Perché non serve (quasi mai) un antivirus su GNU/Linux (parte 1)

12 luglio 2008

Molti si chiedono se serva, o meno un antivirus su GNU/Linux. Come primo post di questo blog, voglio spiegare bene la questione, anche perché in rete si trovano notizie discordanti e spesso fasulle. I produttori di antivirus è già da qualche anno che predicano la nascita di virus su GNU/Linux, ma non se ne sono visti. E i motivi sono tecnici e cercherò di spiegarli.

Prima di tutto, una definizione dei diversi tipi di virus, o meglio di "malware".

- **Virus**: un virus è un programma malevolo che usa un altro programma come veicolo di diffusione e replicazione, esattamente come fanno i virus biologici che usano le cellule per riprodursi. Un virus ha quindi bisogno di un altro programma da infettare.
- **Trojan**: un trojan (cavallo di Troia) è un programma che fa credere all'utente di essere utile, mascherandosi da qualcos'altro. Ad esempio alcuni trojan appaiono inizialmente come dei codec per la riproduzione di contenuti multimediali.
- **Worm**: un worm (verme) è un programma malevolo che può riprodursi senza bisogno di farsi veicolare da un altro programma.
- **Toolkit/Rootkit**: un toolkit può essere malevolo o no. Con lo stesso termine infatti si indicano sia programmi utili (come le librerie GTK) sia programmi malevoli. In questo secondo caso ci si riferisce a librerie che vanno a sostituirsi o affiancarsi a quelle di sistema o di programmi per procurare danni, nascondendosi in modo da sfuggire all'attenzione dell'utente. Quando un toolkit coinvolge il kernel del sistema operativo (ad esempio come finto driver), si parla di **rootkit**. Di norma l'uso di questo malware è quello di installare una **backdoor** ("porta sul retro") attraverso cui l'attaccante può entrare nel sistema colpito e prelevarne i dati o addirittura prenderne il controllo.
- **Wabbit**: è un programma malevolo che non usa i servizi di rete o altri file o programmi per riprodursi. Un esempio è la **fork bomb**.
- **Altri tipi di malware:** altri tipi di malware si distinguono più per lo scopo che per le modalità di azione e diffusione, di solito riconducibili alle categoria precedenti. Tra questi ricordiamo gli **spyware** (codice spia), gli **adware** (pubblicità indesiderate che compaiono sul desktop) e i **keylogger**, programmi che registrano l'attività dell'utente soprattutto al fine di scoprire le password e i numeri di carta di credito digitati. Inoltre la diffusione di formati di file che possono contenere codice anche se non sono programmi veri e propri (ad esempio i formati documenti che possono contenere macro o le pagine web che possono contenere javascript) ha portato alla nascita di **macrovirus**.

Bene, ma come agisce un malware?

Non è sufficiente che il malware entri a contatto con il sistema (ad esempio attraverso uno scambio di file, una e-mail o la visualizzazione di una pagina web), ma è necessario che entri in esecuzione. Difatti gli antivirus mettono i file infetti in "quarantena", ossia in una cartella controllata dove non possono più agire.
Quando il malware entra in contatto con il sistema deve presentarsi uno dei seguenti casi affinché esso possa entrare in esecuzione:

- una azione volontaria dell'utente mette in esecuzione il malware: questo è il caso dei trojan e di molti worm;
- il malware entra in esecuzione anche in mancanza di una azione volontaria: in tal caso è stata sfruttata una **vulnerabilità**.

Una vulnerabilità è una falla di un programma che produce un comportamento non previsto dal programmatore o considerato (a torto) non pericoloso.

Ed ora, ecco perché un antivirus è quasi sempre inutile.

1. I permessi

I sistemi operativi di tipo Unix hanno una rigida e complessa gestione dei permessi. Ogni utente, e quindi ogni programmi eseguito da tale utente, può fare con un file solo ciò che è consentito in base ai permessi che egli possiede. Si consulti la guida del comando sudo per approfondire la logica dei permessi. Questo implica alcune conseguenze:

- i programmi utente sono separati da quelli di amministrazione;
- I programmi utente possono agire solo sulla home di quell'utente, non sui file di amministratore né su quelli di altri utenti;
- i programmi per essere eseguiti devono avere lo speciale attributo di eseguibili.

In base a ciò, un malware che agisce a livello utente non può creare danni al sistema, ma può al limite cancellare o infettare solo i file appartenenti a quel determinato utente.
Di norma nessun sistema di tipo Unix installa i programmi (neppure i programmi utente) nella directory home dell'utente. Ciò, unito alla suddetta gestione dei permessi, mette al riparo il sistema dall'infezione da parte dei tradizionali virus che non trovano eseguibili a cui "attaccarsi". I **worm** non possono agire perché per farlo devono avere i permessi di esecuzione. I **rootkit** non possono installarsi autonomamente in quanto caricare un modulo/driver nel kernel richiede i permessi di amministrazione.
Ciò a meno di vulnerabilità del sistema. Infatti una vulnerabilità grave può permettere al malware di superare tali restrizioni e acquisire i permessi di amministratore.

Ciò è già accaduto per i sistemi di tipo Unix. Il primo worm della storia è nato proprio per Unix sfruttando una vulnerabilità.

2. Essere open source

Un software open source, e quindi GNU/Linux, ha la caratteristica di avere il codice sorgente liberamente consultabile e modificabile. Questo apparentemente potrebbe rendere meno sicuro il sistema. In teoria, se tutti conoscono il codice sorgente, chiunque può scoprirne le vulnerabilità e quindi sfruttarle con fini fraudolenti. Nella pratica, però, si realizza l'esatto opposto: proprio perché tutti possono scoprire facilmente le vulnerabilità, esse possono venire tempestivamente corrette. Molte vulnerabilità vengono infatti corrette ancora prima che possano essere sfruttate a danno del sistema.

Navigare sul Web con un browser open source è più sicuro che navigare con uno proprietario e usare una suite per l'ufficio open source è più sicuro che usarne una proprietaria.

3. Rafforziamo i permessi

Sono stati adottati vari meccanismi preventivi per rafforzare la sicurezza del sistema come:

- l'uso di chiavi di autenticazione per il software e i repository che assicurano la provenienza originale e sicura degli stessi;
- la necessità, quando si esegue un programma nella directory corrente, di anteporre il suo percorso `./` in modo tale che un programma che abbia lo stesso nome di un comando comunemente usato, non possa essere per sbaglio eseguito al posto di tale comando (questa semplice precauzione ha stroncato la diffusione di worm come **ls**);
- ulteriori rafforzamenti del meccanismo dei permessi come SELinux (sviluppato dalle forze armate statunitensi) e AppArmor (sviluppato da Novell e presente in Ubuntu): tali sistemi creano i cosiddetti "contesti": ad esempio una pagina html creata nella home dell'utente, anche se trasferita nella directory di Apache `/var/www` non funzionerà in quanto nata in un contesto differente; un programma presente nella directory utente non verrà eseguito se trasferito in una directory di sistema come `/usr/bin/`.

4. Unix e il malware

Per comprendere quanto i sistemi Unix siano sicuri è utile consultare alcune fonti:

- la pagina di uno dei programmi più noti, apprezzati e premiati nella lotta al malware chkrootkit. Questa elenca solo una decina di malware (sia rootkit che worm) in oltre 10 anni di sviluppo del programma. Alcuni di questi sono worm ormai desueti come il citato **ls**, altri sono **rootkit** solo per alcuni sistemi Unix che quindi non coinvolgono gli altri sistemi della stessa famiglia (ad esempio un malware per Solaris non può agire su GNU/Linux o *BSD), altri ancora si riferiscono a determinate versioni del kernel di tali sistemi (infatti una volta corretta la vulnerabilità il malware è diventato innocuo). Sfogliando il changelog del programma si nota che i malware aggiunti annualmente per i sistemi Unix supportati dal programma sono dell'ordine di qualche unità;
- la pagina sui virus per Linux nella documentazione internazionale di Ubuntu, nella quale si illustrano i pochi malware conosciuti per Linux, la maggior parte dei quali nei fatti risulta innocua (perché, per esempio, necessità dei permessi di amministratore).

Nella realtà il concetto di virus è praticamente sconosciuto nei sistemi di tipo Unix essendo i pochi finora scoperti non in grado di diffondersi efficacemente, perché necessiterebbero di entrare fraudolentemente in possesso dei permessi di amministratore.

Nella prossima puntata vedremo le eccezioni, ovvero quando è utile avere un antivirus.

https://guiodic.wordpress.com/2008/07/17/virus-e-gnulinux-precauzioni-intelligenti/#more-75

Virus e GNU/Linux: prendiamo un po' di precauzioni intelligenti (terza parte)

17 luglio 2008

Abbiamo visto perché su GNU/Linux non ci sono virus. E abbiamo visto i casi particolari in cui, comunque, può essere utile installare un antivirus.

Ricordo la definizione di virus: un programma che si attacca ad un altro programma per **riprodursi**. Si può fare qualcosa del genere su GNU/Linux o in generale su un sistema di tipo Unix. Sì, certo. Il problema è che muore lì. Senza permessi il virus non va da nessuna parte e rimane confinato. Fine della riproduzione.

Però non ci sono solo i virus…

Vero. Ad esempio ci sono anche i **trojan**. I trojan sono programmi che fanno finta di essere utili ma in realtà sono malevoli. Contro i trojan c'è poco da fare, anche se qualcosa si può fare: ad esempio i meccanismi di rafforzamento dei permessi che abbiamo già visto possono limitare l'azione di molti tipi di malware. Ma la vera difesa contro di essi, almeno nell'ambito desktop, non è un antivirus, né SELinux o Apparmor. **La vera difesa è l'intelligenza dell'utente.**

Proviamo a vedere cosa possiamo fare per eliminare pressoché ogni tipo di problema, prendendo alcune semplici precauzioni.

La gran parte di esse consiste semplicemente nell'usare il sistema operativo come dovrebbe essere usato. Niente di più. Nessuno sforzo particolare, nessun programma che ruba cicli di clock preziosi per poter finire prima il ripping del DVD che stiamo trasferendo sul pc, nessun "terrore" ogni volta che succede qualcosa di strano.

Ecco una lista di buoni comportamenti:

- **Non installate programmi né date permessi di esecuzione ad un file né eseguitelo tramite il comando *sh* se non siete è certi della sua provenienza e affidabilità**
  *Un giorno potrebbe capitarci un bel file .deb da installare, allegato ad una e-mail. Agli utenti Windows capita spesso, molti famosi malware erano dei file .exe che promettevano chissà cosa. La gente ha incominciato piano piano a capire che non è bene fidarsi. Noi utenti GNU/Linux ancora non abbiamo queste "fortune" ma pensateci: l'eeepc usa XandrOS, una distribuzione derivata da Debian. I .deb di XandrOS di solito funzionano anche su Debian, Ubuntu e altre distro simili. Un click e qualcuno può fregarci. – "*Oddio, vado a prendere ClamAV!" *– Fermo lì. Basta non clickare sulla prima cosa che capita, no? E' più facile.*

- **Preferite i repository che possiedono una chiave di autenticazione GPG**
  Avete presente quando aggiungete un repository e poi compare un errore tipo:
- `W: Errore GPG: http://packages.medibuntu.org hardy Release: Le seguenti firme non sono state verificate perché la chiave pubblica non è disponibile`

  Ecco, questo accade perché medibuntu è autenticato da una chiave GPG (GNU Privacy Guard) che assicura all'utente che non ci stiamo sbagliando e che non siamo vittime di qualche scherzo dei server DNS del nostro provider. Il repository è autentico. Dobbiamo però aggiungere la chiave al sistema per permettere la verifica e così l'errore sparisce. Una piccola seccatura, certo, ma un bel vantaggio in termini di sicurezza.

- **Se disponibile, controllate che l'md5sum corrisponda a quello dichiarato sul sito del programma o file che avete scaricato**
  Molte volte sul sito da cui scarichiamo programmi o immagini ISO vediamo un codice indicato con MD5. Si tratta di un numero che identifica il file senza possibilità di errore (o meglio con una possibilità di errore così bassa da essere insignificante). Per sapere l'MD5 del file, una volta scaricato, basta dare il comando:

  ```
  md5sum nome_del_file
  ```

  Confrontiamolo con quello sul sito e così saremo sicuri al 100% che si tratta dell'originale.

- **Navigate preferibilmente con un browser open source aggiornato all'ultima versione disponibile, usate preferibilmente programmi open source e nativi per il sistema (per non dover usare Wine), anch'essi aggiornati**
  E qui qualcuno storcerà il naso: "Ma io uso Opera". E' un ottimo browser, seppure non migliore di Firefox (farò un confronto oggettivo fra i due appena posso). Ma è proprietario. E' difficile sapere cosa effettivamente fa Opera. Certo ci sono metodi per scoprire se manda dei dati a qualcuno che non dovrebbe conoscerli, ma qui il problema è un altro: la sicurezza. Firefox è uno dei browser più usati al mondo, ha alle spalle una grande fondazione finanziata da Google, e soprattutto il suo codice può essere letto da chiunque. Questo significa che è facile scoprire un problema di sicurezza e difatti accade

relativamente spesso. Ma tra la scoperta e la correzione passa pochissimo tempo perché essendo Open Source chiunque può apportare la correzione. Se quelli della Mozilla Foundation dormono (ma non succede) possono pensarci i programmatori della distribuzione GNU/Linux che usiamo. E spesso accade davvero così per programmi meno famosi di Firefox. Ecco quindi che un browser Open è più sicuro di uno proprietario e in generale qualsiasi programma Open Source è più sicuro che uno proprietario. Un buon motivo per usare OpenOffice al posto di MS Office anche su Windows.

- **Usate formati di dati sicuri**
  Evitate il più possibile i .doc e similari. Meglio .odt *o al massimo* .rtf*, se dovete mantenere la compatibilità con MS Office.*
- **Evitate di usare Internet Explorer con Wine tramite Ie4Linux a meno che non sia assolutamente necessario per verificare il funzionamento di vostre pagine web; in tal caso comunque preferite la visualizzazione di pagine locali, oppure usate il servizio IE NetRender**
  Navigare con Internet Explorer è come mandare un messaggio al mondo con su scritto:
  "Ehi ragazzi, sono qui, prendetemi!"
  *Non fatelo. Se quella c*** di pagina non vi viene visualizzata bene con Firefox, beh, non apritela e basta. Chi non sa fare siti web a norma non merita che voi gli facciate aumentare il contatore delle visite. Anzi mandategli una e-mail:*
  "Ehi ciccio, il tuo sito non funziona con Firefox!
  Lo sai che è un programma che è stato scaricato da 8 milioni di persone in un solo giorno?
  Che aspetti ad aggiornarti?"
  E se per caso parliamo della vostra banca, il mio consiglio è uno solo: cambiate banca. C'è poco da fidarsi. Se non hanno saputo fare bene l'impaginazione, figuratevi il resto.
- **Non eseguite mai Wine come utente root: in tal caso lascereste all'eventuale malware accesso alle directory di sistema**
  Dal punto di vista della sicurezza, Wine equivale a mettere Erode Antipa a guardia di un asilo nido. No, dai, sto esagerando. Ma è meglio tenerlo confinato al suo posto. L'ideale sarebbe usarlo con un utente secondario invece che con il nostro account principale.
- **Eseguite sempre gli aggiornamenti di sicurezza del sistema operativo: in Ubuntu è possibile accettarli in maniera predefinita tramite il gestore aggiornamenti; questa è la principale precauzione che mette al riparo dai malware: Ubuntu e Debian hanno infatti una gestione molto efficiente dei problemi di sicurezza**
  Un click e dormiamo sonni tranquilli.
- **Non usare distribuzioni per le quali sia scaduto il supporto di sicurezza**

  E' giunta l'ora di mandare in pensione la nostra amata Debian Potato
- **Usate cautela quando si è in possesso dei permessi di amministratore (tramite sudo, su, gksudo o kdesu), evitando di eseguire programmi di cui non si conosce l'affidabilità**
- **Non entrate nel sistema come root**
  E possibilmente non attivate affatto l'utente root; nel caso sia necessario, evitate comunque di navigare sul web, di scaricare posta e di usare programmi che interagiscono con la rete. Anche se è comunque difficile essere colpiti da malware in queste circostanze, se tutti non applicassero tale precauzione si renderebbe più facile il compito ai cracker aprendo loro le difese del sistema
- **Non siate paranoici**
  Chi viene da sistemi Windows è abituato a blindare tutto e avere paura; è portato ad attribuire qualsiasi malfunzionamento a qualche ignoto virus; ma GNU/Linux non è Windows: take it easy.

ATTENZIONE! Questi consigli vanno presi come tali. Non è necessario affrettarsi a installare l'ultima versione disponibile di Firefox dal sito di Mozilla: se contiene correzioni di sicurezza significative, verrà

segnalata tra gli aggiornamenti della distribuzione in breve tempo. Ripeto: *take it easy*

Alla prossima.