BCS ISSG Linux Day
Securing Linux

Peter SJF Bance CEng MBCS
Rhye Internet Solutions Limited
http://www.rhyeinternet.com/

Overview

- Why harden? Default installations.
- How hard to be? Cost v benefit.
- First steps – audit and plan.
- Hardening Process:
  - Removing unnecessary services
  - Users and permissions
  - Patching and configuration
- Advanced hardening – introduction
- Next steps – an ongoing process

Why Harden?

- With few exceptions, default Operating System installations expose unnecessary and dangerous services
- For example, in a typical RedHat 7.2 default installation (even without GUI components):
  - ~16 open TCP ports, mostly unnecessary
  - ~11 warnings produced by a basic Nessus scan
  - ~2 serious holes identified by Nessus
- Without a hardening process being followed, data is at serious risk
- Vendors are beginning to assist in the hardening process, and so the manual effort involved is reducing
- Simply adding a firewall is almost never sufficient

How hard to be?

- The basic steps (removing unnecessary services and patching the rest) should always be followed
- How much further you go depends on:
  - The machine's purpose
  - The user base (security v. convenience)
  - The availability and capability of maintenance personnel (or you!)

Audit and Plan

1. Identify exactly what you have on your system, in as much detail as possible
2. Identify your aims (see the previous slide)
3. Plan the process:
   - What services can be removed without impacting functionality (what is the machine's purpose)?
   - How should the remaining services be configured?
   - What user accounts are required on the system?
   - What permissions should each user be granted?
   - Are there any additional known threats to your system that need countermeasures? E.g. exposure to curious minds (University?!)

Hardening Process – Services

- As much hardening as possible should be performed **before** connecting to a network or the Internet
- Based on your assessment of the machine's purpose, remove unnecessary services and other software
- On the RH 7.2 system mentioned earlier, intended as a simple static Web server:
  - 64 packages removed using RPM (SMB, NFS, DHCP, etc.)
  - 2 packages added (Apache and SSH) also using RPM
- Configure necessary services as securely as possible:
  - Apply patches (inevitably necessary) – with RH, vendor-provided packages are updated reasonably rapidly
  - Apply sensible configuration (e.g. disable SSH1 capability, reduce Apache header information, disable SMTP relay, etc.)

Hardening Process – Users

- Without users, security would be simple, but unfortunately systems would be mostly useless ☹
- Only create necessary user accounts, and remove any unnecessary default ones
- Ensure all passwords are strong, either by issuing them yourself, or through regular audit
- Produce and publish a system security policy if relevant to your machine's purpose
- Monitor activity as much as possible – unusual hours of access, exceptional disk usage, etc.
- Be the BOFH! System security is most commonly put at risk by unaware users – consider implementing a security awareness training programme.

Hardening Process – Permissions

- Identify the rights and file-based permissions your users should have
- Identify the permissions your applications should have
- Even identify the permissions other systems should have when interacting with yours
- If unsure, it is far better to grant lower privileges than necessary and then increase them later than to have a system full of super-users from the outset
- Checking and altering file permissions system-wide can be onerous – consider implementing a chroot environment (jail)
- Never grant su or sudo privileges if you can avoid it

Other Risks

- Security is not just about Confidentiality and Integrity of information – Availability can also be crucial. Disaster Recovery and Business Continuity levels and plans should be defined and implemented.
- Security Awareness training becomes more important if users are easily susceptible to social engineering attacks.
- Dial-up remote control connections, FTP services, administration via Telnet, etc. – all should be carefully considered and alternatives found where possible.

Advanced Hardening

- Bear in mind Benefit v. Cost
- TCP Wrappers (e.g. hosts.deny/allow):
  - A basic firewall
  - Allow simple filtering of requests (e.g. by host)
- Application-level filtering:
  - E.g. Apache mod_rewrite
  - Proxy services
- Kernel modifications – e.g. IP stack tuning to reject particular requests
- Advanced authentication techniques (tokens, single-use passwords, etc.)
- Fake headers or identification of running services
- Tripwires, traps, WORM drives, Intrusion Detection, Firewalls, Bastions, Gateways, Proxies, Honey Pots, Counter attacks, Defense in Depth, Port Sentries, how far do you want to go?!
- Firewalling (IP Tables/Chains) and Intrusion Detection will be covered in other presentations
- Whatever you do beyond the basic steps, document everything, even if only for your own later reference!

Next Steps

- Security is not a state, it is a process. It is everyone's responsibility, and is a state of mind. Most of all, it is a balance – a difficult balance between Risk, Benefit and Cost.
- Independent Audit or Penetration Test, BS7799 Accreditation – may all be worthwhile depending on your environment.
- Policies are as important as technical security - ensure your users know how you expect them to use the system, and how to protect it and themselves from malicious outsiders.
- Keep up-to-date – today's vulnerability-free system is tomorrow's sieve
- Change control – ensure you know what is being put on your system (what could your users do with PHP? Perl?)

Summary

Slides available later today from:

http://www.rhyeinternet.com/papers/issg-linux/

Be secure – within reason!