

# Using Free Software to Secure Your Network

LinuxBiz @ SIExpo 2004

17:30 – 18:15

Tycho Fruru

([tycho.fruru@conostix.com](mailto:tycho.fruru@conostix.com))

# Contents

- **Introduction**
- Why use FOSS ?
- Some security products based on FOSS

# Introduction

- This presentation is about using FOSS for security
  - Why should you (not) use FOSS ?
  - What products/projects exist ?
    - very quick overview, not exhaustive
- But first : what *is* “FOSS” ?

# Free Software

## Open Source Software

- Both FS and OSS use existing copyright laws to give you some rights
  - The right to run and use, for any reason
  - The right to distribute
  - The right to inspect/learn
  - The right to modify/adapt/improve
  - The right to distribute modified versions
- Details vary between different licences, but for *users* (as opposed to *developers* or *distributors*), these changes do not matter much.

# FS + OSS = F(L)OSS

- FS                      Free Software
- +
- OSS                    Open Source Software
- =
- F(L)OSS              Free (Libre) and  
Open Source Software

# Free Software Misconceptions

- Free Software
  - is not freeware
  - is not public domain software
  - has nothing to do with monetary value
    - Free Speech vs Free Beer

# FOSS

- More info on Free Software

<http://www.gnu.org/>

- More info on Open Source Software

<http://www.opensource.org/>

# Technical roots of FOSS

- FOSS – “scratch that itch”
  - Technical solutions for technical problems
  - Network and Security projects are popular FOSS projects

# FOSS in the IT world

- FOSS use in business is increasing
  - First often as “hidden” (technical) solution
  - Now often as (strategic) business decision
- FOSS is not “hobbyist” / “amateur” anymore
  - Some big companies are actively supporting and contributing
  - ... but not because they are “just nice guys” ...

# FOSS and Big Players

- IBM, Sun, HP, Oracle, Novell, ...
  - no altruism here ...
  - FOSS is a means to
    - Limit their dependency on other OS vendors
    - Hedge against failure/obsolescence of their own products
    - Enter into new markets (eg. smaller UNIX-like systems)
    - Promote professional services

# Contents

- Introduction
- **Why use FOSS ?**
- Some security products based on FOSS

# Why use FOSS : Some false reasons

- “Because it's free (cost)”
  - There's no such thing as a free lunch ...
- “Because it's free (freedom) !”
  - Very noble ! So what ?
- “Because it's more secure !”
  - Is it, really ?
- “Because I like it ! It's cool !”

# Why NOT use FOSS :

## Some false reasons

- “It's not supported”
- “It's just some hobby project, done by amateurs in their spare time”

Some more hilarious ones :

- “It will 'infect' internal development”
- “It destroys IP ('Intellectual Property')”
- “It's bad for the Software Development Ecosystem”

# “Because it's free (cost)”

- It's not free ...
  - Hardware
  - Installation
  - Training
  - Support
  - Software Licencing (also possible for FOSS)

# “Because it's free (freedom)”

- True, but this doesn't *directly* impact *users* of the system.
  - It might be proprietary and it would still function in exactly the same way.

# “Because it's more secure”

- Source code is available
  - Bad programming becomes visible
    - “Darwinian selection”
- “Given enough eyes, all bugs are shallow” - Linus's law
- Everyone can look, learn and fix
  - even if the vendor/original author doesn't want to or is not reactive

# Is FOSS less secure, then ?

- Source code is available
  - ... and is ignored by most users
  - ... but gives more information to crackers
- “Given enough eyes, all bugs are shallow”
  - ... but who will find them first ?
- Everyone can fix it
  - Not all projects have good change management / problem resolution procedures

# Conclusion :

## FOSS and security are orthogonal

- There are “secure” and “insecure” FOSS projects, as there are “secure” and “insecure” proprietary projects.
  - The trick is to know which ones are “secure” ...
- Make an informed choice !
  - ... as with everything, really ...

## But it's not supported !

- Not true, commercial support is available through a multitude of channels ...
  - Good quality support will go even further than just inform the original author about potential problems - active contribution to the community

# But it's just some hackers in their basement !

- Some projects are indeed like that, while others aren't.
  - Decide on the merits of each project. Being a hacker in a basement does not preclude writing a solid security tool and having solid security procedures

# Why use FOSS :

## Some better reasons

- “Because it solves my (security) problem in a cost-effective way”
  - Compare TCOs (in an unbiased way)
- Because it simplifies security management
- “Because it decreases our dependency on external proprietary software vendors”
- “Because I will be able to get better, more thorough support”
  - And switch support vendor if needed ...

# Compare TCO

- Initial (One-off) costs
  - Everything needed to get it working
- Recurring costs
  - Everything needed to keep it working
- Risk

# TCO components

## ● One-off costs

- Software licence
- Hardware
- Initial installation & Configuration
- Initial training

## ● Recurring costs

- Software & configuration evolution
- Support & Maintenance
- On-going training

# TCO components (cont.)

## ● Risk

- Software stability and security
- Flexibility (covering future needs)
- Openness, Integration with other products
- (Financial) stability of the software vendor
- Support risk
  - International support organisation troubles
  - Business reorientation
  - Forced software upgrades
  - Forced hardware upgrades
  - Up to what point can your support REALLY help you ?

# TCO Conclusion

- It's very difficult to calculate “complete” TCO
  - Depends as much on external factors (ex. licence price) as on internal ones (ex. available skills)
- Canned results are “per definition” wrong.

# Managing (FOSS) Security

- Lots of discussion points for security information
  - p.ex. vuln-dev, bugtraq, ...
    - Too much information
    - Low signal/noise ratio
- Every security-aware FOSS project has its own incident and vulnerability handling procedures, but
  - It's difficult to find the right information**

# Managing (FOSS) Security

- Centralised vulnerability / fix databases

Open Source Vulnerability Database

<http://www.osvdb.org/>

Secunia Advisories

<http://www.secunia.com/>

# Managing (FOSS) Security

- FOSS Distributions (eg. Debian, Gentoo, Red Hat) track security information and make updates readily available

These update mechanisms do work.

Choose your vendor based on the quality of these updates. (RedHat, SuSE, Debian, Gentoo, ...)

- Simplified security management

# Openness

- For FOSS, openness is an advantage
  - mix-and-match best building blocks to make a solution
  - forces FOSS solutions to compete on real merits because vendor lock-in is reduced
- For proprietary software, openness is a necessary evil
  - because it can decrease customer fidelity

# Vendor Support Lock-In

- FOSS :

- Switching first-line support is possible
- Switching support vendor is possible

- Proprietary :

- Switching first-line support is possible
- Switching support vendor is impossible
  - He's the only one who can really change how the software works and correct bugs !

# Contents

- Introduction
- Why use FOSS ?
- **Some security products based on FOSS**

# FOSS Security Products

- Large range of security-aware products
  - “Security applications”
    - Security is their main goal
  - “Business applications”
    - Security is not their main goal, but is nonetheless a very important component

# FOSS Security Applications

- Firewall / VPN
- Intrusion Detection Systems
- Reverse Proxy
- Secure Login, Authentication Systems
- Anti-Virus

# FOSS Business Applications

- Web/Database services
  - LAMP (Linux/Apache/Mysql/PHP)
  - PostgreSQL
  - CMS systems (Zope, ...)
- File services
  - Samba
- Mail Services
  - Postfix (with amavis, clamav and spamassassin)
- And many more ...

# But which application to use ?

- A good (FOSS) security application should
  - ... meet the security and business needs of the organisation
  - ... be well supported (through support vendors and/or by the original authors)
  - ... have solid security incident handling procedures and track record
    - eg. number of days between discovery and resolution of vulnerabilities

# How to choose a solution ?

- Talk to people who have the experience.
- Talk with companies that can implement both FOSS and proprietary solutions and have them defend their proposal.

# Top reason to use FOSS

- Because often
  - it just makes business sense !
  - if it solves your problem
  - with less lock-in
  - and a free market (competition) for support
  - and possibility to audit and verify independently
- FOSS enables you to re-take control of your IT and security infrastructure.

Questions ?

Thank you !

*tycho.fruru@conostix.com*