ♦ Implemented memory mapped files to avoid costly reads and buffer copies when sending requested documents ♦ Implemented the Verity search engine to perform search queries and list documents meeting the search

criterion ◆ Implemented NT Performance Monitor (PerfMon) information updates to allow remote monitoring.

NorPASS API - Tested the NorPASS API for multi-threading capability ◆ During the Windows Password Integration project it was discovered that the NorPASS API was not thread safe ◆ Provided a multi-threaded test application to test NorPASS API stability ◆ Configured OpenSSL, the SSL library used by the NorPASS API, to compile in various modes, including assembly and debug modes ◆ Integrated the OpenSSL, NorPASS API, and NorPASS API test application into one VC++ workspace for rapid building and debugging ◆ Provided low level technical consultation on the causes of multi-threaded bugs and the complexities of discovering multi-threaded errors ◆ Built, debugged, and fixed multi-threaded bugs.

Independent Contractor November 2001 to Present

Installation and support of the following systems. Skill areas focused on Security Engineering, Linux Server Administration, and Programming.

Phalanx Security System - Designed and coded in C/C++ a Linux-based multi-threaded daemon that monitors open-source security applications and the system logs to detect and deter Internet-based attacks and probes - since it's inception in 2001 over 8 million documented attacks and probes have been stopped without a single security incident ◆ Designed and coded a C++ module interfaced with Linux iptables/netfilter, managing all of the firewall rules of allowed traffic, their configured bandwidth rates, and blacklisted IPs ◆ Designed and coded a C++ module that monitors various open-source security systems like snort and tripwire for changes in the system and activity that may indicate an attack ◆ Designed and coded a C++ module that monitors the system log for failed login attempts in ssh, FTP, POP3, and SMTP, once the failed attempts from a single source or range of IPs reached a configured threshold, the module interfaced with the iptables module to have that IP blacklisted ◆ Designed and coded in C++ a module that emails alerts when states in the system change, including configuration files, network link status, and attack thresholds.

Federated ID Management System - Designed and coded in PHP/MySQL a Single Sign-On (SSO) mechanism that implements a single-token-per-user across an infinite number of diverse systems ◆ Implemented PKI signed cookies for secure web-based tokens ◆ Implemented HTTP redirection so when a member visits a site where authentication is necessary and didn't have a user token they would be sent to the login mechanism at the management domain; upon authentication they would be issued a signed identity cookie (token) within the management domain, then redirected with a PKI signed command to the domain they were attempting to access; the signed command would authorize the local domain to issue an identity cookie (token) in that domain; then they would be returned to their original URL ◆ If a visitor didn't have a local domain token, but had already been issued a management domain token, they would not be

prompted to authenticate by the management domain, instead it automatically issued a signed command authorizing the local domain to issue an identity cookie

- ◆ The management domain implemented name/password authentication as well as PKI certificate authentication; when used in combination (configurable by the local domain) they create a 3-factor authentication; when used separately it provides a fast way to authenticate - only the cert would need to be checked to authenticate the user
- ◆ Signed commands and the tokens were time-stamped to prevent them from being saved and used again in the future.

References: Provided on request.