

Openly Secure

Staying secure by using Open Source

Mohamed Hussein Sayed

mechanic@unixgarage.com

[http://www.unixgarage.com/
openlysecure.pdf](http://www.unixgarage.com/openlysecure.pdf)

Using OSS where security matters

- Some corporations and government agencies forbid it.
- The source is available!: Easier for bad guys to find holes.
- The source is available!: Good guys can find holes and fix them.
- The source is available! You can fix it yourself or pay someone to fix it.

The OS: Choosing a secure foundation

- The OS controls everything; if it is compromised the game is over.
- Linux kernel size and pace makes it harder to audit.
- SELinux was developed to cover the gaps and provide a more secure linux platform.
- SELinux main features: Mandatory access controls, Security policies, implemented as LSM.

The OS: Choosing a secure foundation

- SELinux is now enabled in many mainstream distributions.
- SELinux core idea is to protect the system in case of a partial compromise.
- Many sysadmins are frustrated by the configuration and turn it off.

The OS: Choosing a secure foundation.

OpenSolaris

- Made serious efforts towards improving security infrastructure at the OS level.
- Common Criteria certified EAL4+.
- Role based access control.
- Supports many popular firewalls.(PF, IPFilter).
- Policy management and roles definition requires some practice.
- Supports Pluggable Authentication Modules , Kerberos, LDAP.

The OS: Choosing a secure foundation

OpenBSD

- Designed from the ground up with security in mind.
- Best security track record among OS OS's.
- Gave birth to OpenSSH, the defacto secure shell.
- Gave birth to PF, solid and robust firewall engine.
- Great IPv6 and IPSec support.

The OS: Choosing a secure foundation.

OpenBSD

- May be late to implement features, especially if it affects security.
- May not support the latest and greatest hardware.
- Excellent choice for network services type work including Firewalls, VPNs, DNS , FTP, Mail and Web servers.

The applications: Email

- Crucial service in today's world.
- Insecure email servers have been notoriously targeted in the past.
- Must be closely monitored.
- The grandfather of email servers had several major security issues.
- Newer alternatives like postfix had security as a core requirement.

The application: FTP servers

- Older ftp servers were commonly targeted.
- Vulnerabilities usually required only a normal user which may be easily sniffed or socially engineered.
- Newer ftp servers implement chroots, allow for stricter policies.
- VSFTPD puts a great effort into providing the service securely

The application: Web Servers

- Crucial service for modern business.
- Industry standard has been apache.
- Apache has a great track record, but as the project adds features this record may come to test.
- Apache has support for various authentication backends.
- Alternatives with less features are available and may be easily audited.

Loose the fat: Less is more

- Many main stream distributions turn on services by default.
- It is important to disable any additional services you may not use.
- Linux distributions can have a laxer service startup policy.
- OpenBSD service management is easy via /etc/rc.conf

Secure the perimeter: Firewalls

- OS firewall engines are plenty and excellent in features.
- IPTables has very large feature set, excellent modules repository and great library for developers.
- PF has a very clean syntax, excellent feature set. Excellent performance and have been ported to many OS's

Secure the perimeter: Firewalls

Successful implementation

- Decide on a posture.
- Be Paranoid.
- Understand the business requirements.
- Create a policy.
- Review with your peers, ask the experts for tips without leaking information.
- Implement.
- Monitor.
- Adjust.

Watch the action: Intrusion detection

- Goal: Detect failed and successful attempts to compromise your digital assets.
- Different types exist: Host intrusion detection, network intrusion detection.
- HIDS aim to detect attacks at the machine level. It leverages filesystem monitoring, process monitoring, etc. Example: Tripwire.
- NIDS aim to detect attacks by monitoring and compare against known attacks signature.

Analyze the logs

- Unexamined data is worthless.
- Tune your environment to record what you care about.
- Centralize your logs.
- Make it a habit to surf your logs.
- Create reports and graphs. A picture is worth a 1000 words.

References

- www.openbsd.org/security.html
- opensolaris.org/os/community/security/
- www.nsa.gov/research/selinux/
- www.snort.org
- www.tripwire.org
- www.wireshark.org
- en.wikipedia.org/wiki/Syslog-ng
- www.apache.org
- www.packetstormsecurity.org
- www.cert.org
- <http://www.openbsd.org/cgi-bin/man.cgi?query=pf.conf&apropos=0&sektion=0&manpath=OpenBSD+Current&arch=i386&format=html>