A Platform-Independent Approach to Securing Enterprise Hosts

EUGLUG - Jason Chan

November 1, 2003

Where Security & Business Intersect℠

Agenda

- Introduction
- Background
- Details
- Questions

Introduction

- **@stake - Digital security consultancy based in Cambridge**
  - Offices in NYC, RTP, Chicago, San Francisco, Seattle, London
  - Conduct independent security research
  - Also produce several security products - LC4 and WebProxy
  - Work with world's top 6/10 banks, 4/10 software companies, 7/10 telcos
- **Me - Jason Chan**
  - With @stake for 3 years
  - Previously worked with US Navy Space and Naval Warfare Engineering Center - Information Warfare

Background

Why secure hosts?

- Web page defacement = Bad
- Denial of Service (DoS) = Worse
- Loss of customer or corporate data = Yikes!!
- Being fired for any of the above = @#$*!

Why an OS independent methodology?

- **Most organizations use multiple operating systems**
- **To ensure consistent security standards are applied throughout enterprise**
- **Many tools are available, but few are cross-platform**
  - Bastille Linux, SST/JASS, YASSP, MS Security Templates
- **Engineers are technology oriented**
  - Methodologies are your friend

What's needed for success?

- **Security Policy**
  - Not necessarily formalized (though it is preferred)
  - A statement of the guidelines, restrictions, and enforcement actions associated with the appropriate use of organizational information assets
  - Technological security controls are merely implements of security policy
- **Documentation**
  - Modifications from default should be recorded
- **Centralization**
- **Automation**
  - Host build
  - Monitoring
  - Periodic assessment

Details

Host Security Methodology

- **Goal - To provide security level appropriate for the organization and the exposure and criticality of system**
- **Component of 'Defense in Depth'**
- **Will provide appropriate framework in any stage of host security lifecycle**
  - Initial build
  - Post-build hardening or build review
  - Ongoing assessment
- **12 high-level areas**
  - Can be considered 'to-do's'
  - Individual requirements may require more, less, or different areas

Physical Security

- No production servers in cubicles!
- EEPROM, BIOS passwords, LILO and GRUB passwords
- Tamper-evident cases
- Locking racks
- Co-lo and hosting environments

Patches

- **The most important step in securing a system**
- **Estimated that 90% of attacks exploit known vulnerabilities**
- **Running latest patches protects you from all PUBLIC vulnerabilities**
- **Sample attacks:**
  - Worms - SQL Slammer, Code Red, Ramen
- **This is an area the industry is racing to catch up in**
  - Automation is key
  - RedHat Network, Windows Update
  - SMS, Tivoli, ZenWorks

Network Services

- Like minimizing the OS, ‘run only what you need’
- Avoid unsafe services (i.e. r-services, TFTP)
- Use encrypted alternatives when possible and appropriate (i.e. SSH, HTTPS)
- Implement access controls (i.e. TCP Wrappers, IPSec policies) on non-public services
- Bind network services to specific interfaces
  - i.e Listen, ListenAddress on Apache, OpenSSH

User Accounts and Account Policy

- **Ensure only authorized users have access to systems**
- **Ensure password and system access policy are configured securely**
- **Remove unused and legacy user accounts and disable guest access**
- **Restrict direct administrative access**
  - /etc/default/login, /etc/securetty, PermitRootLogin
  - su, sudo, runas, RBAC

User Accounts and Account Policy (Cont.)

- **Configure password policies**
  - Expiry, reuse, history, complexity, lockout, etc.
  - /etc/login.defs, /etc/default files, Local Security Policy
- **Configure login and user rights auditing**
  - Event log, sulog, loginlog, inetd, network services
- **Sample attacks**
  - Password brute forcing
  - Blank passwords

Application Configuration

- **Implementing best security practices when installing and configuring applications - different than 'secure coding'**
- **Install and run applications as specific, non-administrative users - watch for users installed by default**
- **Beware of service accounts**
- **chroot()**
- **'Security by Obscurity' can help here**
  - Version hiding and non-default ports
  - Don't put IIS web content in C:\inetpub\wwwroot!!!

System Management

- **Use encrypted and/or out-of-band means for managing systems**
  - ssh, scp, sftp, not Telnet, rsh, rcp, ftp
- **Use ACLs and other restrictions to control who can login, and from where**
- **Sample attacks**
  - Password sniffing (i.e. dsniff, ettercap, standard sniffing tools)
  - .rhosts exploitation

Operating System Minimization

- **Reduce feature set and software available on system**
- **Improves security, performance, and manageability**
- **Includes kernel tweaking (i.e. enabling only required subsystems and components)**
- **Use tools like pkginfo and rpm to find installed packages**
- **This issue is best addressed at installation:**
  - Anaconda/KickStart, JumpStart, Ignite-UX, Windows answer files, Ghost
- **Sample attacks**
  - Windows Media Player
  - Solaris IPv6 Multicast tunneling

Network Stack Hardening

- **Configuring TCP/IP to prevent information disclosure, DoS, and other attacks**
- **Includes ICMP, TCP, UDP, and other controls**
- **Sample attacks:**
  - Smurf - ICMP echo broadcasts from spoofed addresses
  - Session hijacking as a result of predictable ISNs
  - User-installed network daemons on unprivileged ports
- **Solaris - ndd, Linux, BSDs - sysctl, Windows - Registry**

Filesystem Security

- **Designing and implementing filesystems to resist attack**
- **Choosing the right filesystem and features (i.e. journaling filesystem, NTFS, RAID)**
- **Creating partitioning schemes**
- **Applying filesystem security controls**
  - Mounting home directories, /tmp, /var nosuid
  - Mounting /bin read only

File Permissions

- **Implement 'least privilege' for file permissions**
- **Reduce and document setuid, setgid, world writable files**
- **Tighten file permissions of root-executed binaries**
- **Eliminate default shares and default permissions on shares**
- **Developer and administrator workstations are huge targets**
  - Personal encryption tools
- **Sample attacks**
  - in.rexecd
  - Incorrectly configured umask

Integrity Checking and System Verification

- **Ensuring system files have not been inappropriately modified (maliciously or accidentally)**
- **Includes deployed systems and software repositories (i.e. build servers, CVS)**
- **Tools like Tripwire, hashing utilities (i.e. l5, md5sum), PGP, sfdb, chrootkit**
- **Verifiable backups**
- **Sample attacks**
  - Binary replacement
  - Rootkits

Logging and Monitoring

- Proactively detecting security issues and other system anomalies
- Multi-level log filtering, correlation, reduction, and retention
- Improves security (incident detection, forensic readiness), troubleshooting, and availability
- Syslog, Event Logs, network monitoring tools

Extras

Host-Based Security Applications

- **Need for these applications depends on system use and exposure**
- **Anti-virus**
  - Crucial for desktop systems and mail servers
- **Firewall**
  - Becoming very popular on both servers and desktops
  - ipfilter, ipchains, iptables, Black Ice, Zone Alarm, ICF
  - Management can be an issue
- **HIDS**
  - From TCP Wrappers to Tripwire to other active host-based agents

Trusted Systems and Security Add-ons

- For highly-sensitive systems and data
- grsecurity (RBAC, TPE, Netfilter)
- Argus Pitbull
- Trusted Solaris, Trusted HP-UX/VirtualVault
- Various compliance levels (i.e. NCSC C2)
- These solutions generally create additionally administrative and performance overhead

Questions?