Don't trust in Technology

Consigli per la privacy e sicurezza

INDICE:

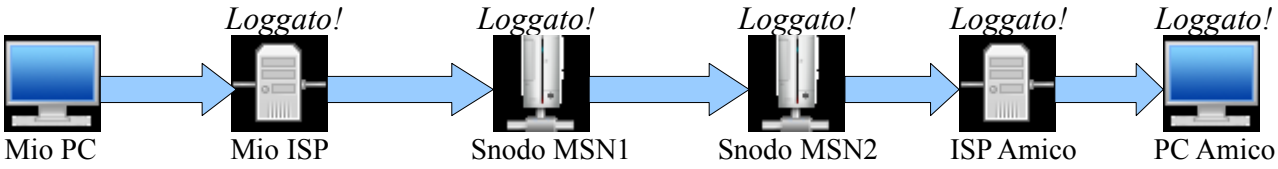
Original source: www.ihteam.net/DontTrustInTech.pdf

1.0 – INTRODUZIONE:

Vorrei cominciare con lo spiegarvi le motivazioni di questo titolo provocatorio “Don't trust in Technology” (non fidarsi della tecnologia). Da qualche anno a questa parte il computer è diventato un “amico” a cui confidiamo ciò che si pensa, i nostri dati personali, codici di carte di credito, codici bancari e chi più ne ha più ne metta! Il punto è che quando ce ne andiamo a dormire, quei dati rimangono li', a disposizione di chiunque abbia un minimo di capacità o di potere. Tutte le attività che abbiamo svolto online (chattato, navigato, visto video...) vengono registrate all'interno di più server sparsi nel mondo. Con questa guida vorrei farvi aprire gli occhi e cercare di proteggere la vostra privacy.

2.0 – PRIVACY NELL'ERA MODERNA:

Dall'introduzione cerchiamo ora di capire chi sono questi server che hanno i nostri dati e soprattutto come mai li hanno! Tutto nasce da un pensiero comune:
“OK, io metto a disposizione la mia struttura, ma nel caso in cui qualcuno voglia fare il furbetto io devo sapere chi è stato, così, se mi denunciano, so a chi dare la colpa”
Questa è la filosofia di ¾ degli ISP Internet. Da Telecom Italia alla NSA. Per questo motivo tutto quello che facciamo online è TRACCIATO... Scritto su file di testo, pronto ad essere tirato fuori in caso di necessità. Facciamo un esempio con Messenger:
Mi devo incontrare con un mio amico a casa sua.
Mi connetto su MSN, contatto il mio amico dicendogli “Ci vediamo fra poco a casa tua”.
Chiudo la conversazione, chiudo messenger e parto da casa per andare dal mio amico.
Ok, niente di speciale, ma pensiamoci su... Da quel momento in poi circa 3-4 server sparsi nel mondo hanno immagazzinato l'informazione, 3-4 server sanno che mi stò per incontrare con il mio amico. Mentre io sono convinto che il messaggio sia arrivato solamente al suo destinatario.



Finchè sia un computer a “sapere” che mi sto per incontrare con il mio amico me ne frega poco, ma il problema si pone quando un potenziale ficcanaso voglia controllare la mia attività. In questo breve esempio l'attaccante ha ben 5 obiettivi dal quale può reperire l'informazione che gli interessa! E non c'è neanche bisogno che la prende immediatamente, tanto i dati rimangono registrati per anni!

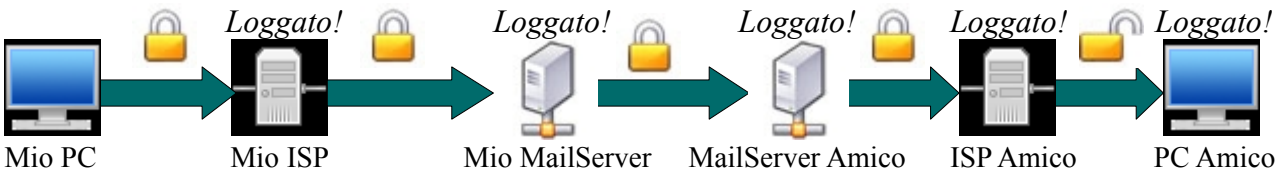
2.1 – CRITTOGRAFIA:

Tutto quello che possiamo fare per difendere la nostra privacy gira attorno a questa parolina magica: “Crittografia”.
Cominciarono ad utilizzarla gli ebrei, gli spartani, Giulio Cesare... Fino ad arrivare ai giorni nostri.
Per definizione (da Wikipedia): *La crittografia tratta delle "scritture nascoste", ovvero dei metodi per rendere un messaggio "offuscato" in modo da non essere comprensibile a persone non autorizzate a leggerlo.*
Esistono vari tipi, ma il più utilizzato nell'informatica è la crittografia asimmetrica o crittografia a chiave pubblica.
Il funzionamento è semplice: Entrambi gli interlucutori hanno due chiavi, una personale ed una che si scambiano.
L'idea base della crittografia asimmetrica diviene più chiara se si usa un'analogia postale, in cui il mittente è Alice ed il destinatario Bob e i lucchetti fanno le veci delle chiavi pubbliche e le chiavi recitano la parte delle chiavi private:

- Alice chiede a Bob di spedirle il suo lucchetto, già aperto. La chiave dello stesso verrà però gelosamente conservata da Bob.
- Alice riceve il lucchetto e, con esso, chiude il pacco e lo spedisce a Bob.
- Bob riceve il pacco e può aprirlo con la chiave di cui è l'unico proprietario.

Con la Crittografia noi possiamo cifrare qualsiasi tipo di informazione!
Per le e-mail potremmo utilizzare Gnu Privacy Guard (GPG, http://www.gnupg.org/) che è uno dei programmi più usati. Utilizza uno standard molto diffuso (RFC 440), è OpenSource, gratuito e multipiattaforma. E’ compatibile con moltissimi client di posta.

Facendo un esempio di invio mail cifrata:



Possiamo notare che la comunicazione viene comunque tracciata, ma essendo “chiusa dal lucchetto” è incomprensibile. Solo il destinatario (PC Amico) ha la facoltà di leggere il vero contenuto. Ora l'utente ficcanaso può solamente capire che io ho inviato una mail, ma sarà impossibilitato nel leggere il contenuto.
Esistono protocolli che permettono la comunicazione crittografata. Fra i piu famosi:

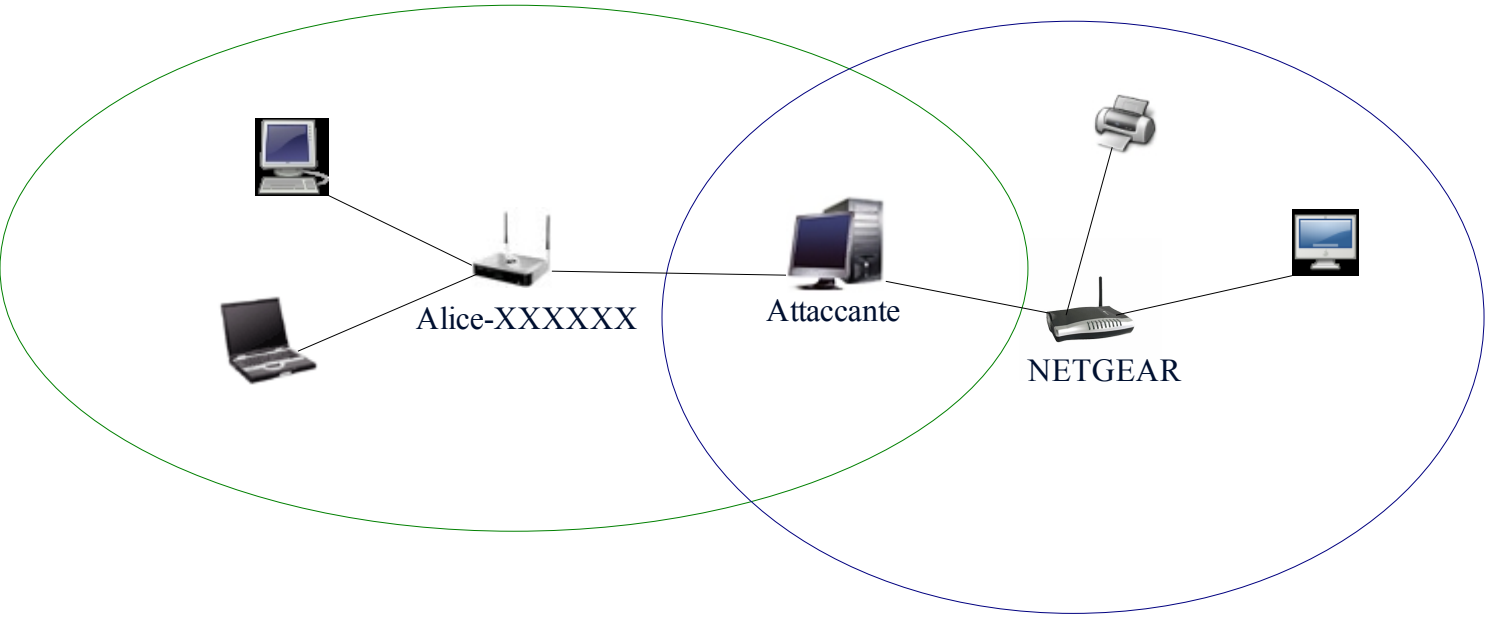
http -> https
pop3 -> pop3s
imap -> imaps
smtp -> smtps
telnet -> ssh
ftp -> sftp

2.2 – WIRELESS:

Mi piace definire il Wireless come la tecnologia più insicura del momento (insieme al bluetooth =P). Dobbiamo immaginare gli AccessPoint WiFi come dei circuiti che sparano ininterrottamente nell'etere dei segnali carpibili da chiunque all'interno del loro raggio d'azione. Cominciamo con un po' di storia e di nozioni base:
Il Wireless può lavorare a 2,4 o 5,0GHz ed inizialmente non era prevista una codifica delle informazioni che passavano, in questo modo però chiunque rientrasse nel raggio d'azione dell'AP poteva veder passare il traffico in chiaro. Successivamente vennero introdotti i primi standard di crittografia per il wireless (vedi 2.1): il WEP (Wired Equivalent Privacy). Ma anche quest'ultimo si rivelò un fallimento sotto l'aspetto della sicurezza. Infine, per perfezionare il tutto, nacquero il WPA e WPA2 (Wi-Fi Protected Access). La differenza è che il WPA utilizza una chiave condivisa (PSK) per l'autentificazione, mentre il WPA2 sfrutta avanzati metodi di crittografia come AES e CCMP.

Codifica	Algoritmo cifratura	Bit cifratura	Sicurezza
[Nessuna]	-	-	INSICURA
WEP	RC4	40bit, 64bit, 104bit	INSICURA
WPA	RC4, PSK	128bit	POCO SICURA
WPA2	AES, CCMP	256bit	MEDIAMENTE SICURA

Le prime 3 categorie (nessuna codifica, wep e wpa) consentono al solito utente ficcanaso di entrare nella vostra rete (nel caso WPA solo se avete scelto una passphrase piu corta di 6-7 caratteri o parole contenute in un dizionario), mentre in WPA2 (e logicamente anche su tutte le precedenti) può disturbare la vostra connessione facendovi saltare il collegamento fra client e AP ogni qualvolta che vuole.
Direte... “Ma se è una tecnologia così insicura, perchè starebbe in commercio?”.
A questa domanda non so rispondervi... Probabilmente perchè alle persone fa' comodo avere una connessione Internet anche quando stanno in bagno.
Di seguito è presentato uno scenario tipo dell'utilizzo di connessioni Wireless:

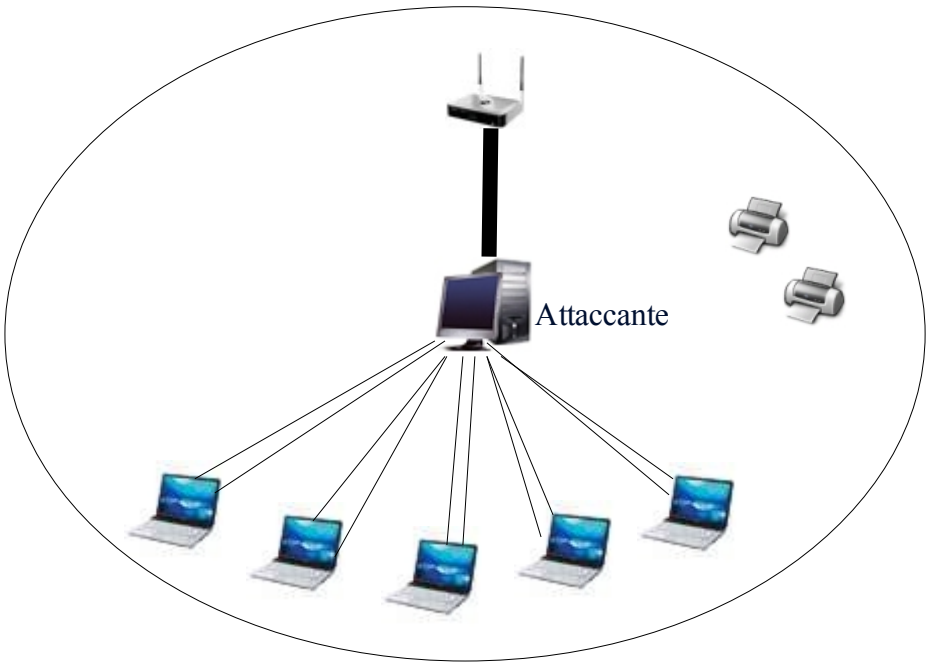


2.2.1 – ARP SPOOFING:

Dopo questa breve introduzione alle insicurezze del WiFi torniamo sull'argomento privacy, introducendo la tecnica hacking prediletta che si utilizza spesso sulle reti Wireless: L'ARP Spoofing o ARP poisoning. Definendolo (da wikipedia): *L'ARP poisoning (detto anche ARP spoofing) è una tecnica di hacking che consente ad un attacker, in una switched lan, di concretizzare un attacco di tipo man in the middle verso tutte le macchine che si trovano nello stesso segmento di rete. L'ARP poisoning è oggi la principale tecnica di attacco alle lan commutate. Consiste nell'inviare intenzionalmente e in modo forzato risposte ARP contenenti dati inesatti o, meglio, non corrispondenti a quelli reali. In questo modo la tabella ARP (ARP entry cache) di un host conterrà dati alterati (da qui i termini poisoning, letteralmente avvelenamento e spoofing, raggiro). Molto spesso lo scopo di questo tipo di attacco è quello di redirigere, in una rete commutata, i pacchetti destinati ad un host verso un altro al fine di leggere il contenuto di questi per catturare le password che in alcuni protocolli viaggiano in chiaro.*
Ora come possiamo difenderci?

- Il modo migliore sarebbe quello di non collegarsi a reti Wireless pubbliche (Università, Bar, Hotel ecc);
- Se proprio ne avete necessità ci sono dei programmi fatti ad hoc per esaminare le attività di rete ed evidenziarne delle discrepanze come: arpwatch (http://freequaos.host.sk/arpwatch/) o Snort (http://www.snort.org/);
- Se utilizzate un apparato WiFi a casa o al lavoro cambiate immediatamente tipologia di codifica a WPA2/AES (la meno peggio); Se la vostra scheda WiFi del computer client non supporta questo tipo di codifica cercate un aggiornamento driver o firmware sul sito della casa produttrice.

Di seguito un esempio di ARP Spoofing in esecuzione:

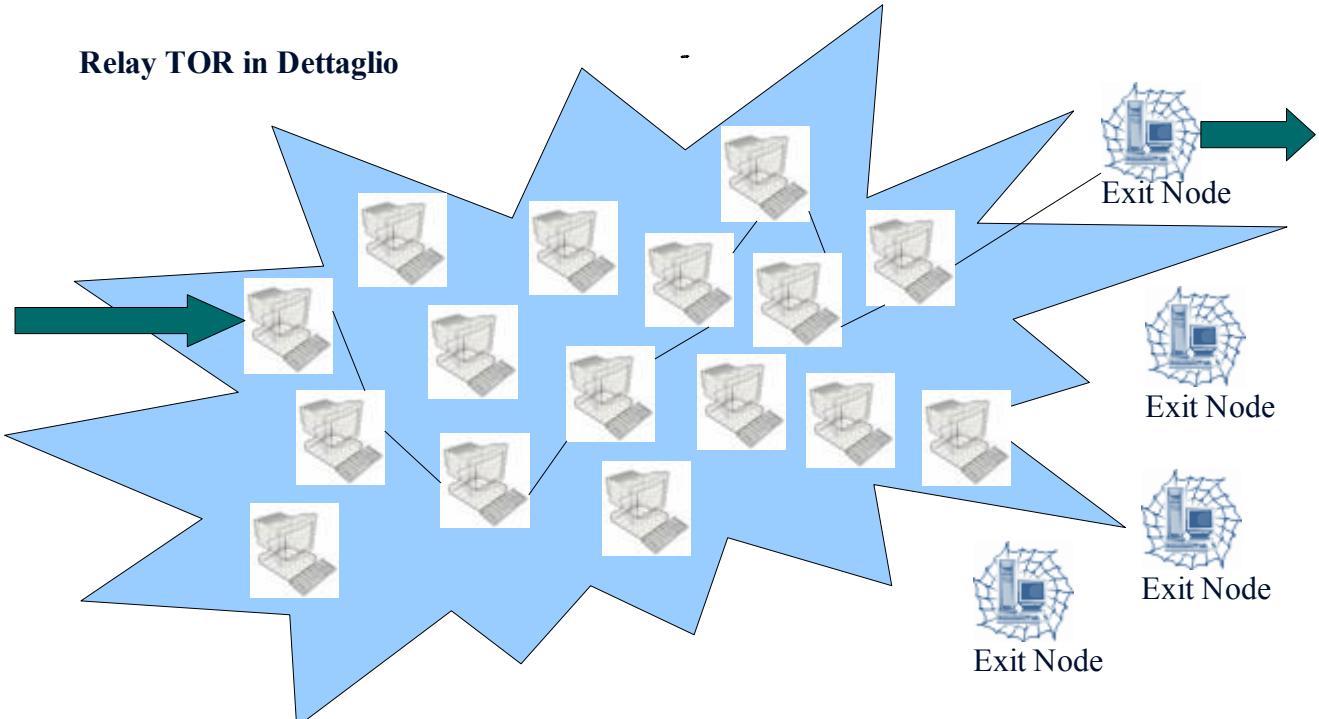


2.3 - PRIVACY SU INTERNET:

Cerchiamo ora di allargare i nostri orizzonti e pensare alla privacy su tutta la rete internet. Come nell'esempio del cap. 2.0 anche quando navighiamo succede la stessa cosa: si lasciano i dati un po qua e là per il mondo.
I problemi da risolvere in questo caso sono 2:

- Come faccio a non far passare i dati in chiaro?
- Come faccio a non far capire al destinatario che il traffico proviene da me?

Per prendere due piccioni con una fava possiamo utilizzare TOR (The Onion Router), da Wikipedia: *è un sistema di comunicazione anonima per internet basato sulla seconda generazione del protocollo di onion routing. Tor protegge gli utenti dall'analisi del traffico attraverso una rete di onion router (detti anche relay), gestiti da volontari, che permettono il traffico anonimo in uscita e la realizzazione di servizi anonimi nascosti.*



Questo incasinato disegno che ho fatto è il funzionamento della rete TOR. Detto a parole: quando inviamo una richiesta usando la Onion Router criptiamo immediatamente le informazioni e le affidiamo ai relay TOR che hanno il compito di farla saltare da un serve all'altro in modo da "incasinare" le tracce fin quando non viene inviata all'"Exit Node" che ha lo scopo di decodificare l'informazione e mandarla in chiro al nostro destinatario (Pc Amico) che riceverà l'informazione come se fosse stato l'exit node ad inviargliela per primo e non noi. Tutti questi server (Relay ed Exit node) sono sparsi nel mondo, quindi prima di inviare un'informazione, ci può volere del tempo, ecco perchè spesso questa rete è lenta.
Avete visto però dove stanno 2 grandi problemi?

1. Spesso i Relay Tor registrano le attività svolte e ripercorrere la catena all'indietro è possibile (accedendo fisicamente ad ogni relay ed esaminando i log);
2. All'Exit Node arrivano i dati in chiaro, lui sa cosa c'è nell'informazione, ma non da chi è stata inviata originariamente.

Fortunatamente in una sessione di scambio informazioni le catene ed exit node cambiano continuamente e raccapezzarci sarebbe difficile!
La rete TOR può essere utilizzata in tutti gli ambiti di internet, dalla navigazione alla messagistica istantanea (IRC, MSN, Skype ecc..).

2.4 – TELEFONATE:

E' un argomento attuale quello delle intercettazioni, degli spionaggi industriali e dei soliti spioni che non si fanno gli affari loro. Tra i campi preferiti di queste persone ci sono i telefoni. C'è anche qui il modo di godersi serenamente una telefona stando tranquilli che nessuno sia in ascolto.
Il miglior modo è sicuramente quello di utilizzare il VoIP criptato, un esempio famosissimo è Skype. Tuttavia non conosciamo il funzionamento del programma, chi lo può controllare. Le nostre informazioni sono conservate su un server di cui non abbiamo alcun controllo.
La soluzione è sempre quella: appoggiarsi a strumenti OpenSource di cui conosciamo il funzionamento (es. OpenWengo). Utilizzare carrier di cui abbiamo fiducia e che permettono di utilizzare tecnologie aperte e messe in sicurezza (es. SIP con TLS ed SRTP).

3.0 - SICUREZZA NELL'ERA MODERNA:

Con questo altro capitolo introduciamo la sicurezza dei nostri PC a livello locale. Metodi che sicuramente avrete gia risentito centinaia di volte, ma che ancora ¾ degli utenti medi non ascoltano ancora! Andremo ad analizzare tre principali argomenti:

- Backup
- Password
- Crittografia Locale

Gli attacchi piu semplici sono quelli che si effettuano quando si ha accesso fisico ad una macchina, per questo motivo anche la sicurezza locale è molto importante. A questo punto siamo capaci di proteggere la nostra privacy online, vediamo come affrontare quella in locale!

3.1 – BACKUP:

Banale, ma è giusto dare importanza ai backup! In questa parte impareremo come automatizzare al meglio il compito di backup del nostro PC su un sistema Linux (per windows si può usare la stessa logica);
Il backup si pianifica in base all'utilizzo, alle esigenze e alla tipologia di dati da backuppare.
Lo script che sto per proporvi è nato dall'esigenza di fare il backup di uno spazio web + un database mysql di 33GB in continua espansione. Quindi ho impostato una frequenza di backup di 24h e per sicurezza una seconda copia mensile in un'altra partizione dell'HDD.
Per impostare la frequenza utilizziamo su Linux il crontab; Per motivi di tempo non posso spiegarvi l'utilizzo di cron, vi consiglio di consultare il man per aiuto.
Prima di tutto creiamo lo script necessario al backup dei file:

```
giornaliero.sh
#!/bin/sh
rm -vf $HOME/backup/bk_files_*.tgz 'Rimuovo backup giorno precedente
tar -cvzpf $HOME/backup/bk_files_.tgz /var/www/ihteam.net/httpdocs/ 'Creo il
nuovo archivio prendendo tutti i file della httpdocs
```

Passiamo ora a vedere lo script per il backup del database MySql. Per far ciò utilizzeremo il comando mysqldump che stampa a video i dati presenti all'interno di un database (o di tutti in questo caso);

```
giornaliero_sql.sh
#!/bin/sh
rm -vf $HOME/backup/giornaliero_sql_* 'Rimuovo precedenti backup
mysqldump -uroot -p"sql_password" --all-databases >
$HOME/backup/giornaliero_sql.sql 'Prendo i risultati e li metto su file
gzip $HOME/backup/giornaliero_sql.sql 'Comprimo il tutto risparmiando spazio
rm -vf $HOME/backup/giornaliero_sql.sql 'Rimuovo il file non compresso
```

Ora possiamo passare a dare il compito di eseguire questi script a cron ogni giorno ad un'ora che preferite (magari in un momento che c'è meno traffico sul vostro sito). Date circa 2 ore di distanza fra l'avvio del primo script e del secondo per evitare che il PC si sovraccarichi di lavoro.
Esempio:

*30 3 * * * $HOME/backup/./script_giornaliero_sql.sh*

In questo esempio ogni giorno alle 3.30AM esegue quel comando.
A questo punto lasciare i dati sull'HD è inutile in caso di rottura del disco, quindi ogni mese bisogna avere l'accortezza di masterizzare i file su DVD e anche su un altro HD esterno per avere maggiore sicurezza.

3.2 – PASSWORDS:

L'argomento è stato ripreso moltissime volte, ma sarebbe meglio aggiornarvi sulle ultime “novità” riguardanti il cracking delle password per capire meglio che tipi di password sono da evitae e quali da utilizzare.
Da un sacco di tempo girano in rete le Rainbow Tables. Da poco sono uscite le Rainbow Tables Ibride che utilizzano un meccanismo di combinazione intelligente. Vi riporto alcune informazioni dal sito freerainbowtables.com dal quale è possibile scaricare queste tabelle per:
LM:
lm_alpha-numeric#1-7 (4 table(s), 99.9% total success probability)
lm_all-space#1-7 (4 table(s), 99.9175% total success probability)
(Il che vuol dire che qualsiasi password sotto i sette caratteri, anche con caratteri speciali è innocua)
NTLM:
ntlm_alpha-space#1-9 (4 table(s), 99.92% total success probability)
ntlm_mixalpha-numeric-all-space#1-6 (4 table(s), 99.9554% total success probability)
ntlm_loweralpha-space#1-9 (4 table(s), 99.9323% total success probability)
ntlm_loweralpha-numeric-space#1-8 (4 table(s), 99.9304% total success probability)
[….]

(Il che vuol dire che password sotto una media di 9 caratteri, anche caratteri speciali, sono innocue)
Esistono poi RainbowTables per altri formati come l'MD5 e SHA1.
Un'altra arma per il cracking sono le GPU (potenza di calcolo che sfruttano le schede Video) che con una 9800gtx da 512MB di cache arriva a fare 390MILIONI di combinazioni al secondo (la velocità dipende anche dalla codifica dell'hash). Questo diminiusce di ben 3 volte i tempi di bruteforce.
Attenzione quindi a non inserire password presenti su dizionario, aggiungete sempre caratteri speciali e assicuratevi che la password sia lunga un minimo di 12-15 caratteri.
Attenzione a tenere le vostre password scritte su post-it in giro per ufficio/casa.

3.3 – CRITTOGRAFIA LOCALE:

Come dicevo nell'introduzione alla sicurezza locale, gli attacchi piu completi vengono effettuati tramite accesso fisico alla macchina. Ricordatevi che se qualcuno ha accesso alla vostra macchina le password sono vane, perchè c'è sempre un modo per bypassarle, sia su Linux che su Windows. Per questo l'unica soluzione è la crittografia del disco.
Come la crittografia nel capitolo 2.1, anche quella per cifrare gli HardDisk rende il tutto illegibile senza l'univoca chiave del proprietario (in questo caso una sola password). Esistono programmi appositi che fanno questo lavoro come TrueCrypt (sia per Windows che per Linux) che permette di generare volumi cifrati sia da un intero drive fisico o da una partizione, oppure trasformando un normale file in un disco virtuale cifrato. Tutti i dati che passano da e per il volume sono codificati e decodificati on-the-fly secondo la procedura di crittografia scelta (256-bit AES, Serpent, Twofish o loro combinazioni) o algoritmi hash (Whirlpool, RIPEMD-160 e SHA-1).

4.0 – CONCLUSIONI:

Abbiamo fatto un excursius sulle principali problematiche di privacy e di sicurezza dei nostri tempi, spero di avervi insegnato qualcosa in più di quello che già sapevate. Questo articolo non deve essere preso come spunto per “scappare dalle forze dell'ordine”, o per mettere su un'attività illegale con i fiocchi bensì deve essere preso per quello che è: condivisione di informazione! Chiunque può usare le proprie conoscenze per fare del bene o per fare del male, sarà solo responsabilità di chi compie queste azioni! Se c'è qualche dubbio o domande da farmi potete contattarmi sul sito: www.ihteam.net

5.0 – ABOUT ME:

Io sono R00T[ATI] della ihteam.net. Lavoro come consulente di sicurezza e webmaster persso un'azienda della mia città. Potete contattarmi tramite mail a r00t.ati[at]gmail[dot]com