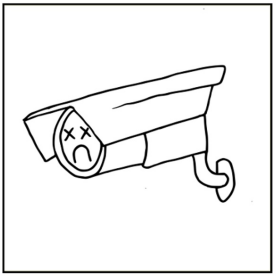
ATTENZIONE!



Guida all'autodifesa digitale
#2

SOMMARIO

Illustrazioni di Pinza666

QUALCHE ILLUSIONE DI SICUREZZA

Bene. Abbiamo cominciato facendo un tour delle possibili tracce involontariamente lasciate nel nostro computer e la quantità di informazioni che dei malintenzionati avrebbero potuto sottrarci. Ora non resta che mettere da parte qualche preconcetto.

SOFTWARE PROPRIETARI, OPEN SOURCE E LIBERI

Abbiamo visto che un software potrebbe fare un sacco di cose che non vorremmo. Perciò, è necessario fare il possibile per ridurre il problema. Da questo punto di vista, ai software liberi possiamo dare una maggiore fiducia rispetto al cosiddetto software “proprietario”: vedremo perché.

LA METAFORA DELLA TORTA

Per comprendere la differenza tra software libero e proprietario, usiamo la metafora della torta. Per

fare una torta, hai bisogno di una ricetta: è una lista di istruzioni da seguire, ingredienti da utilizzare e un processo di trasformazione. Allo stesso modo, la ricetta di un software si chiama “codice sorgente”. È scritto in una lingua che è pensata per essere comprensibile dagli esseri umani. Questa ricetta viene quindi trasformata in un codice interpretabile dal processore, un po’ come la cottura di una torta ci dà l’opportunità di mangiarla.

Il software proprietario è disponibile e “pronto da mangiare” come una torta industriale, senza la sua ricetta. È quindi molto difficile garantire i suoi ingredienti: è fattibile, ma il processo è lungo e complicato. Inoltre, rileggere una serie di milioni di aggiunte, di sottrazioni, di letture e di scritti in memoria per ricostruirne lo scopo e il funzionamento, non è proprio la prima cosa che si fa con un computer. Il software libero, invece, offre la ricetta per chiunque voglia comprendere o modificare il funzionamento del programma. È quindi più facile sapere cosa viene inviato al

nostro processore e, quindi, cosa accadrà ai nostri dati.

SOFTWARE PROPRIETARI: UNA CIECA FIDUCIA

Un software proprietario è un po' come una scatola impenetrabile: possiamo constatare ciò che viene richiesto al software, ha una bella interfaccia grafica, etc. Ma non possiamo davvero sapere in dettaglio come funziona. Non sappiamo se è costretto a fare ciò che gli viene chiesto di fare o se fa anche altre cose. Per scoprirlo dovremmo essere in grado di studiare il suo funzionamento, il che è difficile da fare senza codice sorgente … quindi dobbiamo fidarci ciecamente.

Windows e Mac OS X in primis sono enormi scatole, ermeticamente sigillate, su cui vengono installate altre scatole altrettanto ermetiche (da Microsoft Office agli anti-virus), che possono fare molte più altre cose rispetto a quelle che gli chiediamo.

Come per esempio, riunire le informazioni che questi software potrebbero averci sottratto oppure permettere l'accesso all'interno del nostro computer a backdoor fornite insieme al software, in modo che chiunque abbia la chiave possa hackerare i nostri computer. Dato che non possiamo sapere come è stato scritto il sistema operativo, possiamo immaginarci di tutto.
Dunque, lasciare che la riservatezza e l'integrità dei nostri dati si basino su programmi che devono essere considerati affidabili a scatola chiusa, è la più pia illusione di sicurezza. E installando questi software che sostengono sulla loro confezione di garantirci la sicurezza, mentre il loro funzionamento non è per nulla trasparente, non può certo risolvere i nostri problemi.

IL VANTAGGIO DI AVERE LA RICETTA: I SOFTWARE LIBERI

La maggior fiducia che possiamo accordare ad un sistema libero come GNU / Linux è principalmente correlata al fatto che abbiamo la

“ricetta”. Teniamo presente che non c’è nulla di magico: il software libero non lancia alcun “incantesimo di protezione” sui nostri computer.
Tuttavia, GNU / Linux offre maggiori possibilità di rendere l’uso dei computer un po’ più sicuro, in particolare consentendo di affinare le configurazioni di sistema. Questo troppo spesso implica però una conoscenza specializzata, ma almeno è possibile. Il mondo che si aggira attorno al software libero non è molto compatibile con l’introduzione di backdoor: è un modo di produzione collettivo, piuttosto aperto e trasparente, al quale partecipano persone molto diverse; insomma non è quindi facile lasciare dei regalini all’interno dei software senza che nessuno se ne accorga.
Tuttavia, tocca fare attenzione anche ad alcuni software che si definiscono open source. Essi consentono sì l’accesso alle proprie “viscere”, ma hanno anche delle modalità di sviluppo chiuse e opache. La modifica e la ridistribuzione di questi software è nella peggiore delle ipotesi proibita e

nella migliore autorizzata, ma resa nella pratica quasi impossibile. Solo il team originario può sviluppare il codice sorgente, quindi si può considerare che in pratica nessuno lo leggerà nel dettaglio e nessuno controllerà realmente il suo funzionamento.
Questo è il caso ad esempio di TrueCrypt, il cui sviluppo è stato interrotto a maggio 2014. Si trattava di un software di crittografia il cui codice sorgente è disponibile, ma il cui sviluppo è fermo e la licenza limita la modifica e la ridistribuzione. Secondo noi, per poter definire un software open source si dovrebbe giudicare anche come viene messo in commercio, e non basarsi solo su una promessa.
A meno che … la distinzione tra software libero e open source diventi sempre più sfocata: buona parte del software libero piu' importante viene scritto da impiegati dell'IBM o simili, e non andiamo ogni volta a guardare da vicino cosa scrivono.
Un altro esempio da mostrare sono le statistiche

di coloro che sviluppano il kernel di Linux- che è libero- e le aziende per cui lavorano, attraverso il numero di righe di codice sorgente modificate nell'ultimo periodo di tempo:

organizzazione	**percentuale**
Intel	20,4%
AMD	8,7%
Samsung	6,6%
Red Hat	4,8%
(sconosciuto)	4,0%
Linaro	3,8%
SUSE	3,6%
IBM	3,0%
(consulente)	3,0%
Solarflare Comm.	2,3%
MediaTek	1,8%
Cavium	1,8%
etc.	

Quindi non è impossibile che chi ha scritto in un angolo un pezzo di software su cui la comunità fa

affidamento, abbia potuto far scivolare all'interno pezzi di codice dannoso. Questo è stato anche il caso dell'errore noto come Heartbleed. Se usiamo solo software libero fornito da una distribuzione GNU / Linux non commerciale, è improbabile che ciò accada, ma c'è comunque una possibilità. Dobbiamo allora affidarci a chi si occupa della distribuzione perché studi il funzionamento dei programmi che vi sono integrati.

È comunque importante ricordare che questa fiducia può essere valida solo se non si installano cose a caso nel nostro sistema. Ad esempio, su Debian i pacchetti ufficiali della distribuzione sono "firmati", il che rende possibile controllarne l'origine. Ma se si installano pacchetti o estensioni per Firefox trovati su Internet senza controllarli, ci esponiamo a tutti i rischi di malware.

Per concludere e non farci più illusioni: gratis o no, non esiste un software che possa da solo garantire la privacy e l'intimità dei nostri dati; per

fare questo, ci sono pratiche associate all'uso di determinati software. Software scelti perché diversi elementi ci permettono di dare loro un certo livello di fiducia.

LA PASSWORD DI UN ACCOUNT NON NE PROTEGGE I DATI

Tutti i sistemi operativi recenti (Windows, Mac OS X, GNU/Linux) offrono la possibilità di avere utenti diversi sullo stesso computer. È importante sapere che le password che a volte proteggono questi account non garantiscono affatto la riservatezza dei dati. E' certo pratico e conveniente avere un proprio spazio con le proprie impostazioni (segnalibri, sfondi …), ma una persona che volesse avere accesso a tutti i dati sul quel computer condiviso non avrebbe grossi problemi ad ottenerlo: basta collegare l'hard-disk a un altro computer o leggerlo con un altro sistema operativo. Inoltre, se l'utilizzo di utente e password può avere alcuni vantaggi – ad esempio, la possibilità di bloccare lo schermo

quando ci si allontana per alcuni minuti- è necessario tenere presente che ciò non protegge realmente i dati.

LA "CANCELLAZIONE" DEI DATI

Ne abbiamo già parlato, il contenuto di un file diventato inaccessibile o invisibile non è che si volatilizza; ora spiegheremo perché.

LA CANCELLAZIONE DI UN DATO NON NE ELIMINA IL SUO CONTENUTO

... e può essere molto facile trovarlo. Infatti, quando "cancelliamo" un file – mettendolo nel cestino e poi svuotandolo – stiamo solo dicendo al sistema operativo che il contenuto di questo file non ci interessa più. Il sistema sta quindi eliminando la sua voce nell'indice dei file esistenti in modo da poter riutilizzare lo spazio per aggiungerci qualcos'altro in futuro. Ma potrebbero volerci settimane, mesi o anni prima che questo spazio venga effettivamente utilizzato per i nuovi file, e prima che i vecchi dati

scompaiano quindi effettivamente. Nel frattempo, se guardiamo direttamente ciò che è scritto sull'hard-disk, troviamo il contenuto dei file.
E' un'operazione abbastanza semplice, automatizzata da molti software di recupero o ripristino dati come PhotoRec.

UNA POSSIBILE SOLUZIONE: RISCRIVERE PIU VOLTE I DATI

Una volta riscritto lo spazio su un disco rigido, diventa difficile trovare ciò che c'era prima. Ma questo non è impossibile: quando il computer riscrive 1 su 0, il risultato è invece 0,95 e quando riscrive 1 su 1, 1.051 (per approfondire: Peter Gutmann, 1996, Secure Deletion of Data from Magnetic and Solid-State Memory – http://sugate.vado.li/.) Un po' come quando riusciamo a leggere su un taccuino ciò che era stato scritto su una pagina strappata via, grazie alle depressioni create sulla pagina vuota sottostante.
Diventa molto difficile invece, se non impossibile,

recuperare i dati quando vengono sovrascritti più volte da altri dati casuali. Il modo migliore per rendere inaccessibile il contenuto di questi file "eliminati" consiste quindi nell'utilizzare un software che garantisce questo tipo di scrittura multipla (azione chiamata "wipe" in inglese).

QUALCHE LIMITE DELLA POSSIBILITÀ DI RISCRITTURA

Anche se è possibile sovrascrivere più volte i dati su un hard-disk per renderli inaccessibili, ciò non garantisce la loro completa scomparsa.

DISCHI "INTELLIGENTI".

I dischi attuali riorganizzano il loro contenuto "in modo intelligente": parte del disco è riservata per sostituire spazi eventualmente difettosi. Queste operazioni di sostituzione sono difficili da rilevare e non possiamo mai veramente essere certi che la posizione che stiamo riscrivendo sia quella in cui il file era stato originariamente scritto. Per le

unità USB e SSD (Solid State Drive), è corretto dire che nella maggior parte dei casi in realtà si riscrive in un posto diverso. La memoria flash, che viene utilizzata dalle unità flash USB e dalle SSD, smette di funzionare correttamente dopo un certo numero di scritture e contiene chip che riorganizzano automaticamente il contenuto per diffondere le informazioni in posizioni diverse. Prendendo in considerazione questi meccanismi, diventa difficile garantire che i dati che si desidera distruggere scompaiano veramente nel nulla.

Malgrado ciò, esplorare un hard-disk per esaminarne l'interno richiede tempo e significative risorse materiali e umane … investimenti che non sono necessariamente così immediati e alla portata di tutti. Per i chip di memoria flash di una chiave USB o SSD, anche se non immediata, l'operazione può essere più semplice: servono solo un saldatore e un dispositivo per la lettura diretta dei chip di memoria, come ad esempio il PC-3000 Flash

SSD Edition, venduto come strumento professionale per il recupero dei dati su dispositivi flash danneggiati, al costo di circa 1.500 dollari.

FILE SYSTEM "INTELLIGENTI"

Un altro problema sono i file system "intelligenti". I file system sviluppati negli ultimi anni, come NTFS o ext4, tengono traccia all'interno di un log delle modifiche successive apportate ai file. Dopo un arresto improvviso del computer, consentono al sistema di riprendere semplicemente le ultime operazioni da eseguire, invece di dover riesaminare l'intero disco per correggere le incoerenze. Nel farlo, potrebbero di nuovo aggiungere tracce su quei file che uno voleva vedere scomparire. Ext4, il file system attualmente più usato sotto GNU / Linux, può funzionare con diverse modalità e dentro ai log inserisce solo i nomi dei file e altri metadati, non il loro contenuto. Anche altre tecniche, meno comuni, possono dare problemi: i file system con

scrittura ridondante e che continuano a scrivere anche in caso di errore, come i file system RAID; i file system che eseguono immagini istantanee del sistema (gli snapshot); i file system che si nascondono nelle cartelle temporanee, come i client NFS (file system di rete); i file system compressi etc.
Infine, non dobbiamo dimenticare che il file, anche se perfettamente cancellato, potrebbe aver lasciato tracce altrove.

CIÒ CHE NON SI SA

Riguardo ai CD-RW o ai DVD ± RW (riscrivibili), sembra che non sia stato condotto alcuno studio serio sull'efficacia della riscrittura. Le attuali raccomandazioni sono quindi di distruggere metodicamente i supporti di questo tipo che potrebbero aver contenuto dati che vogliamo far sparire (NIST, 2014, Guidelines for Media Sanitization - http://vugaso.vado.li).

QUANDO "CANCELLIAMO"

Non eliminiamo i file mettendoli nel "cestino". Ad esempio, quando si utilizza l'opzione "Cancella la cronologia" del browser Firefox, non si fa altro che "cancellare" i file. I dati diventano inaccessibili per Firefox, ma sono ancora accessibili all'interno dell'hard disk. Vale anche la pena sottolineare che la riformattazione di un hard disk non ne cancella a pieno il contenuto. Come nel caso della cancellazione dei file, la riformattazione non fa altro che rendere disponibile lo spazio dove si trovavano i contenuti precedenti, ma i dati rimangono fisicamente presenti sul disco. Allo stesso modo in cui distruggere il catalogo di una biblioteca non fa automaticamente sparire i libri presenti negli scaffali. Quindi possiamo sempre trovare i file dopo la riformattazione, come se fossero stati semplicemente "cancellati". PhotoRec offre anche questo tipo di funzionalità.

E PER NON LASCIARE ALCUNA TRACCIA?

Sfortunatamente, non esiste un modo semplice per risolvere radicalmente il problema. La soluzione meno difficile per ora, è avviare il computer con un sistema Live, come Tails, configurato per utilizzare la sola RAM. In questo modo è possibile non scrivere nulla sull'hard disk o sulla swap e mantenere le informazioni solo nella RAM, quindi solo fino a quando il computer rimane acceso.

SOFTWARE PORTATILI: UNA FALSA SOLUZIONE

Il cosiddetto "software portatile" è un software che non è installato su un determinato sistema operativo, ma che può essere avviato da una chiavetta USB o da un hard disk esterno – e quindi portato con noi per avercelo su qualsiasi computer.

È diventato facile scaricare queste applicazioni su Internet. "Pacchetti portatili" come Firefox + Tor o Thunderbird + Enigmail sono disponibili online.

Tuttavia, a differenza dei sistemi Live, utilizzano il sistema operativo installato del computer in cui vengono fatti girare (la maggior parte delle volte sono destinati a Windows).
L'idea alla base è di avere sempre il software di cui abbiamo bisogno, a portata di mano e personalizzato per il nostro utilizzo. Ma "trasportare il desktop ovunque" non è necessariamente il modo migliore per preservare la riservatezza dei dati.
Diciamolo subito: questi software non proteggono le persone che lo utilizzano più di quanto faccia un software non portatile. Peggio ancora, per ragioni di marketing inducono nell'utente false sicurezze, attraverso enormi sciocchezze. L'estratto della seguente frase proviene dalla home page del sito Framakey, una serie di software portatili realizzati da Framasoft, un sito francese di promozione di software libero: "L'uso del software avviene in modo sicuro e senza lasciare alcuna informazione personale sulle macchine in cui si utilizza Framakey" . Questo,

sfortunatamente, non è corretto.
Ci saranno tracce sull'hard disk: se il software è stato reso "portatile" in modo corretto, non dovrebbe lasciare deliberatamente tracce sul disco rigido del computer. Ma in realtà, il software non ha mai il controllo assoluto. Ciò dipende, per la maggior parte delle volte, dal sistema operativo utilizzato, che potrebbe aver avuto necessità di sfruttare la "memoria virtuale" sul disco rigido oppure lasciare diverse tracce nei log o nei "documenti recenti". Tutto ciò rimarrà quindi sull'hard disk.

NON C'È MOTIVO DI FIDARSI DI UN SISTEMA SCONOSCIUTO

Come abbiamo visto molti sistemi operativi non fanno assolutamente nulla di ciò che crediamo e dunque, poiché il software portatile utilizza il sistema installato sul computer su cui viene lanciato, potremmo essere esposti a malware.

NON SAPPIAMO CHI LI ABBIA COMPILATI, NÉ COME

Le modifiche apportate al software per renderlo portatile sono raramente verificate, anche se di solito non sono fatte dagli autori del software stesso. Pertanto, possiamo sospettare che il software possa contenere vulnerabilità di sicurezza, volontarie o no.

Più avanti affronteremo il problema “dell’igiene” minima da tenere quando si sceglie un software da installare o scaricare.

UN MODO PER PROTEGGERSI: LA CRITTOGRAFIA

La crittografia è la branca della matematica che si occupa specificatamente di proteggere i messaggi. Fino alla fine degli anni 90, l'utilizzo di tecniche crittografiche non era concesso al grande pubblico. In molti paesi esso è divenuto legale tra le altre cose, per permettere ai servizi commerciali su Internet di farsi pagare senza che i clienti si facessero rubare il proprio numero di carta di credito.
La crittoanalisi è quella parte che consiste nel "rompere" le tecniche crittografiche, permettendo per esempio di recuperare un messaggio che era stato protetto.

Quando si vuole proteggere un messaggio, si distinguono tre aspetti:
riservatezza: impedire gli sguardi indiscreti;
autenticità: essere sicuri circa l'autore del messaggio;
integrità: essere sicuri che il messaggio non abbia subito modifiche.

Si possono volere tutte e tre queste cose, oppure se ne può volere soltanto una. Una persona che scrive un messaggio confidenziale potrebbe voler negare di esserne l'autore (e quindi non vorrebbe che il messaggio fosse autenticato). Oppure è possibile si voglia certificare la provenienza (autenticare) e l'integrità di una comunicazione ufficiale diffusa pubblicamente (quindi in modo tutt'altro che confidenziale).

In ciascuno di questi casi si parla di messaggi, ma le tecniche crittografiche si applicano di fatto a qualsiasi numero, ovvero a qualsiasi dato, una volta digitalizzato.

E' importante notare che la crittografia non cerca di nascondere i messaggi, ma di proteggerli. Per nascondere dei messaggi, bisogna invece ricorrere a delle tecniche steganografiche (come quelle utilizzate dalle stampanti di cui abbiamo parlato nelle puntate precedenti), che non spiegheremo ora.

PROTEGGERE I DATI DAGLI SGUARDI INDISCRETI

Come sanno bene i bambini che usano delle parole in codice o i militari che si comunicano gli ordini, il metodo più serio affinché dei dati possano essere compresi soltanto dalle persone “all’interno del segreto”, è quello della cifratura.

La cifratura di un file o di un supporto di archiviazione permette di renderlo illeggibile a tutti coloro che non hanno il codice d’accesso (spesso una passphrase). Sarà comunque possibile accedere al contenuto, ma i dati assomiglieranno a una serie di numeri a caso, e saranno quindi illeggibili.
Spesso si dice “criptare e decriptare” invece di “cifrare e decifrare”, il che può risultare confondente; i termini in realtà sono sinonimi.

COME FUNZIONA?

Grosso modo, servono tre grossi concetti per capire come si fa a cifrare un messaggio. (1)

Il primo concetto: la confusione. Si deve ofuscare la relazione tra il messaggio originale e il messaggio cifrato. Un esempio molto semplice è il “Cifrario di Cesare”:

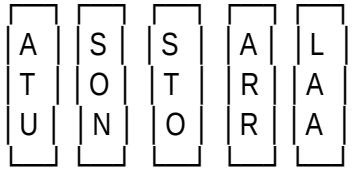
testo in chiaro:
ASSALTO TRA UN' ORA
↓↓↓↓↓↓↓ ↓↓↓ ↓↓ ↓↓↓
testo cifrato:
DVVDOZR ZUD AQ RUD

A + 3 lettere = D

Però nel cifrario di Cesare è facile analizzare la frequenza delle lettere e risalire alle parole.

Quindi si passa al secondo grande concetto: la diffusione. Si sparpaglia il messaggio per renderlo più difficile da riconoscere.

Un esempio di questa tecnica, è la trasposizione per colonne:



ATU SON STO ARR LAA
diffusione in tre punti.

Ciò che viene chiamato un algoritmo di cifratura, sono le diverse tecniche utilizzate per trasformare il testo originale. Riguardo alla chiave di cifratura, si tratta, per esempio nel caso del cifrario di Cesare, del numero che indica di quanti caratteri si deve slittare (nell'esempio: 3), oppure, nella tecnica della diffusione, del numero di linee delle colonne. Il valore di questa chiave è variabile, si può scegliere di fare delle colonne da due linee, o uno slittamento di sei caratteri.

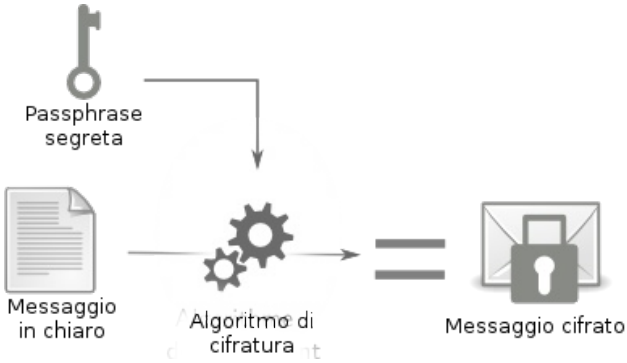
Questo ci porta al terzo grande concetto: il segreto risiede soltanto nella chiave. Dopo qualche millennio, ci siamo accorti che non era una buona idea quella di partire dal principio che nessuno avrebbe capito l'algoritmo di cifratura. Prima o poi qualcuno finirà per scoprirlo… con la forza, se necessario.

Ai giorni nostri, l'algoritmo si può quindi trovare per intero su Wikipedia, dettagliato in lungo e in largo, in modo che chiunque possa verificare che non abbia particolari punti deboli. Questo perchè l'unico modo per decifrare un testo sarà quello di disporre della chiave che è stata usata con quell'algoritmo.

Note:
1) Il passaggio qui di seguito è un adattamento molto parziale del fumetto di Jeff Moser sull'algoritmo AES – http://nbl.gs/qhB

Nel concreto, per assicurarare la riservatezza dei nostri dati, si utilizzano due operazioni: cifrare e poi decifrare.

PRIMO PASSO: CIFRARE



Prendiamo il messaggio seguente per fare un esempio:

Gli spaghetti sono nelll'armadio.

Dopo aver cifrato questo messaggio utilizzando il programma GnuPG con l'algoritmo AES256, e come passphrase "questo è un segreto", si ottiene:

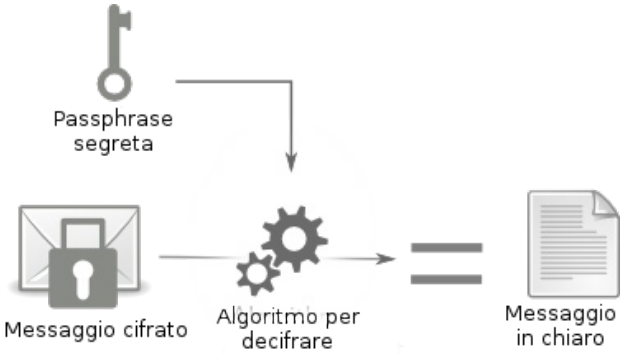
-—BEGINPGPMESSAGE——-

jA0ECQMCRM0lmTSIONRg0lkBWGQI76cQOoc
EvdBhX6BM2AU6aYSPYymSqj8ihFXuwV1GVra
WuwEt4XnLc3F+OxT3EaXINMHdH9oydA92WD
kaqPEnjsWQs/oSCeZ3WXoB9mf9y6jzqozEHw=
==T6eN
-—ENDPGPMESSAGE——-

Questo è quindi l'aspetto che assume un testo dopo la cifratura: il suo contenuto è diventato assolutamente incomprensibile. I dati "in chiaro", leggibili, sono stati trasformati in un altro formato, illeggibile a chi non possiede la chiave.

SECONDO PASSO: DECIFRARE

Per decifrare, ci basterà utilizzare di nuovo GnuPG questa volta sul nostro testo cifrato. GnuPG ci chiederà la passphrase che prima ci era servita per cifrare, e se è giusta, ne otterremo l'informazione che ci mancava per preparare il pranzo.



RIGUARDO ALL'HARD DISK..

Se vogliamo tenere su un supporto di archiviazione (hard disk, penna USB, etc.) soltanto dati cifrati, sarà il sistema operativo che dovrà farsi carico “al volo” di effettuare le operazioni di cifratura e decifratura.
In questo modo, ogni volta che un dato viene letto dall'hard disk, nel passaggio deve essere decifrato in modo che i programmi che ne hanno bisogno possano accedervi. Al contrario, ogni volta che un programma chiederà di scrivere un dato, questo verrà cifrato prima di essere salvato sull'hard disk.

Affinché queste operazioni possano funzionare, è necessario che la chiave di cifratura si trovi nella memoria viva per tutto il tempo che il supporto avrà bisogno di essere utilizzato.

Inoltre, la chiave di cifratura non può essere cambiata. Una volta che è stata usata per cifrare dei dati scritti sul disco, diventa indispensabile

per poterli rileggere. Per poter cambiare la chiave, si deve quindi rileggere e poi riscrivere tutti i dati del disco.

Per evitare questa operazione faticosa, la maggior parte dei sistemi usati per cifrare i supporti di archiviazione si servono di un trucco: la chiave di cifratura in effetti è un grande numero, completamente casuale, che sarà a sua volta cifrato con l'aiuto di una passphrase (1). Questa versione cifrata della chiave viene generalmente scritta sul supporto di archiviazione all'inizio del disco, "in cima" ai dati cifrati.
Con questo sistema, cambiare il codice d'accesso diventa semplice, visto che basterà sostituire soltanto questa intestazione con una nuova.

Note:
(1) Il sistema LUKS, usato da GNU/Linux, permette anche di utilizzare diverse versioni

cifrate della chiave di cifratura. Ciascuna di queste versioni potrà essere cifrata con una passphrase diversa, in modo da permettere a persone diverse di accedere agli stessi dati senza condividere lo stesso segreto.

RIASSUNTO E LIMITI

La crittografia permette insomma di proteggere bene i propri dati (1), cifrando per intero o in parte l'hard disk e gli altri supporti di archiviazione (penna USB, CD, etc) oppure cifrando le comunicazioni. Inoltre i computer moderni sono abbastanza potenti da permetterci di rendere la cifratura un'operazione di routine, invece di riservarla a circostanze speciali o a delle informazioni particolarmente sensibili (che altrimenti verrebbero subito identificate come importanti, e invece sarebbe meglio fossero confuse nella massa).

Si può anche predisporre una passphrase per cifrare l'intero hard disk, e/o dare accesso ad

alcune persone a una parte cifrata con una loro passphrase. Si può anche cifrare singolarmente l'uno o l'altro file, o e-mail, o dividere in parti un file e cifrarle con passphrase differenti.

Nonostante sia uno strumento potente ed essenziale per la sicurezza delle informazioni, la cifratura ha dei limiti – in particolare quando non è utilizzata correttamente.

Come spiegato prima, quando si accede a dei dati cifrati, bisogna avere ben chiare alcune cose. Prima di tutto, una volta che i dati vengono decifrati, essi risiederanno come minimo nella ram. E poi, durante il tempo necessario a cifrare e decifrare, la ram contiene anche la chiave di cifratura.

Chiunque disponga della chiave di cifratura potrà leggere tutto ciò con la quale è stato cifrato, e potrà anche servirsene per cifrare a sua volta dei dati.

Occorre quindi fare attenzione agli elementi qui di seguito:

Il sistema operativo e i programmi hanno accesso ai dati e alla chiave di cifratura tanto quanto noi, per cui si tratta di stabilire quanta fiducia abbiamo in essi. Ancora una volta, si tratta di non installare cose a caso.

Chiunque ottenga un accesso fisico al computer acceso ha, di fatto, accesso al contenuto della RAM. Finché un disco cifrato è attivato, la RAM contiene in chiaro i dati sui quali ha lavorato dall'accensione del computer in poi (anche nel caso in cui questi dati siano cifrati sul disco). Ma sopratutto contiene, come abbiamo detto prima, la chiave di cifratura, che può quindi essere copiata. Quindi è meglio abituarsi, quando non lo si usa, a spegnere il computer e a disattivare (smontare, espellere) i dischi cifrati.
In certi casi, può essere necessario prevedere delle soluzioni materiali per staccare la corrente

facilmente e rapidamente (2); in questo modo i dischi cifrati torneranno inaccessibili senza la passphrase – a meno che non si effettui un cold boot attack.
Anche in questo caso però potrebbe esserci un keylogger installato sul computer, e in questo modo registrerebbe la passphrase.

Infine, può essere saggio ricordare che la matematica utilizzata negli algoritmi crittografici ha talvolta dei difetti. E molto più spesso ancora, i programmi che la applicano hanno delle fragilità. Alcuni di questi problemi possono trasformare, dall'oggi al domani, ciò che pensavamo la migliore delle protezioni in un semplice giochetto..

1) Un articolo di rue89 sulle rivelazioni di Snowden riguardo all'impotenza della NSA verso la critografia. "Marie Gutlub, 2014, Crimes de guerre et décryptage de données : nouvelles révélations de Snowden." – http://nbl.gs/qhE

2) Per questa ragione, è buona norma non lasciare la batteria attaccata a un portatile quando non viene utilizzato. In questo modo per spegnerlo basterà staccare il cavo.

VERIFICARE L'INTEGRITÀ DEI DATI

Abbiamo visto qualche metodo per assicurare la riservatezza dei nostri dati. Però a volte può essere anche importante essere sicuri della loro integrità, ovvero verificare che non abbiano subito modifiche (per sbaglio o di proposito). Possiamo anche volerci accertare della loro provenienza e verificarne l'autenticità.

In concreto, dopo la lettura di queste paginette, ci si può fare un'idea di quanto sia complesso essere sicuri che i programmi che installiamo sul nostro computer non siano stati modificati in corso d'opera per introdurre dei software malevoli.

LA POTENZA DELL`ASCIA (1)

Le più importanti tecniche di verifica di integrità o di autenticità, si fondano su degli strumenti matematici che la crittografia chiama “funzioni di hash”.

Questi strumenti sono come delle grosse mannaie capaci di ridurre tutto in piccoli pezzi. Sappiamo che la nostra mannaia funziona abbastanza bene da poter essere utilizzata in crittografia, se siamo in grado di dire che:

- avendo i piccoli pezzettini, è impossibile ricostruire l'oggetto originale senza provare tutti gli oggetti della terra;
- lo stesso oggetto, ogni volta che lo passiamo alla mannaia, darà sempre gli stessi piccoli pezzettini;
- due oggetti differenti devono dare pezzettini differenti.

Se queste tre proprietà sono verificate, ci basterà allora confrontare i piccoli pezzi derivati da due oggetti per sapere se quest'ultimi sono identici.

I piccoli pezzetti che vengono fuori dalla nostra mannaia si chiamano più comunemente "somma di controllo" o "impronta", e viene in genere rappresentata sotto forma di qualcosa di simile a questo:

f9f5a68a721e3d10baca4d9751bb27f0ac35c7ba

Visto che la nostra mannaia funziona con dati di qualunque dimensione e forma, confrontare le impronte ci permette di confrontare più facilmente immagini, CD, software, etc.

Però la nostra mannaia non è magica. Facciamo finta che invece a un certo punto riduca una cosa in piccoli cubetti di taglia identica: ci potremmo ritrovare con gli stessi cubetti usciti da due oggetti differenti. Questo si chiama "collisione". Questa

carambola matematica è fortunatamente pericolosa solo quando è possibile provocarla… cosa già successa per molte funzioni di hash dopo qualche anno di ricerca, in particolare per la funzione SHA1 (2).

Note:
1) I francesi traducono gran parte dei termini tecnici inglesi. Così è anche per “hash”, che in italiano lasciamo in inglese, e che in francese diventa invece “hache" (ascia) in riferimento al concetto del tagliare, estrapolare un pezzo dal codice. (NDT)

2) Marc Stevens e Al., 2017, Announcing the first SHA1 collision, Google Security Blog. – http://nbl.gs/qhG

VERIFICARE L'INTEGRITÀ DI UN SOFTWARE

Facciamo un esempio: Alice ha scritto un programma e lo distribuisce in CD che si possono trovare tra le associazioni di utilizzatori di GNU/Linux. Betty vorrebbe utilizzare il programma di Alice, ma pensa che sarebbe molto facile per un amministratore malintenzionato rimpiazzare uno dei CD di Alice con un software malevolo.

Betty non può andare direttamente da Alice a prendere un CD, perchè Alice abita in un'altra città. Però l'ha incontrata qualche tempo fa e sa riconoscere la sua voce. Quindi le telefona e Alice le detta la somma di controllo del contenuto del CD:

CD di Alice

94d93910609f65475a189d178ca6a45f
22b50c95416affb1d8feb125dc3069d0

Betty allora può confrontarla con quella generata a partire dal CD che si è procurata:

CD di Betty
94d93910609f65475a189d178ca6a45f
22b50c95416affb1d8feb125dc3069d0

Siccome i numeri sono gli stessi, Betty è contenta e si sente sicura che sta utilizzando lo stesso CD che ha fatto Alice.

Calcolare queste somme di controllo non richiede molto più tempo che non la lettura completa del CD, giusto qualche minuto in più.

Invece adesso mettiamoci nei panni di Carole, che è stata pagata per prendere il controllo del computer di Betty a sua insaputa. Per fare questo ha creato un CD che sembra quello di Alice, ma che invece contiene un software malevolo.

Purtroppo per lei, l'hash funziona solo in un verso. Si deve quindi per prima cosa procurare il CD originale di Alice.

A quel punto, modifica questo CD introducendoci il software malevolo. Questa prima versione assomiglia molto all'originale e potrebbe ingannare più di una persona, ma lei sa che Betty verificherà la somma di controllo del CD.

Siccome Alice utlizza la funzione hash SHA256, che non ha difetti noti, a Carole non resta che provare un gran numero di varianti del suo CD, nella speranza di ottenere una collisione, ovvero la stessa somma di controllo di quella di Alice.

Sfortunatamente per lei, e fortunatamente per Betty, anche disponendo di molti computer potenti, le possibilità di riuscita di Carole in un tempo ragionevole (diciamo qualche anno) sono estremamente basse.

Per verificare l'integrità dei dati, basta insomma procurarsi un'impronta, o una somma di controllo, da degli intermediari di fiducia. Tutto sta poi a procurarsi queste impronte attraverso un mezzo di fiducia, ovvero di essere in grado di verificare la loro autenticità..

VERIFICARE UNA PASSWORD

Un altro esempio di utilizzo delle funzioni di hash riguarda la verifica dell'autenticità di una richiesta d'accesso.

Se l'accesso a un computer è protetto da password, come per esempio durante l'apertura di una sessione sotto GNU/Linux (ma ricordiamoci sempre che questa password non protegge i dati!), bisogna fare in modo che il computer possa verificare che la password che abbiamo messo sia quella giusta. Le password però non sono salvate in chiaro sul computer, sennò sarebbe troppo facile impadronirsene.

Allora come fa il computer a essere certo che la password immessa sia esatta?

Quando abbiamo scelto una password sul nostro computer, il sistema ha salvato, grazie a una funzione di hash, un'impronta della password. Per verificare l'accesso, lui "spezzetta" alla stessa maniera la password che abbiamo immesso. Se le impronte sono le stesse, decide che la password è giusta.

E' insomma possibile verificare che le password corrispondano senza custodire la stessa password.

SIMMETRICA O ASIMMETRICA?

Le tecniche di cifratura menzionate fin qui si basano su di una sola chiave segreta, che permette si effettuare sia la cifratura che la decifratura. In questo caso si parla di "crittografia simmetrica".

Questo per contrapporla alla crittografia asimmetrica, che non utilizza la stessa chiave per entrambe le azioni. Anche detta “cifratura a chiave pubblica”, quest’ultima viene usata sopratutto per la comunicazione “online”, di cui parleremo nel dettaglio nelle prossime puntate.
Una delle proprietà più interessanti della crittografia asimmetrica che può essere evocata brevemente, è la possibilità di realizzare delle firme digitali. Come il suo equivalente sulla carta, una firma digitale permette di apporre un marchio di riconoscimento sui dati.
Queste firme digitali che utilizzano la crittografia asimmetrica, costituiscono il modo più semplice per verificare la provenienza di un software. Nei prossimi capitoli di questa guida vedremo come servircene.

Nel prossimo numero:
Scegliere le risposte adatte - Valutazione dei rischi - Definire una policy di sicurezza...

Quello che avete tra le mani è il terzo numero della traduzione a puntate della Guide d'autodéfence numerique.
L'edizione originale integrale (in francese) è leggibile online e scaricabile liberamente qui:
http://guide.boum.org
Trovate invece le puntate precedenti della traduzione in italiano qui:
http://numerique.noblogs.org