SEC506: **Securing Linux/Unix**

GCUX
Unix Security Administrator
giac.org/gcux

6 Day Program	36 CPEs	Laptop Required

You Will Learn

- Significantly reduce the number of vulnerabilities in the average Linux/Unix system by disabling unnecessary services
- Protect your systems from buffer overflows, denial-of-service, and physical access attacks by leveraging OS configuration settings
- Configure host-based firewalls to block attacks from outside
- Deploy SSH to protect administrative sessions, and leverage SSH functionality to securely automate routine administrative tasks
- Use sudo to control and monitor administrative access
- Create a centralized logging infrastructure with Syslog-NG, and deploy log monitoring tools to scan for significant events
- Use SELinux to effectively isolate compromised applications from harming other system services
- Securely configure common Internet-facing applications such as Apache and BIND
- Investigate compromised Unix/Linux systems with the Sleuthkit, lsof, and other open-source tools
- Understand attacker rootkits and how to detect them with AIDE and rkhunter/chkrootkit

“Linux security courses are a rare commodity and a valuable resource to the security professional.”

— Trevor Sellers, **IDA Center for Communications Research**

Course Preview
available at: **sans.org/demo**

This course provides in-depth coverage of Linux and Unix security issues that includes specific configuration guidance and practical, real-world examples, tips, and tricks. We examine how to mitigate or eliminate general problems that apply to all Unix-like operating systems, including vulnerabilities in the password authentication system, file system, virtual memory system, and applications that commonly run on Linux and Unix.

The course will teach you the skills to use freely available tools to handle security issues, including SSH, AIDE, sudo, lsof, and many others. SANS’s practical approach uses hands-on exercises every day to ensure that you will be able to use these tools as soon as you return to work. We will also put these tools to work in a special section that covers simple forensic techniques for investigating compromised systems.

Topics

- Memory Attacks, Buffer Overflows
- File System Attacks, Race Conditions
- Trojan Horse Programs and Rootkits
- Monitoring and Alerting Tools
- Unix Logging and Kernel-Level Auditing
- Building a Centralized Logging Infrastructure
- Network Security Tools
- SSH for Secure Administration
- Server Lockdown for Linux and Unix
- Controlling Root Access with sudo
- SELinux and chroot() for Application Security
- DNSSEC Deployment and Automation
- mod_security and Web Application Firewalls
- Secure Configuration of BIND, Sendmail, and Apache
- Forensic Investigation of Linux Systems

Course Author Statement

“A wise man once said, ‘How are you going to learn anything if you know everything already?’ And yet there seems to be a quiet arrogance in the Unix community that we have figured out all of our security problems, as if to say, ‘Been there, done that.’ All I can say is that what keeps me going in the Unix field, and the security industry in particular, is that there is always something new to learn, discover, or invent. In 20 plus years on the job, what I have learned is how much more there is that I can learn. I think this is also true for the students in my courses. I regularly get comments back from students who say things like, ‘I have been using Unix for 20 years, and I still learned a lot in this class.’ That is really rewarding.”
— Hal Pomeranz

Available Training Formats

Live Training

Live Events
sans.org/information-security-training/by-location/all

Summit Events
sans.org/cyber-security-summit

Private Training
sans.org/private-training

Online Training

OnDemand
sans.org/ondemand

Simulcast
sans.org/simulcast

Section Descriptions

SECTION 1: Hardening Linux/Unix Systems – Part 1

This course section tackles some of the most important techniques for protecting your Linux/Unix systems from external attacks, and it also covers what those attacks are so that you know what you're defending against. This is a full-disclosure course with in-class demos of actual exploits and hands-on exercises to experiment with various examples of malicious software, as well as different techniques for protecting Linux/Unix systems.

TOPICS: Memory Attacks and Overflows; Vulnerability Minimization; Boot-Time Configuration; Encrypted Access; Host-Based Firewalls

SECTION 2: Hardening Linux/Unix Systems – Part 2

Continuing our exploration of Linux/Unix security issues, this course section focuses on local exploits and access control issues. What do attackers do once they gain access to your systems? How can you detect their presence? How do you protect against attackers with physical access to your systems? What can you do to protect against mistakes (or malicious activity) by your own users?

TOPICS: Rootkits and Malicious Software; File Integrity Assessment; Physical Attacks and Defenses; User Access Controls; Root Access Control with sudo; Warning Banners; Kernel Tuning for Security

SECTION 3: Hardening Linux/Unix Systems – Part 3

Monitoring your systems is critical for maintaining a secure environment. This course section digs into the different logging and monitoring tools available in Linux/Unix, and looks at additional tools for creating a centralized monitoring infrastructure such as Syslog-NG. Along the way, the course introduces a number of useful SSH tips and tricks for automating tasks and tunneling different network protocols in a secure fashion.

TOPICS: Automating Tasks With SSH; AIDE via SSH; Linux/Unix Logging Overview; SSH Tunneling; Centralized Logging with Syslog-NG

SECTION 4: Application Security – Part 1

This course section examines common application security tools and techniques. The SCP-Only Shell will be presented as an example of using an application under chroot() restriction, and as a more secure alternative to file-sharing protocols like anonymous FTP. The SELinux application whitelisting mechanism will be examined in-depth. Tips for troubleshooting common SELinux problems will be covered and students will learn how to craft new SELinux policies from scratch for new and locally developed applications. Significant hands-on time will be provided for students to practice these concepts.

TOPICS: chroot() for Application Security; The SCP-Only Shell; SELinux Basics; SELinux and the Reference Policy

SECTION 5: Application Security – Part 2

This course section is a full day of in-depth analysis on how to manage some of the most popular application-level services securely on a Linux/Unix platform. We will tackle the practical issues involved with securing three of the most commonly used Internet servers on Linux and Unix: BIND, Sendmail, and Apache. Beyond basic security configuration information, we will take an in-depth look at topics like DNSSec and Web Application Firewalls with mod_security and the Core Rules.

TOPICS: BIND; DNSSec; Apache; Web Application Firewalls with mod_security

SECTION 6: Digital Forensics for Linux/Unix

This hands-on course section is designed to be an information-rich introduction to basic forensic principles and techniques for investigating compromised Linux and Unix systems. At a high level, it introduces the critical forensic concepts and tools that every administrator should know and provides a real-world compromise for students to investigate using the tools and strategies discussed in class.

TOPICS: Tools Throughout; Forensic Preparation and Best Practices; Incident Response and Evidence Acquisition; Media Analysis; Incident Reporting

Who Should Attend

- Security professionals looking to learn the basics of securing Unix operating systems
- Experienced administrators looking for in-depth descriptions of attacks on Unix systems and how they can be prevented
- Administrators needing information on how to secure common Internet applications on the Unix platform
- Auditors, incident responders, and InfoSec analysts who need greater insight into Linux and Unix security tools, procedures, and best practices

> **"This course gave me a better understanding of Linux internals and specific threat hunting ideas that I will use in my environment."**
>
> — Shelby Peterson, **Adobe**