

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 sendmail 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

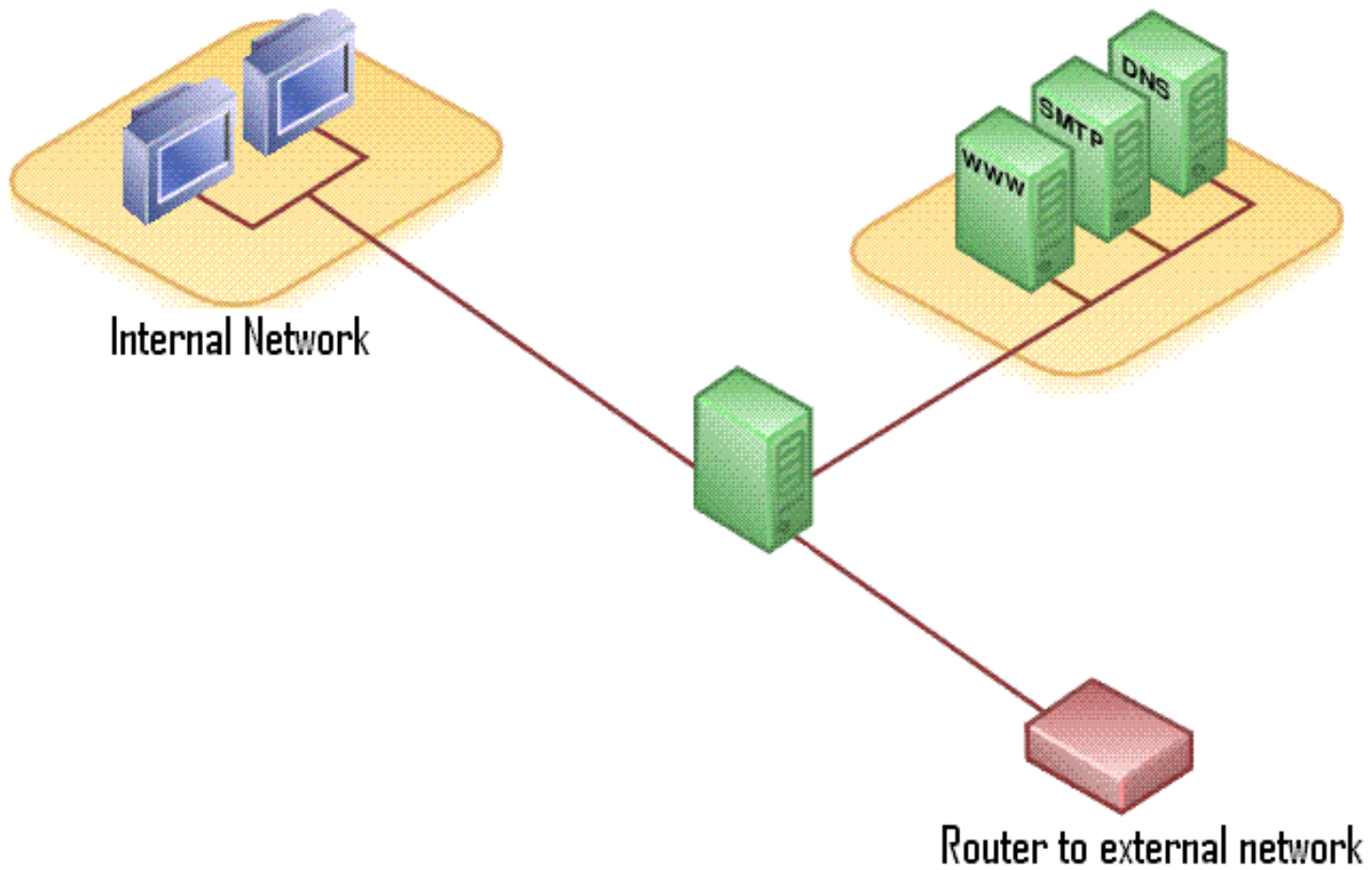
The application: Web Servers

- 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 better

- 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



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.

Sample PF ruleset

```
# macros
ext_if="fxp0"
int_if="xl0"
tcp_services="{ 22, 113 }"
icmp_types="echoreq"
comp3="192.168.0.3"
# optionsset
block-policy return
set loginterface $ext_if
set skip on lo
# scrub
scrub in
# nat/rdr
nat on $ext_if from !($ext_if) -> ($ext_if:0)
nat-anchor "ftp-proxy/*"
rdr-anchor "ftp-proxy/*"
rdr pass on $int_if proto tcp to port ftp -> 127.0.0.1 port 8021
rdr on $ext_if proto tcp from any to any port 80 -> $comp3
```

Sample PF (CONT.)

```
# filter rules
block in
pass out keep state
anchor "ftp-proxy/*"
antispoof quick for { lo $int_if }
pass in on $ext_if inet proto tcp from any to ($ext_if) \ port $tcp_services flags S/SA
keep state
pass in on $ext_if inet proto tcp from any to $comp3 port 80 \ flags S/SA synproxy
state
pass in inet proto icmp all icmp-type $icmp_types keep state
pass in quick on $int_if
```

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.

OS IDS: Snort

- Snort is a mature and well supported IDS.
- Sniffs traffic off the network and compare against signatures(Rules)
- Plenty of tools to analyze events or integrate with other systems
- Configuration system is straightforward.
- Lately adopted a subscription model.

OS IDS: Snort

```
var HOME_NET any
var EXTERNAL_NET any
var DNS_SERVERS $HOME_NET
var SMTP_SERVERS $HOME_NET
var HTTP_SERVERS $HOME_NET
var SQL_SERVERS $HOME_NET
var TELNET_SERVERS $HOME_NET
var SNMP_SERVERS $HOME_NET
portvar SSH_PORTS 22
portvar HTTP_PORTS 80
portvar SHELLCODE_PORTS !80
portvar ORACLE_PORTS 1521
var RULE_PATH /etc/snort/rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules
dynamicpreprocessor directory /usr/local/lib/snort_dynamicpreprocessor/
dynamicengine /usr/local/lib/snort_dynamicengine/libsf_engine.so
preprocessor frag3_global: max_fragments 65536
preprocessor frag3_engine: policy first detect_anomalies
```

OS IDS: BASE

Basic Analysis and Security Engine (BASE) - Mozilla

File Edit View Go Bookmarks Tools Window Help

Basic Analysis and Security Engine (BASE)

- Most recent Alerts: **any protocol, TCP, UDP, ICMP**
- Today's: alerts **unique, listing**; IP **src / dst**
- Last 24 Hours: alerts **unique, listing**; IP **src / dst**
- Last 72 Hours: alerts **unique, listing**; IP **src / dst**
- Most **recent 15 Unique Alerts**
- Last Source Ports: **any, TCP, UDP**
- Last Destination Ports: **any, TCP, UDP**
- Most **frequent 5 Alerts**
- Most Frequent Source Ports: **any, TCP, UDP**
- Most Frequent Destination Ports: **any, TCP, UDP**
- Most frequent 15 addresses: **source, destination**

Added 0 alert(s) to the Alert cache
Queried on : Thu October 14, 2004 22:02:36
Database: snort_log@localhost (schema version: 106)
Time window: [2004-09-02 16:05:49] - [2004-10-08 11:25:41]

[Search](#)
[Graph Alert data](#)
Graph alert **detection time**

Sensors: 1 Unique Alerts: 14 categories:5 Total Number of Alerts: 84 <ul style="list-style-type: none">• Src IP addrs: 5• Dest. IP addrs: 9• Unique IP links 13 • Source Ports: 68<ul style="list-style-type: none">◦ TCP (68) UDP (0)• Dest. Ports: 12<ul style="list-style-type: none">◦ TCP (12) UDP (0)	Traffic Profile by Protocol TCP (96%) UDP (0%) ICMP (4%) Portscan Traffic (0%)
---	---

[Alert Group Maintenance](#) | [Cache & Status](#) | [Administration](#)

BASE 0.9.7.2 (by **Kevin Johnson** and the BASE Project Team
Built on ACID by Roman Danyliw)

[Loaded in 0 seconds]

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
- www.sans.org