



## La sicurezza in Unix – *Prima parte*

Quando si parla di sicurezza in Unix, si deve andare sempre con i piedi di piombo. Da quando fu creato questo fantastico sistema operativo Open Source, tutti i cyber-criminali hanno da sempre cercato di entrare in un sistema unix per prenderne i comandi di utente root.

Dobbiamo dire che la storia di Unix ci insegna che questo sistema operativo ha avuto nel tempo un forte incremento per l'implementazione della sicurezza, fino a quando riuscì a superare altri sistemi operativi proprietari.

Ovviamente, anche con Unix, è possibile effettuare un hacking, anzi paradossalmente esistono molte più risorse per “catturare” le falle di sicurezza per questi sistemi piuttosto che ad esempio per Windows.

Da sempre l'etica hacker, definisce un exploit di hackeraggio verso sistemi esclusivamente proprietari, ma non dobbiamo dimenticare che almeno per le aziende che fanno business con Unix, etica o no, vengono irrimediabilmente attaccate da qualche persona non autorizzata. Come un buon hacker sa, prima di attaccare un sistema, si devono almeno verificare le vulnerabilità aperte. Per fare questo, si fa una mappatura delle vulnerabilità e si cerca di trovare qualche programma “bucabile”. Innanzitutto dovremmo prima definire quale tipi di attacchi possono essere portati ad un sistema operativo : locale e remoto. Noi ci concentriamo su quello remoto, che di sicuro è quello più soggetto agli attacchi. Per effettuare un attacco remoto, il cyber-criminale deve necessariamente trovare un porto (e non una porta che è la traduzione italiana sbagliata ma più comune di port) aperto.

Per fare questo ci vuole una buona conoscenza dei protocolli TCP/IP e UDP. Infatti, quando parliamo di attacco remoto, stiamo sostanzialmente parlando di entrare nel sistema grazie allo sniffer di pacchetti che viaggiano su internet. Detto questo, in Unix, così come in altri sistemi operativi, possono esservi vari servizi in ascolto. Più servizi ci sono, più attacchi possono essere fatali per il nostro business. L'ideale ovviamente sarebbe quello di chiudere quanti più servizi possibili, ma questo non è sempre possibile, soprattutto se c'è un'evidente esigenza di mantenere aperto quel porto. La prima soluzione, che generalmente il lettore pensa in questo momento, è sicuramente l'uso di un firewall. Bene, chiunque lo ha pensato deve ricredersi, perchè un firewall di per sé non protegge al cento per cento il porto aperto da sguardi indiscreti. Esiste, infatti, anche la possibilità di aggirare un firewall Unix, per mezzo dell' IP forwarding. Ma questo lo vedremo meglio nelle prossime puntate.

Detto questo, per questa prima parte ci concentriamo genericamente su due tipologie di attacco, che sono anche quelle con più facili contromisure. Analizziamo l'attacco di forza bruta. Un attacco di questo tipo serve per catturare la password dell'utente root per poi avere accesso a tutto il sistema come amministratore.

Ci sono tantissimi programmi che scansionano migliaia di password nel giro di qualche secondo, automaticamente e senza che il firewall linux se ne accorga. In questi programmi, infatti, basta inserire un intervallo di ricerca password e il software hacker cercherà di effettuare direttamente l'accesso con alcune password, informando l'utente qualora sia avvenuto un accesso.

Per difendersi da questo tipo di attacco, c'è sempre la solita buon vecchia regola che dice di utilizzare password difficili da catturare oppure potremmo usare programmi appositi che ci realizzano password impossibili da decifrare.

Il problema, è che la maggior parte degli utenti utilizzano password non complesse, facili da ricordare e dimenticano o anche per pigrizia, di cambiare almeno ogni mese la root password. Un altro tipo di attacco molto utilizzato è quello del buffer overflow.

Questa tipologia di attacco è quella di sovvertire la funzione di un programma privilegiato in modo che l'attaccante possa prendere il controllo di quel programma, controllando anche l'host. Una forma d'attacco di questo tipo combina la tecnica d'iniezione con la corruzione dei records.

Ovvero si trova una variabile soggetta ad overflow e si inserisce l'attacco per avere controllo del programma. Una soluzione a questo problema è semplicemente : realizzare buoni programmi. Cosa più facile a dirsi che a farsi. Nelle prossime puntate descriveremo meglio questa tecnica e indicheremo qualche utile software di difesa.

Dott. Nicola Savino

<http://www.seensolution.com>

## **PROFILO DOTT. SAVINO**

Socio Sostenitore ANORC.

Laurea in Ingegneria Informatica presso l'Università Federico II di Napoli.

Presidente della Commissione Tecnica Open Source in ANORC.

Master ANORC, Xplor Italia e AssolIT in Gestione e Conservazione Digitale e Sostitutiva dei Documenti - Il Responsabile della Conservazione sostitutiva : ruoli, funzioni, strategie.

Relatore come esperto della Conservazione Sostitutiva a Docubusiness.

Corso avanzato di 3 giorni sulla Gestione del documento informatico negli adempimenti amministrativi del dottore commercialista presso la Sede Ordine dei Commercialisti ed Esperti Contabili di Napoli.

ISO 9001:2000 certified Networking Concepts Test.

ISO 9001:2000 certified Test of Knowledge of Joomla 1.5.

ISO 9001:2000 certified Information Technology Awareness and Terminology Certification.

Information Management Day IBM 2009.

Corso avanzato su Windows Server 2008 e Red Hat Linux Technical Overview a cura di Microsoft, Computer Gross S.p.a. e Fujitsu.

Seminari tecnici di 20 ore IBM Power System I presso l'Università Federico II di Napoli.

Documento Digitale : procedure per rendere l'ufficio più agile. Roma 1 Ottobre 2009.

Autore e giornalista presso la HTML S.r.l, sezione PMI.it (sito di informazione tecnologica dedicato interamente alle Piccole e Medie imprese Italiane)