



Open vs. Closed Source



Un confronto aperto tra i modelli
Open Source e Closed Source



L'esigenza della sicurezza

- Un tempo i sistemi “sicuri” erano pochi, oscuri, acceduti infrequentemente.
- Oggi, quasi tutti i sistemi contengono dati importanti, critici o semplicemente privati, e vanno protetti. Al contempo, devono essere sempre più aperti e interconnessi.
- “Virtual is real... open is secure”: oltre che un ottimo motto pubblicitario, una verità
- Esigiamo una “garanzia di sicurezza”

Cos'è la sicurezza?

- Definiamo accademicamente la sicurezza come il raggiungimento di 3 obiettivi:
 - **Confidenzialità**
 - **Integrità**
 - **Disponibilità**
- Accesso, in lettura o in scrittura, “if and only if” l'utente è autorizzato appropriatamente
- Tutto ciò è di una semplicità ingannevole

Molti punti di vulnerabilità...

- I sistemi informativi devono essere progettati per garantire la sicurezza, mediante un uso appropriato e ragionevole di tecniche di crittografia, autenticazione, auditing, controllo.
- I processi, devono essere pensati per la sicurezza: per esempio, l'assegnazione di login e password
- Le persone sono vulnerabili: possono essere convinte o costrette a utilizzare male i loro privilegi, possono essere risentite e volersi vendicare
- **Il software deve essere sicuro e robusto, comportarsi nel modo in cui è previsto che si comporti, sempre.**

Sicurezza del software?

Non esistono software sicuri a priori!

Ogni progetto software può contenere vari tipi di errori e problemi:

■ Interni

- Di tipo concettuale (errori di design)
- Di tipo programmatico

■ Esterni

- Causati da librerie e componenti riutilizzabili
- Causati dall'interazione con altri programmi e con il sistema operativo
- Causati da misconfigurazione

Software open source vs. Software proprietario

OSS:

- Codice sorgente disponibile a chiunque
- Programmato e concepito da appassionati
- Non sottoposto ad esigenze economiche e di marketing tipiche di un vendor

CSS:

- Nessuno ha accesso al codice sorgente a parte gli sviluppatori
- Sviluppato da programmatori di professione
- Sostenuto da capitali investiti da società interessate al ROI

Riflessi sulla sicurezza

OSS:

- Il sorgente può essere auditato da chiunque
- Patch, e sviluppi alternativi possono essere creati liberamente
- Collaborazione e riutilizzo di codice tra progetti
- Rapidissimo processo code/audit/exploit/patch

CSS:

- Security through obscurity
- E' impossibile fare auditing, ricerca di bug "per tentativi" (black box)
- "Reinventing the wheel"
- Anche trovando un bug, è difficile comprenderne esattamente la natura, e spesso i vendor non approfondiscono

... ma i torti non stanno da una parte sola ...

OSS:

- Non sai “chi è il padre” del codice che usi (trojans, anyone?)
- Non ci sono impegni commerciali: le release possono avvenire troppo presto... o mai.
- Falso senso di “sicurezza a priori” per la presenza del codice... ma l'avete letto?

CSS:

- Il fatto che non ci siano bug pubblici non significa che non ne esistano...
- Dipendenza totale dalla casa produttrice per il rilascio di patch... e a volte bisogna convincerli!
- Rassegnata fiducia in quello che state eseguendo (spyware and backdoors, anyone?)

Nessun mondo è perfetto...

- La morale è che non è corretto a priori affermare che il software open source sia “più sicuro” di quello closed source.
- Volete un esempio? Andate su: <http://packetstormsecurity.nl> e troverete “exploit” in abbondanza per ogni piattaforma che vi venga in mente...
- Addirittura, ad un'occhiata superficiale, vi sono più exploit per programmi open source!

Un problema (anche) economico

- Noi sostenitori dell'open source siamo abituati a discorsi romantici, ma la sicurezza ha un solido conto da presentare
- Gli investimenti per aggiornare e riscrivere il codice non sicuro sono notevoli, per gli stessi produttori!
- Lo sforzo economico e lavorativo per mantenere “up to date” l'ambiente di produzione è enorme
- Il fattore rischio creato dall'insicurezza non va trascurato: la diminuzione del rischio operativo è tradizionalmente considerata un obiettivo per qualsiasi azienda!
- E non abbiamo nemmeno parlato dei **danni...**

Full disclosure

- Strettamente connesso al discorso dell'open source è il tema della “full disclosure”, che potrebbe essere”, visto come un metodo di analisi “open”.
- Full disclosure significa la pratica, diffusasi negli ultimi anni, di annunciare pubblicamente le vulnerabilità scoperte, diffondendo spesso anche degli “exploit” dimostrativi.
- Molti esperti di sicurezza applicano la disclosure su forum aperti (come bugtraq, <http://onlineonline..securityfocussecurityfocus.com>), in genere dopo aver avvertito del problema i produttori ed aver loro concesso un lasso di tempo ragionevole per risolverlo.

... ma non tutto è oro ...

- La full disclosure, specie su progetti open source in cui la advisory spesso contiene anche una patch preliminare, ha contribuito alla soluzione di molti problemi di sicurezza.
- Tuttavia, molti indicano la disclosure come responsabile dell'aumento vertiginoso del fenomeno degli “script kiddies”, i teppistelli che utilizzano in modo ignorante vulnerabilità scoperte da altri per creare danni.
- Di conseguenza si sta riaffermando il “blackhat mood”, mentre i danni causati dai ragazzini hanno suscitato un fiorente mercato per le società di sicurezza informatica, su alcune delle quali si potrebbero affrontare lunghi discorsi...

La sicurezza si fa open source

- Anche nel campo dei prodotti per la creazione di sistemi di sicurezza esistono progetti open source di qualità pari, se non superiore, agli equivalenti del mondo commerciale:
- Firewall: IPFilter,, IPChains.....
- IDS: Snort, LIDS, AIDE, TripWire.....
- Criptografia: GPG
- Implementazioni di IPSEC e Kerberos
- Vulnerability scanner: Nessus, SAINT
- Unica eccezione: antivirus (i costi di ricerca sono elevati...)

Perché scegliere l'open source, in definitiva?

Non perché sia intrinsecamente più sicuro, o perché sia genericamente fatto meglio, ma perché lascia all'utente e all'amministratore di sistema la possibilità di adoperarsi attivamente per proteggere la propria sicurezza. Chi sceglie soluzioni closed source sceglie di affidarsi esclusivamente alle capacità dei vendor. In conclusione il modello open source responsabilizza la comunità e l'utente, e li mette in grado di agire, anche se non è la panacea tutti i mali.

.....

Grazie per l'attenzione!



vincenzo.calabro@computer.org