

## UNIX<sup>®</sup> and Linux<sup>®</sup> Security: Hands-On - 4 Days Protecting Against System and Network Intrusion

*Course 433 Overview*

### You Will Learn How To

- Secure UNIX and Linux systems from internal and external threats
- Control authenticated access to local and remote resources
- Scan servers for vulnerabilities and correct the problems that are found
- Reduce security risk by limiting superuser privileges
- Configure tools and utilities to minimize exposure and detect intrusions
- Tackle security problems by swapping out insecure software components

### Course Benefits

The UNIX family of operating systems, including the Linux versions, is prized by IT professionals for its flexibility and openness. However, vulnerabilities can make UNIX systems susceptible to information assurance threats. In this course, you gain the skills needed to secure your UNIX and Linux platforms. You learn to use tools and utilities to assess vulnerabilities, detect threats and provide effective access controls.

### Who Should Attend

UNIX systems administrators and others responsible for deploying secure open systems. Knowledge of Linux or UNIX at the level of Course 143, "Linux Comprehensive Introduction," or Course 428, "UNIX Comprehensive Introduction," is required.

### Hands-On Training

Hands-on experience in securing UNIX and Linux systems is provided throughout this course using Red Hat<sup>®</sup> Enterprise Linux<sup>®</sup>, Solaris and BSD. Exercises include:

- Scanning systems for network vulnerabilities with Nessus
- Detecting weak configuration settings with Suse
- Analyzing compromised systems to help prevent attacks
- Enforcing password quality and user account usage policies with PAM
- Configuring OpenSSH servers and clients
- Securing limited administrative privileges with **sudo**

# UNIX<sup>®</sup> and Linux<sup>®</sup> Security: Hands-On - 4 Days

## Protecting Against System and Network Intrusion

Course 433 Outline

### UNIX and Security

#### Achieving UNIX security

- Detecting intrusions with audits and logs
- Avoiding security loopholes
- Discovering software vulnerabilities and configuration errors

#### Protecting data and systems with cryptography

- Pretty Good Privacy (PGP)
- Gnu Privacy Guard (GnuPG)
- Authenticity and integrity through digital signatures and cryptographic hashes

#### Protecting User Accounts and Strengthening Authentication

##### Controlling secure account usage

- The UNIX login process
- Enforcing password quality and account use policy
- Controlling access with Pluggable Authentication Modules (PAM)
- Logging all account access and login failures

##### Monitoring and disabling accounts

- Tracking account usage
- Managing user and group IDs
- How and when to disable accounts

##### Logging in across the network

- Risks of application protocols
- Providing strong user authentication with cryptography and tokens
- Tunneling application protocols through SSH

#### Reducing Exposure to Threats by Limiting Superuser Privileges

##### Controlling root access

- Configuring secure terminals
- Preventing insecure network access
- Gaining **root** privileges with **su**
- Using groups instead of **root** identity

##### Auditing superuser activity

- Limiting access to privileged accounts
- Detecting misuse and attacks with log files

##### Role-based access control (RBAC)

- Risks of UNIX all-or-nothing access
- RBAC in Solaris

- Adding RBAC with **sudo**

#### Safeguarding Vital Data by Securing Local and Network File Systems

##### Directory structure and partitioning for security

- Files, directories, devices and links
- Employing read-only file systems
- Ownership and access permissions
- Immutable and append-only files
- Identifying NFS vulnerabilities

##### Backup and integrity testing

- Safeguarding backed-up data
- Detecting intrusions with Tripwire

##### Hardening UNIX systems

- Increasing information assurance with yassp, TITAN and Bastille
- Scanning for network vulnerabilities with Nessus
- Detecting weak configuration choices with Sussen

#### Avoiding the Exploitation of Programs

##### Risks from unwanted program execution

- Starting programs surreptitiously
- Running programs as other users
- Scheduling jobs with **cron** and **at**
- Minimizing start-up script vulnerabilities

##### Reacting to attacks and intrusions

- Finding signs of intrusion in syslog data
- Analyzing a compromised system
- Reducing the effects of buffer overflow exploits

#### Minimizing Threats to Network Services

##### TCP/IP and its security loopholes

- Sniffing passwords with **Ethereal** and **dsniff**
- Testing network exposure with **netstat**, **lsof** and **nmap**

##### Securing internal network services

- Enabling enhanced logging
- Configuring OpenSSH and OpenSSL
- Network authentication using Kerberos
- X Window System vulnerabilities/solutions

##### Safely connecting to external networks

- Controlling and logging server access with **TCPwrappers** and **xinetd**
- Reducing information leakage
- Securing FTP, e-mail and Web access