Open Source & Security

Linux Meeting 2002
Maggio - Bologna

http://www.infosec.it **info@infosec.it**

Relatore: Igor Falcomatà
igor@infosec.it
http://www.infosec.it/progetti.html

© 2002 Infosec srl – Tutti i diritti riservati.

The BIG dilemma..

- **è più sicuro il software Open Source?**
- **è più sicuro il software closed source/proprietario?**

in queste slide NON troverete una risposta....

...perché (imho) non esiste una risposta

© 2002 Infosec srl – Tutti i diritti riservati.

INFOSEC

Full disclosure

- una filosofia per affrontare le tematiche di sicurezza (informatica) che recita:
  - un sistema realmente sicuro deve essere in grado di superare una revisione pubblica a tutti i livelli (protocollo, codice sorgente, etc.)
  - i dettagli delle vulnerabilità di sicurezza devono essere accessibili a chiunque
- adottata da quelli che vengono definiti (o si definiscono) "white-hats"

http://www.securityfocus.com/popups/forums/bugtraq/faq.shtml

- vantaggi
  - un grande numero di individui può valutare le debolezze di sicurezza del sistema
  - i produttori sono stimolati a fornire le soluzioni in tempo breve
  - programmatori e progettisti possono imparare dagli errori di altri
  - gli utenti possono identificare vulnerabilità simili su sistemi diversi dall'originale
- svantaggi
  - nello stesso momento in cui informi le persone "costruttive" delle vulnerabilità di sicurezza, informi anche i malintenzionati

© 2002 Infosec srl – Tutti i diritti riservati.

Modello "Open Source" - vantaggi

- review, debugging e analisi
  - "esperti"
  - community
  - utilizzatore finale
  - hacker e "cappelli colorati" di vario tipo

Attraverso l'analisi (del codice, del protocollo, dell'algoritmo, del progetto, ...) è possibile valutare la bontà della soluzione (verifica, auditing, controllo qualità, ...).

Problemi:

- **ha senso/è possibile fare l'analisi?**
  (progetti "enormi"; flawed by design; di scarso interesse; aggiornati continuamente; ...)
- **qualcuno ha fatto l'analisi?**
- **è stata resa pubblica?**
- **l'analisi è corretta e/o sufficiente; sono state trovate tutte le problematiche?**
- **l'analisi viene "mantenuta"?**

© 2002 Infosec srl – Tutti i diritti riservati.

Modello "Open Source" - vantaggi

- review, debugging e analisi
  - "esperti"
  - community
  - utilizzatore finale
  - hacker e "cappelli colorati" di vario tipo
- fix, modifiche, supporto, personalizzazione,
  - tempestività
  - indipendenza dal produttore
  - contributi della community
  - utenti "avanzati" e "hacker/smanettoni" con conoscenza approfondita

Non si è necessariamente legati al produttore per gli aggiornamenti (anche di sicurezza), l'ampliamento delle funzionalità, il supporto, l'integrazione con il proprio sistema informativo e la personalizzazione.

Problemi:

- **chi verifica tutte queste cose?**
- **vengono poi integrate nel progetto originale?**
- **frammentazione**

© 2002 Infosec srl – Tutti i diritti riservati.

INFOSEC

Modello "Open Source" - vantaggi

- review, debugging e analisi
  - "esperti"
  - community
  - utilizzatore finale
  - hacker e "cappelli colorati" di vario tipo
- fix, modifiche, supporto, personalizzazione,
  - tempestività
  - indipendenza dal produttore
  - contributi della community
  - utenti "avanzati" e "hacker/smanettoni" con conoscenza approfondita
- standard e compatibilità

A parte qualche vendor che ama reinventare l'acqua calda ogni volta, molti standard sono ormai "aperti". L'analisi del protocollo e/o codice permette di riutilizzare parti di altri progetti per facilitare lo sviluppo, oppure di creare standard "de facto" (nuove applicazioni, standard non definiti, ...).

Problemi:

- **review del "primo" standard?**
- **bug comuni/ereditati**

© 2002 Infosec srl – Tutti i diritti riservati.

INFOSEC

Modello "Open Source" - svantaggi

- "classici"
  - TCO (costi di gestione/manutenzione)
  - mancanza di supporto commerciale
  - difficoltà di utilizzo
  - interfaccia e manutenzione non uniformi
  - frammentazione dei progetti
- "security related"
  - falso senso di sicurezza
  - maggiore facilità nel trovare le vulnerabilità
  - scarsa qualità (codice, progettazione, implementazione)
  - scarsa manutenzione

Il solo fatto che un progetto sia Open Source o comunque pubblico non garantisce assolutamente che sia sicuro o anche solo che qualcuno si sia preso la briga di analizzarlo.

Molti progetti sono enormi (vari milioni di righe di codice: kernel, Xfree86, KDE/Gnome, Mozilla, OpenOffice, etc.) .

Esistono numerosissimi progetti con codice in qualità "pre-alpha", "alpha", "beta"; sviluppati e mantenuti amatorialmente; non più mantenuti…

© 2002 Infosec srl – Tutti i diritti riservati.

INFOSEC

Modello "Open Source" - svantaggi

- "classici"
  - TCO (costi di gestione/manutenzione)
  - mancanza di supporto commerciale
  - difficoltà di utilizzo

Il solo fatto che un progetto sia Open Source o comunque pubblico non garantisce assolutamente che sia sicuro o anche solo che qualcuno

Solo nei progetti Open Source? :)

implementazione)
- scarsa manutenzione

"alpha", "beta"; sviluppati e mantenuti amatorialmente; non più mantenuti...

© 2002 Infosec srl – Tutti i diritti riservati.

INFOSEC

Modello "Closed Source" – vantaggi[?]

- sorgenti non pubblici
  è più difficile trovare le vulnerabilità

E' realmente un vantaggio? Le vulnerabilità esistono (e spesso vengono trovate) comunque…

E' davvero così difficile trovarle? Esistono strumenti che permettono comunque di cercare vulnerabilità anche in software e librerie disponibile solamente come "binari" (fuzzy tester, tracer e debugger, disassembler, protocol e/o binary bruteforcer, …).

Non vediamo il codice ed il design interno.. Chi garantisce che non siano scritti/progettati coi piedi? Chi garantisce che non ci siano "backdoor"?

© 2002 Infosec srl – Tutti i diritti riservati.

INFOSEC

Modello "Closed Source" – vantaggi[?]

- sorgenti non pubblici
  è più difficile trovare le vulnerabilità
- supporto ed affidabilità
  Il produttore "garantisce" l'affidabilità del progetto e del supporto, attraverso la sua professionalità ed i suoi partner selezionati...

"security through obscurity"

Avete mai letto le clausole della licenza di un qualsiasi software commerciale?

Possibile che i tanto professionali supporti tecnici dei produttori a volte impieghino mesi per trovare e risolvere le vulnerabilità? (posso capire verifica approfondita, regression test & co, ma spesso si esagera...).

Possibile che sia così difficile trovare informazioni tecniche <u>approfondite</u> sulla struttura interna e sul design di protocolli e software proprietari?

E se il produttore fallisce e/o elimina il prodotto?

© 2002 Infosec srl – Tutti i diritti riservati.

Le cose stanno cambiando...

- le cose stanno cambiando, grazie anche a risorse come bugtraq ed alla full disclosure
- i produttori hanno cominciato a considerare la sicurezza come parte importante/fondamentale della loro strategia
- vengono sempre più utilizzati standard aperti, almeno per le componenti critiche
  - protocolli
  - crittografia
  - autentificazione
- creano risorse specifiche per il supporto di security
  - staff specializzati, siti, mailing list, libri, corsi, certificazioni, ...
- utilizzano società specializzate
  - per valutare la sicurezza dei sistemi
  - per analizzare o fare "l'auditing" di protocolli e codice sorgente
  - per sviluppare parti del sistema sensibili ai fini della sicurezza
- adottano un approccio attivo
  - seguono e ascoltano anche risorse "non ufficiali"
  - rivedono il design, le impostazioni ed il codice dei loro prodotti in ottica di sicurezza

© 2002 Infosec srl – Tutti i diritti riservati.

INFOSEC

The right way: KISS & qualità

- il rasoio di Occam
  Keep It Simple, Stupid!
  - design semplice
  - modularità
  - razionalità
  - standard "aperti" e consolidati
  - concedere/utilizzare i privilegi minimi indispensabili
  - verificare tutto l'input e l'output
  - adottare metodologie di programmazione "sicura"
  - evitare funzioni "insicure"
  - documentare e commentare codice, progetto e protocolli
  - fare rivedere il codice da terze parti
  - supporto

Secure Programming for Linux and Unix HOWTO:
http://www.dwheeler.com/secure-programs/

Secure UNIX Programming FAQ
http://www.whitefang.com/sup/

Secure Programming Standards Methodology Manual
http://www.ideahamster.org/projects.html

The World Wide Web Security FAQ
http://web3.w3.org/Security/faq/

© 2002 Infosec srl – Tutti i diritti riservati.

INFOSEC

Domande?

<Pubblicità Progresso>

http://www.sikurezza.org/
Italian Security Mailing List

</Pubblicità Progresso>

© 2002 Infosec srl – Tutti i diritti riservati.

Alcuni riferimenti...

- **google/teoma: open +source +security**
- **http://www.dwheeler.com/oss_fs_why.html**
  Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers!, D. A. Wheeler
- **http://softwaredev.earthweb.com/sdopen/article/0,,12077_990711,00.html**
  The Myth of Open Source Security Revisited v2.0, D. Obasanjo
- **http://www.infosecuritymag.com/articles/march01/features1_open_source_sec.shtml**
  OPEN-SOURCE SECURITY - OPEN SOURCE UNDER THE HOOD, P. Loshin
- **http://www.sardonix.org**
  Promoting Open Source Security
- **http://rr.sans.org/code/open_source.php**
  Is Open Source Software Really More Secure?, B. Rogers
- **http://www.counterpane.com/real-world-security.html**
  Security in the Real World: How to Evaluate Security, B. Schneier
- **http://online.securityfocus.com/news/322**
  U.S. Funds Open Source Security Hub, K. Poulsen
- **http://www-4.ibm.com/software/developer/library/oss-security.html**
  Open source software: Will it make me secure?, J. Viega

© 2002 Infosec srl – Tutti i diritti riservati.

Open Source & Security

Linux Meeting 2002
Maggio - Bologna

http://www.infosec.it **info@infosec.it**

Relatore: Igor Falcomatà
igor@infosec.it
http://www.infosec.it/progetti.html

© 2002 Infosec srl – Tutti i diritti riservati.