Sicurezza, Sistemi Operativi ed Hardware: un connubio che viene da lontano

Ultimamente la relazione tra Sicurezza, Sistemi Operativi e funzionalità hardware sta ricevendo molte attenzioni. In realtà, nella scala dei tempi dell'informatica, questa relazione viene da molto lontano. In questo articolo proponiamo una storia, sfortunatamente molto parziale, della sicurezza dei sistemi operativi per mettere in luce le origini di alcune odierne soluzioni e dare delle indicazioni plausibili su cosa potrebbe succedere nel prossimo futuro.

Partiamo considerando la situazione negli anni '60. In quegli anni i maggiori, se non gli unici, utenti interessati alla sicurezza dei sistemi operativi e delle applicazioni, erano i militari. Ed in ambito militare, la sicurezza è per lo più percepita in maniera olistica ed 'assoluta'. Gli altri principali attori di quegli anni sono le università e gli enti di ricerca, che contribuivano a proporre soluzioni sia per il mercato che per le esigenze governative e militari. Citiamo, ovviamente senza approfondire, la ben nota nascita di Internet, avvenuta negli stessi anni.

Nel 1965 MIT, AT&T, IBM e GE decisero di sviluppare in comune un sistema operativo 'sicuro' chiamato Multics. Questo sistema operativo doveva essere:

1. progettato top-down
2. capace di supportare almeno 1000 utenti contemporaneamente
3. 'Reliable'
4. 'With sufficient control of access to allow selective sharing of information'

ed avere molte altre caratteristiche di sicurezza militare. Purtroppo il sistema operativo riusciva a funzionare con 1 utente, alle volte con 2 ma sicuramente non con 3. Uno dei motivi di questo insoddisfacente risultato era che le richieste di sicurezza intrinseca al sistema, garantite da uno sviluppo controllato top-down, portarono ad una eccessiva complicazione del software rispetto alle piattaforme hardware disponibili in quegli anni.

Così nel 1969 AT&T convinse i partner a chiudere lo sviluppo di Multics, anche se Multics verrà sviluppato da altri ancora per un'altra decina di anni.

Thompson e Ritchie, due giovani sviluppatori che erano appena stati assunti per lavorare su Multics, si trovarono così senza un progetto su cui lavorare. Sotto la spinta di Thompson, in poco più di un mese svilupparono un nuovo sistema operativo, tutto l'opposto di Multics, dal nome Unix. Le caratteristiche principali di Unix sono:

1. sviluppato bottom-up
2. totalmente modulare
3. multi-utente, multi-tasking, multi-...

Lo sviluppo di Unix continuò sino ai primi anni '90, e se lo Unix che conosciamo oggi risale quindi a circa quindici anni fa, i principi fondamentali su cui si basa sono però ancora oggi quelli del 1969.

Ma per noi l'importanza di Unix non è tanto nel sistema operativo in se, ma nel modello e nelle idee su cui è basato. Infatti, a parte il mondo mainframe, possiamo dire che praticamente tutti i sistemi operativi attuali sono in un modo o nell'altro 'figli' di Unix. (Tra parentesi, tra il 1979 ed il 1987 Microsoft produsse Xenix, una delle versioni di Unix con il maggior successo e più grande distribuzione negli anni '80.) E' perciò importante capire il modello di sicurezza di Unix per arrivare a comprendere la situazione di oggi dei sistemi operativi.

Visto il fiasco di Multics, Unix è stato costruito a partire da concetti semplici e modulari. In pratica non si parla di sicurezza come è intesa nel mondo militare, ma il principale problema è quello della integrità. Infatti in un sistema multi-utente e multi-processo il primo problema da risolvere è la gestione delle risorse in modo che tutti vi abbiano accesso ma al contempo nessuno abusi di esse o acceda ad esse quando queste sono in uso a qualcun altro. Questo ovviamente garantisce l'integrità delle elaborazioni.

Il modello di sicurezza adottato divide semplicemente il mondo in due: il controllore ed i controllati.

Se guardiamo agli utenti, le risorse sono amministrate e controllate dall'amministratore che può accedere anche a qualunque dato di ogni utente; gli utenti normali sono limitati solo perché non possono accedere alle risorse riservate all'amministratore od ad un altro utente, per il resto non vi sono limiti a cosa un utente può fare. Chiaramente un modello di sicurezza di questo tipo

difficilmente può imporre politiche di sicurezza di tipo militare.

Questo modello di sicurezza è applicato anche a livello del sistema operativo. Infatti il sistema operativo è diviso in due: kernel-space e user-space. Il kernel-space è la parte più basilare del sistema operativo, quella che gestisce le risorse hardware e l'unica che vi ha accesso diretto. L'user-space è invece la parte di sistema operativo che interfaccia l'utente e l'ambiente in cui vengono eseguite le applicazioni (nel senso di codice) degli utenti.

Il seguente è un punto molto importante per noi: si capì ben presto che l'unico modo di proteggere le parti essenziali del sistema operativo e controllare l'accesso all'hardware, è di avere un meccanismo esso stesso hardware che isoli il kernel-space dall'user-space. Il problema principale è quello di garantire l'integrità del codice che viene eseguito in kernel-space e solo un meccanismo hardware può garantire che il codice in kernel-space non sia modificato o eluso per errore o volontariamente.

La modalità di implementazione di questa divisione hardware è abbastanza semplice: un programma eseguito in user-space, mediante una procedura detta di solito system-call, può invocare il kernel ad esempio per accedere a dati su di un disco. A questo punto l'hardware interrompe l'esecuzione del programma e attiva il kernel che verifica i parametri con cui è stato chiamato ed esegue il codice richiesto.

Vediamo così che la soluzione più semplice ed efficace per risolvere un problema di sicurezza, anche se solo di integrità delle elaborazioni, è data dall'utilizzo di una procedura software basata su di un dispositivo hardware.

Ma questo modello di sicurezza ha dei limiti molto ovvi ed evidenti. Infatti già a partire dalla metà degli anni '70 furono studiati dei modelli di sicurezza molto più avanzati, il più famoso dei quali è il modello di Bell-LaPadula del 1975. Possiamo riassumere semplicemente, con il rischio di banalizzare, i concetti principali dei modelli più avanzati di sicurezza come segue. Prima di tutto gli utenti di un sistema informatico vanno divisi almeno in tre gruppi: gli utenti normali, gli amministratori del sistema ed i responsabili della sicurezza. Tipicamente gli utenti sono gli utilizzatori finali, gli amministratori sono coloro che installano e configurano il sistema e i responsabili della sicurezza coloro che impongono le regole di utilizzo.

Per poter formulare le regole di sicurezza è necessario classificare tutti i file presenti nel sistema (sia che essi siano programmi che dati) e tutte le azioni possibili su/con essi, ed infine assegnare sia agli amministratori che agli utenti i permessi necessari e sufficienti per poter eseguire azioni sui dati. Si noti come anche gli amministratori sono limitati nelle loro azioni dalle regole imposte dai responsabili delle sicurezza.

Ovviamente è molto difficile e gravoso formulare tutte queste regole, e questo al momento rimane un problema non realmente risolto. Ma ipotizzando di essere riusciti a farlo (ed in situazioni semplici è ovviamente possibile) si viene a creare un database di regole da imporre, detto Access Control Database. Questo database ovviamente deve essere isolato e protetto, altrimenti un programma, un amministratore od un utente potrebbero manometterlo od eluderlo. Anche il modo di imporre le regole, detto Reference Monitor, deve essere protetto ed isolato per lo stesso motivo. Quindi tutto il sistema di sicurezza, chiamato Trusted Computing Base (TCB), deve essere protetto ed isolato. Come dovrebbe essere ormai evidente, l'unica maniera per proteggere l'intero TCB è in hardware.

D'altra parte questo può essere visto come una estensione del meccanismo hardware di separazione tra kernel-space e user-space visto precedentemente se pensiamo, come si fa spesso, di inserire i controlli di sicurezza realizzati in hardware al momento del passaggio tra lo user-space ed il kernel-space.

Qualunque sia il sistema di sicurezza avanzato che si voglia implementare, il sistema di sicurezza stesso ed il modo della sua imposizione devono essere protetti dall'hardware, altrimenti il rischio che essi possano essere manomessi od aggirati è relativamente alto. Negli anni '70 e '80 la tecnologia non era ancora capace di produrre moduli hardware che potessero soddisfare questi scopi, ma oggi ovviamente si. In particolare i moduli hardware crittografici sono alla base di qualunque sistema di questo tipo.

Ma torniamo ora alla nostra storia perché quello che è successo negli anni '80 non è sicuramente l'implementazione di quanto abbiamo appena descritto. Al contrario gli anni '80 vedono la nascita del Personal Computer abbinato al DOS, un sistema operativo ancora più semplice ma necessario per la limitatezza della piattaforma hardware su cui viene eseguito. In questo caso non possiamo parlare di sicurezza nel vero senso della parola visto che si tratta di un sistema mono-utente in cui,

almeno inizialmente, si può eseguire solo un programma alla volta. Non vi sono quindi amministratori, ne utenti ne responsabili della sicurezza in un ambiente ad utente unico.

D'altra parte il Personal Computer è il punto di avvio di un'altra rivoluzione: l'informatica di massa. Negli anni '90 molte cose succedono parallelamente: lo sviluppo delle applicazioni, delle comunicazioni ed anche dell'hardware sempre meno costoso e più potente ed in grado di offrire prestazioni spesso impensabili solo un paio di anni prima.

In quegli anni lo sforzo di tutti quanti, scrivente incluso, era più mirato a creare applicazioni, aggiungere funzioni, ottenere di più con meno, troppo spesso purtroppo a scapito della sicurezza. La rivoluzione informatica della fine degli anni '90 è stata probabilmente possibile solo perché si sono lasciate da parte le preoccupazioni dovute alle problematiche di sicurezza, ma oggi possiamo dire che ne stiamo pagando il conto.

Ad ulteriore prova di ciò, basta notare che lo sviluppo hardware a partire dall'inizio degli anni '90 offre delle caratteristiche di sicurezza che i sistemi operativi non sfruttano. Ad esempio, le CPU di tipo INTEL a partire dal i386 hanno ben 4 'anelli' hardware che generalizzano la divisione kernel-space / user-space. In linea teorica un sistema operativo potrebbe oggi sfruttare i 4 anelli per implementare politiche di sicurezza più avanzate, quali isolare le librerie di sistema dalle applicazioni oltre ad isolare il kernel dal resto. Ma nessuno dei principali sistemi operativi sfrutta questa opportunità offerta dall'hardware.

Il risultato della concentrazione quasi totale di tutti quanti sullo sviluppo di nuove applicazioni e l'allargamento dei possibili usi dell'informatica ha portato al successo odierno ed alla importanza odierna dell'informatica. La sfida degli anni '90 è stata ampiamente vinta: oggi non c'è transazione economica che non sia fatta, almeno per parte del suo cammino, in modo elettronico. L'informatica è ovunque, e ben poco funzionerebbe nella nostra vita senza il supporto degli elaboratori. Il valore economico delle transazioni effettuate in via informatica, sia essa pubblica (internet) che privata, è enorme ed in questa situazione non ci si meraviglia che qualcuno voglia approfittarsene in modo illegale.

Le problematiche di sicurezza ritornano quindi oggi in primo piano, molto si è già fatto e si sta facendo per rendere i sistemi più sicuri. A parte eliminare i più semplici errori di programmazione

nei miliardi di righe di codice scritte negli ultimi anni, per poter introdurre sistemi di sicurezza più avanzati è necessario utilizzare moduli hardware per isolare e proteggere i meccanismi di sicurezza stessi.

L'esempio più semplice è dato dal problema di dove archiviare le chiavi segrete che usiamo per cifrare dati sensibili. Una soluzione ragionevole è di metterle su smart-card o device simili (sempre una soluzione hardware!) o di avere un device hardware nella scheda madre del nostro PC che svolge la funzione di salvadanaio delle chiavi segrete. In pratica a ben pensarci le due soluzioni differiscono solo per il luogo ove si trova il device hardware, uno nel nostro portafoglio l'altro sulla scheda madre del PC, ma per il resto a livello di funzionalità sono simili.

Come abbiamo già scritto, le funzionalità crittografiche sono alla base dei possibili moduli di sicurezza hardware e quindi non deve stupire che queste siano le prime ad essere implementate.

Visto che l'hardware oggi offre la possibilità di implementare misure di sicurezza, e che una volta implementate in hardware queste risultano molto affidabili, il problema principale che oggi dobbiamo affrontare è quali misure di sicurezza vogliamo implementare in hardware ed in quale modo. Su questo punto vi sono molti progetti attivi, molte proposte, idee ed opinioni differenti (a partire dal Trusted Computing Group TCG/TCPA con il modulo TPM). Ma quello che ancora manca è di identificare chiaramente uno o più modelli di sicurezza per i futuri sistemi operativi e le applicazioni che li utilizzeranno, perché i moduli hardware sono utili solo ad implementare modelli di sicurezza, non a crearli. Una cosa comunque è quasi certa: che in futuro sempre più funzionalità di sicurezza dei sistemi operativi e quindi delle applicazioni informatiche stesse, saranno basate su moduli hardware.

Andrea Pasquinucci

Andrea Pasquinucci (PhD in fisica teorica, CISA, CISSP) è un consulente freelance in sicurezza informatica. Le sue attività principali sono in progetti strategici e globali di sicurezza ICT, governance, compliance, audit e formazione. I suoi campi di specializzazione tecnica sono la sicurezza delle reti e dei sistemi operativi, e la crittografia.

Precedentemente ha svolto per 10 anni la carriera accademica negli USA ed in Europa. Si dedica ancora all'insegnamento ed a progetti di ricerca universitaria. E' stato un membro del Comitato Direttivo e del Comitato Tecnico-Scientifico dell'Associazione Italiana Sicurezza Informatica (CLUSIT), ed è socio fondatore e membro del Comitato Direttivo dell'Associazione Italiana Professionisti Sicurezza Informatica (AIPSI), capitolo Italiano di ISSA.

Alcune note storiche on-line:

[1] Storia di Multics:
http://www.multicians.org/history.html
http://www.multicians.org/f7y.html

[2] Storia dei sistemi operativi:
http://www.softpanorama.org/History/os_history.shtml
http://snap.nlc.dcccd.edu/learn/drkelly/brf-hist.htm
http://snap.nlc.dcccd.edu/learn/drkelly/hst-hand.htm
http://en.wikipedia.org/wiki/History_of_operating_systems
http://en.wikipedia.org/wiki/Operating_systems_timeline