2. I sistemi Windows e quelli della famiglia Linux sono basati su modelli di sicurezza assai differenti. Illustrare le difformità filosofiche e implementative relative alla crittografia del file system nei sistemi Windows e Linux, sia dal punto di vista del file system stesso (nativo) che da quello di eventuali software (add-on) di terze parti.

I sistemi operativi windows e unix-like sono notevolmente differenti. Il primo è closed source, l'altro invece è open source. Ma questa non è solo l'unica differenza presente, anzi tra di loro si differenziano in molte cose e una di queste è la crittografia del file system. La crittografia del file system permette di avere un ulteriore grado di sicurezza, riuscendo ad effettuare una protezione ulteriore dei dati come ad esempio tenerli fuori dalla portata di occhi indiscreti. Nei sistemi windows vi possono essere vari metodi di crittografia del file system uno di questi può essere IFS (Installable File System). IFS è stata la prima risposta di Windows al sistema Unix sulla crittografia, l'IFS offre un livello di isolamento molto alto ma è anche molto complesso da utilizzare. Ultimamente però sopra IFS è possibile disporre di un'interfaccia assolutamente identica al metodo sviluppato da UNIX il che rende semplice anche il "porting" di altri file system su Windows.
Nei sistemi Unix invece esiste, come metodo di crittazione il VFS (Virtual File System). Il VFS è un layer di dal mondo UNIX e ultimamente disponibile anche in Windows. Come abbiamo visto sopra il sistema Windows ovvero l IFS, dispone della stesa interfaccia di VFS. Lo scopo di quest'ultimo è quello di rendere uniforme l'interfaccia esterna (verso il S.O.) di ogni file system confinando le differenze logiche e funzionali all'interno del file system driver.
Esistono anche degli applicativi che permettono la crittografia dei file system. Ad esempio c'è Loop-AES Questo metodo si basa su un' applicazione che legge da stdin e scrive su stdout chiamata aespipe. Questa applicazione è dunque perfetta per essere usata in "pipe" per la cifratura/decifratura in tempo reale di file "comuni" e anche di intere partizioni e file system.

3. Il DLGS 196/2003 ha un impatto significativo sulla gestione dell'information security, specialmente riguardo alcune attività sistemistiche tipiche del troubleshooting. Esaminare le interferenze tra la legge e l'attività del sistemista che, da un lato deve garantire a se stesso e agli altri la privacy, dall'altro lato deve poter svolgere efficacemente il proprio lavoro.

Il DLGS 196/2003, ovvero la famosa legge sulla privacy in vigore in Italia, impone notevoli sforzi sai logici che pratici, al sistemista informatico soprattutto se specializzato in sicurezza. In breve la legge indica che è considerato reato, qualsiasi informazione acquisita abusivamente, con o senza tecniche informatiche, come l'analisi dei log dei router aziendali o l'analisi dei messaggi di posta elettronica, che possa fornire informazioni sensibili sulla persona o meglio sul dipendente di una ditta, come ad esempio l' orientamento religioso o sessuale oppure le visite di siti web non attinenti allo specifico lavoro del dipendente, o comunque che non portino un interesse per l'azienda. Il sistemista pertanto con la legge 196/2003 si ritrova in notevole difficoltà, infatti ha l'obbligo, essendo stato assunto dall'azienda, di assicurare la protezione e la sicurezza della rete aziendale dall' primo all'ultimo nodo e potrebbe incappare in problemi che la stessa azienda gli presenta del tipo " Tizio frutta poco ma esce tardi dall'ufficio, e richiede sempre gli straordinari come mai?" Dopo una breve analisi il sistemista si potrebbe

accorgere, ad esempio che il dipendente negligente, in realtà utilizza lo straordinario per visitare siti internet e/o spedire email (qui non è importante tanto il contenuto ma proprio a chi viene spedita) che non hanno assolutamente attinenza al mondo del lavoro. Purtroppo già quest' operazione ha fatto si che il sistemista violasse il DLGS 196/2003, visto che ha effettuato operazioni di auditing e log analisys scoprendo, anche se involontariamente, cosa e quando ha svolto quelle operazioni, pertanto le prove portate eventualmente in tribunale risulterebbero nulle , visto che la legge non considera prove che sono state prese abusivamente, anzi a sua volta il sistemista potrebbe essere denunciato.
In fine l'unica soluzione per il sistemista sono delle precauzioni che gli permettano di evitare il più possibile di cozzare con la legge in questione. La prima cosa che un sistemista dovrebbe fare, prima ancora di mettersi a lavoro, sarebbe quello di pensare, facendosi uno schema mentale su cosa il DLGS 196/03 gli permette di fare e cosa invece gli è vietato. In questo modo il sistemista si mette già al riparo da possibili denunce sulla violazione della privacy. Inoltre nel caso fosse necessario proprio violare la privacy, il gestore della rete potrebbe porsi la domanda: "Ma è proprio necessario farlo? Il dipendente ha commesso un reato così grave?". Se la risposta è affermativa potrebbe autodenunciarsi alle forze dell'ordine, dichiarando di aver violato la privacy ma si pone nello stesso momento di poter validare le prve recuperate sul dipendente indisciplinato, altrimenti se la risposta è negativa potrebbe attuare delle serie di precauzioni affichè non avvengano più alcun tipo di violazione . Innanzitutto un ottimo sistemista potrebbe far firmare a ogni dipendente una liberatoria sulla privacy, che informa ai dipendenti della compagnia che in quell' azienda sono presenti tecniche di controllo avanzato per garantire la sicurezza informatica e non, dell'azienda. Successivamente potrebbe ridurre i possibili accessi a siti web inserendo nel proprio proxy o firewall, nell'apposita white list, gli unici siti web autorizzati dalla stessa compagnia ad essere consultati essendo stati ritenuti validi e di interesse per l'azienda. Facendo ciò il sistemista di una rete aziendale ha rispettato la legge 196/03.

4. Quali sono le vulnerabilità più frequentemente utilizzate da virus, trojan e malware in genere? Accennare brevemente all'evoluzione storica dei virus, di che tipo erano i virus di ieri? E come saranno quelli di domani? Su cosa si basano e chi è il loro target?

I virus informatici, hanno come principale scopo, proprio come i virus che colpiscono gli essere umani, quello di infettare e recare il maggior danno possibile al computer vittima, che a sua volta infetterà un altro pc. Esistono ormai un infinità di virus e ogni giorno ne vengono creati un infinità, ed è per questo motivo che una rete o un singolo desktop, non può mai considerasi sicuro. I virus, come abbiamo detto sono tantissimi ma possono essere catalogati in opportune categorie in base alla loro tipologia di infezione. Esistono

Virus Polimorfi: Ivirus polimorfi permettono di mutare la loro disposizione di dati e codice sorgente ad ogni infezione di un PC

Stealth Virus: Gli stealth virus prendono il controllo delle funzioni di sistema e ad esempio leggono e scrivono i file. Quando l'antivirus andrà a leggere determinati file di sistema, modificati dal virus, il virus stesso risponderà con il vecchio contenuto del file in modo da ingannare l'antivirus che crederà che il file non sia stato modificato.

Fast Infectors: I Fast Infectors sono virus che contano tutto sulla rapidità della diffusione dell'infezione. Essi cercano di attaccare quanti più PC possibili. Infatti le

software house produttrici di antivirus sono in grado di rilasciare gli update necessari per debellare il virus dopo qualche ora dalla scoperta del virus.

Armored Virus:Virus difficili da disassemblare, il di assemblaggio dei virus è importantissimo perché permette ai produttori di SWH di capire come funziona il virus.

Virus "tappabuchi": Questo tipo di virus tenta di infettare i file eseguibili senza modificarne le dimensioni. Infatti un file eseguibile contiene svariate parti "sostituibili" senza comprometterne l'eseguibilità.

Tunnelling Virus: Tentano di sfruttare dei tunnel attraverso gli antivirus e i personal firewall per accedere al pc vittima.

Virus cammuffati: Virus antico, ed ormai è difficile costruire un virus camuffato che possa creare danni seri. I virus cammuffati cercano di apparire agli antivirus come programmi legittimi.

In passato i visur più diffusi erano proprio i virus camuffati. Inoltre, visto anche la "semplicità" dei virus stessi. in passato gli antivirus controllavano solo le "signature" dei virus, che identifica univocamente ogni virus. Questo però generava dei "falsi positivi" ossia ogni tanto quelle sequenze di byte
venivano trovate all'interno di programmi non infetti. In quei casi venivano ignorate. Ecco perché i virus tentavano di cammuffarsi. Non, in realtà, per non essere identificati, ma per essere
scambiati per un "falso positivo" e quindi ignorati. Ma come infettano i virus di oggi? Negli ultimi tempi la robustezza e la sempre crescente qualità dei sistemi antivirus abbia reso terribilmente difficile ingannare questi strumenti di difesa. Le moderne tecniche antivirus non si limitano ormai solo a controllare le signature ma effettuano check molto più sofisticati rendendo praticamente impossibile il diffondersi di virus cammuffati in questo modo.
Perciò negli ultimi anni si è affermato un nuovo concetto, quello di social engineering applicato alla diffusione dei virus. Il social engineering, ovvero l' "arte dell'inganno" è oggi la
tecnica più usata per violare la sicurezza di un oggetto/soggetto. Infatti ormai la sicurezza in un sistema informatico, anche domestico, è molto alta grazie alla presenza di Firewall, antivirus, uniti anche a software anti malfare spyware ecc.. Pertanto ormai l'unico modo di creare danno è quello di ingannare l'essere umano e non più il PC, ad esempio ultimamente vi è una forte diffusione di e-mail dove il loro scopo non è altro che quello di fingersi qualcun altro.

5. Un approccio minimalista alla progettazione e configurazione della sicurezza di un sistema operativo è spesso il modo migliore per ridurre i rischi. Il sistemista deve sempre però porre in atto una serie di operazioni necessarie per il buon funzionamento del sistema stesso. Quali sono queste operazioni e perché?

L'approccio minimalista che usa il sistemista è dovuto alla "regola d'oro" la quale specifica che "tutto ciò che non c'è non si può rompere" e questo è proprio il punto fondamentale di questo approccio. Il sistemista durante il processo di progettazione deve appunto cercare di eliminare ogni componente superflua del sistema in modo che questa non possa essere la base per una mancanza di sicurezza del sistema stesso. La realizzazione della

sicurezza di un sistema operativo si differenzia a partire dal tipo di sistema sotto esame: in sistemi Open Source è possibile procedere con l'approccio minimalista modificando direttamente il kernel (con maggiore spazio di manovra ma maggiori possibilita di sbagliare), cioè eliminando ogni modulo che viene ritenuto in eccesso e ricompilando poi il kernel modificato; in sistemi operativi dove non è possibile modificare il kernel bisogna procedere con la disattivazione dei moduli non strettamente necessari anche se questi non verranno eliminati definitivamente come nel caso di kernel modificabili; questa operazione è detta Hardening di un sistema operativo cioè il lavoro svolto per aumentare la sua sicurezza. Oltre ai moduli base bisogna ovviamente assicurare anche i moduli che ci permettono di svolgere le operazioni per cui il sistema è stato creato. Un problema che può sorgere da questo approccio è quello delle dipendenze: il sistemista deve infatti conoscere le dipendenze (esplicite o implicite) tra i vari servizi, cioè la possibilità che un determinato servizio necessiti dell'attivazione automatica di un altro servizio per poter essere attivato. Questa operazione di studio delle dipendenze deve essere una parte fondamentale della progettazione in modo da evitare la mancanza di servizi che impedirebbero così alla macchina di continuare nel suo funzionamento. Tute queste considerazioni fanno capire che l' operazione necessaria e primaria nel rendere un sistema sicuro sia quella di conoscere nei dettagli il sistema e le sue componenti in modo da evitare progettazioni dove mancano dei moduli di fondamentale importanza.

7. Effettuare una attenta disamina dei vari tipi di RAID noti. Spiegare dettagliatamente il funzionamento e le caratteristiche (pro e contro) di ognuno di essi.

L'acronimo RAID (Redundant Array of Inexpensive Disks) sta ad indicare proprio un insieme di piccoli dischi poco costosi che insieme vanno a formare un unità di memorizzazione di maggiori dimensioni. L'introduzione di RAID è dovuta all'aumentare dei prezzi di supporti di memorizzazione che,essendo stati sviluppati e migliorati notevolmente, portavano ad una spesa esorbitante; questo può essere considerato un motivo secondario poiché il motivo principale è cercare di salvaguardare i dati da possibili guasti o malfunzionamento dei supporti fisici in modo da prevenire la perdita dei dati piuttosto che concentrarsi solo sullo sviluppo dei supporti di backup. La funzionalità principale di RAID è quella di conservare più copie di uno stesso dato in modo che la perdita di uno dei dischi non provochi la totale perdita dei dati in esso contenuti visto che questi sono replicati in altri dischi. Esistono sono due concetti fondamentali nell'ambito dei RAID: Mirroring e Striping; il Mirroring è la capacità di un RAID di replicare dei dati scrivendoli contemporaneamente su più dischi dell'array; in questo caso alla rottura di un dischi basterà semplicemente sostituirlo e ricostruire i suoi dati presenti su altri dischi. Lo Striping è invece una tecnologia che permette di aumentare notevolmente le prestazioni del RAID perché prevede la scrittura dei dati distribuita su più dischi contemporaneamente aumentando così la velocità di scrittura del dato. I tipi di RAID più importanti sono il tipo 0, il tipo 1 e il tipo 5; il RAID 0 non offre praticamente alcuna protezione del dato in quanto implementa solo lo striping e quindi aumenta solamente le performance; ha perciò il grande difetto di una perdita dei dati nel caso della rottura di un disco. Il RAID 1 implementa solamente il mirroring e garantisce la migliore protezione possibile dei dati ed ha anche performance elevate (non come il RAID 0 però), ma il difetto maggiore è la maggiore disponibilità di spazio poiché salvare N byte necessita 2N byte di spazio. Il RAID 5 implementa lo striping a livello di blocchi con controlli di parità, suddividendo cioè il dato in blocchi e distribuendolo tra i vari dischi. Esso offre il miglior rapporto qualità/prezzo e garantisce anche un'elevata protezione dei dati. Per contro esso richiede la presenza di

almeno 3 dischi per implementarlo ed il salvataggio di N byte necessita di NK (con K<2) byte di spazio. Infine è stato creato anche il RAID 0+1 che, come si nota dal nome, unisce le caratteristiche del tipo 0 e 1 arrivando a fornire prestazioni elevatissime (grazie allo striping) e ottima protezione dei dati (dovuta al mirroring). I contro di questo tipo RAID sono l'uso dello spazio doppio rispetto ai dati salvati come nel RAID 1 e l'utilizzo di 4 dischi come minimo per supportare RAID 0+1.

8. Illustrare la teoria della complessità evidenziando come essa sia valida ai sistemi in generale. Ci si soffermi poi ad analizzare come l'aumento della complessità incide sui problemi di sicurezza e protezione dei sistemi operativi e dei dati in particolare. Proporre, ove possibile, degli esempi (anche in pseudocodice, volendo)

La complessità di un sistema, e in particolar modo di un sistema informatico è una delle principali cause di errori e di problemi che possono sfociare in bug riguardanti la sicurezza. Come è ben noto infatti, quanto più aumenta il grado di complessità di un sistema qualsiasi, più alta è la probabilità di compiere errori. Tali errori tuttavia non provengono da parte della "macchina", che programmata correttamente non sbaglierà mai pur essendo il problema molto complesso, ma dal lato "Umano", che come è risaputo e scientificamente comprovato all'aumentare della complessità di un problema tende ad aumentare il numero di errori commessi. Questo aspetto è da non sottovalutare nell'ambito della sicurezza, e più in generale nell'informatica, in quanto tutte le macchine programmate sono programmate appunto dall'uomo che in quanto tale commette sistematicamente errori. Si calcola che, per esempio, per quanto possano essere bravi dei programmatori , è presente 1 bug ogni 500 righe di codice. Se si pensa che un sistema operativo è composto da milioni di righe di codice, risulta subito evidente come programmi complessi come un sistema operativo possano essere fatti facilmente vittime di attacchi o di errori dovuti alla natura stessa del programma. Quindi la diminuzione del livello di sicurezza in un sistema è derivante dal fattore umano, fattore tuttavia ineliminabile. Gli errori più comuni dei programmatori sono di solito da ricercare "nei soliti": Division by zero (La maggior parte dei processori generano una eccezione quando viene tentata la divisione intera per zero. Il risultato è tipicamente la terminazione del programma anche se in alcuni casi (specialmente quelli che impiegano l'aritmetica a virgola fissa nel caso in cui non sia disponibile hardware dedicato per la virgola mobile) viene impiegato un comportamento simile allo standard IEEE, utilizzando grandi numeri positivi e negativi per approssimare gli infiniti.), buffer overflow (consiste nel fatto che tale programma non controlla in anticipo la lunghezza dei dati in arrivo, ma si limita a scrivere il loro valore in un buffer di lunghezza prestabilita, confidando che l'utente (o il mittente) non immetta più dati di quanti esso ne possa contenere: questo può accadere se il programma è stato scritto usando funzioni di libreria di input/output che non fanno controlli sulle dimensioni dei dati trasferiti.). Ecco perché è necessario che i software siano testati e sottoposti a verifiche formali, a controllo della qualità etc. in quanto maggiori sono le garanzie derivanti da queste verifiche maggiore è la stabilità del sistema, maggiore sarà la sicurezza.
Ex: Si supponga di avere un server Web che contenga questo codice:

```
void func(char *str) {
char buf[256];
strcpy(buf,str);
}
```

Se la stringa inserita è maggiore di 256 caratteri, la funzione strcpy copierà la parte in eccesso nel buffer, provocando un buffer overflow

Controllo degli accessi: distinzione tra autenticazione e autorizzazione. Tradizionalmente come avvengono questi due momenti in un sistema informatico? E nel caso si decida di adottare un'infrastruttura di single sign-on cosa cambia? Mettere a confronto sistemi tradizionale e biometrici, quali sono i vantaggi e i punti deboli dell'una e dell'altra tecnologia?

Spesso i concetti di autenticazione autorizzazione vengono confusi o non sono chiaramente definiti. Innanzitutto va detto che i due concetti sono sequenziali: ovvero l'autenticazione precede l'autorizzazione. Quindi senza autenticazione non si può essere autorizzati a fare qualcosa. Si può dunque affermare che l'autenticazione è la procedura tramite la quale si procede all' identificazione di un soggetto (teoricamente "unico") a una macchina o a un software. Tuttavia l'autenticazione verifica l'esattezza delle informazioni che si inseriscono (ad esempio password) ma non garantisce l'identità "reale" di chi inserisce la password. Successivamente all'essersi autenticati vi è la fase di autorizzazione, che è il processo di garantire o negare l'accesso alle risorse di un sistema in cui ci si è autenticati. Di solito le specifiche dell'autorizzazione, ovvero chi può usufruire delle risorse del sistema, sono a discrezione dell'amministratore di sistema che ha appunto il compito di assegnare o revocare le autorizzazioni. Per "risorsa del sistema" ovviamente si intende qualunque risorsa, dalla concessione o negazione di un determinato privilegio di accesso ad un file (lettura/scrittura/modifica/cancellazione) alla concessione o negazione dell'accesso ad una specifica risorsa di rete (un disco condiviso, un sito web, …). Nel caso del SSO, basta una sola autenticazione per aver accesso ai vari servizi di cui si vuole disporre, ciò evita il problema di doversi autenticarsi ogni volta che si richiede un nuovo servizio a un nuovo server. Lo stato dell'autenticazione (non i dati) vengono trasmetti attraverso un token agli altri server.Tuttavia se un attacker riuscisse a rubare i dati per l'autenticazione sso di un utente avrebbe accesso a tutti i servizi di cui l'utente poteva regolarmente usufruire.

La metodologia classica e la più usata (ma al contempo più debole) di autenticazione è senza dubbio la coppia username/password. Tramite una stringa alfanumerica conosciuta unicamente dall'utente, è possibile farsi riconoscere dal sistema che così applicherà le autorizzazioni pre-impostate dall'amministratore di sistema. Tuttavia questo metodo ha molti punti deboli: innanzitutto le password possono essere rubate, e quindi l'eventuale ladro potrebbe impersonare il legittimo utente, creando notevoli problemi di sicurezza. Inoltre password brevi sono facilmente scopribili tramite attacchi di tipo "brute force" e inoltre una buona amministrazione richiederebbe il cambio abbastanza frequente delle password. Tuttavia rimane il mezzo di autenticazione più utilizzato in quanto è quello che garantisce costi bassissimi e prestazioni tutto sommato abbastanza buone.

Al contrario la biometria, scienza che ha come oggetto di studio la misurazione delle variabili fisiologiche o comportamentali tipiche degli organismi, attraverso metodologie matematiche e statistiche, ha costi molto alti ma in compenso è ugualmente alta anche la sicurezza dell'autenticazione. Metodi di autenticazione biometrica possono considerarsi le impronte digitali, la lettura dell'iride o della retina, la struttura venosa della mano, la geometria del volto o della mano, il timbro della voce. La più economica tra queste è la lettura delle impronte digitali, che con costi bassi assicura un certo grado di sicurezza nell'autenticazione, anche grazie a ultime tecnologie che oltre alle impronte controllano anche la temperatura e la pulsazione. Anche la scansione della struttura venosa è un ottimo metodo di autenticazione biometrica, sicuramente più costoso di quello delle impronte digitali, ma molto più robusto in quanto la sicurezza del sistema è data

dall'estrema difficoltà di riproduzione del modello dei percorsi che rende dunque praticamente impossibile la clonazione. La scansione del volto e delle mani invece non risulta essere un buon metodo, in quanto le componenti fisiologiche in questione sono soggette a cambiamenti con l'avanzare dell'età e non solo. Forse il sistema più sicuro dal punto di vista della sicurezza dell'autenticazione è la scansione dell'iride, che risulta essere estremamente complesso e ricco di informazioni come modello fisiologico, con oltre 200 punti unici: basti pensare che le iridi dei due occhi di un medesimo individuo sono differenti. Il grosso problema di questo metodo sono i costi esorbitanti che richiedono le apparecchiature atte alla scansione. Infine la scansione del timbro di voce, che forse risulta essere il metodo meno preciso in quanto le prestazioni del riconoscimento della voce possono variare sia in funzione della qualità del segnale così come a causa del cambio di fonte utilizzata per l'enrollment e per la verifica; oltre che a disturbi esterni nella cattura della voce, a malattie (tosse, vento), o addirittura nel caso di ragazzi e adolescenti, il cambio del timbro di voce dovuto alla naturale crescita.

10. Parlando di protezione del dato, quali sono le due tematiche che il buon sistemista dovrà affrontare per configurare al meglio il sottosistema di memorizzazione (lo storage)? In quale ordine e con quali strumenti risolverà tali problematiche?

Le tematiche che il buon sistemista deve rispettare sono:

a) GARANTIRE LA POSSIBILITA' DI RECUPERARE DATI PERSI;
b) EVITARE DI PERDERE I DATI;
c) EVITARE CHE I DATI POSSANO ESSERE LETTI DA PERSONE NON AUTORIZZATE.

Pianificando una strategia di backup sarà essenziale scegliere in modo appropriato dove la replica dei dati verrà effettuata. Vi sono moltissime possibilità, consideriamo solo le più note ed utilizzate:

1. DAT (Digital Audio Tape) interno
2. DAT array esterno
3. DAS (Directly Attached Storage)
4. NAS (Network Attached Storage)
5. SAN (Storage Area Network)

1) DAT: Sono comuni audiocassette digitali che possono contenere grandi quantità di dati, in genere parliamo di 40, 80 o anche oltre 200 Gb.
Possono essere lette/scritte mediante appositi device che possono essere installati all'interno di un PC (occupano lo spazio di un CD-ROM drive) oppure collegati all'esterno mediante cavi USB, FireWire o in fibra ottica.

Vantaggi:
- Il supporto è rimovibile e può essere archiviato altrove
- Costo relativamente contenuto
Svantaggi:
- Estremamente lento
- Fallibilità superiore alla media

2) DAT ARRAY ESTERNO: Si tratta di speciali server contenenti una serie di cassette DAT e dotati di un selettore in grado di caricare/archiviare le varie cassette a seconda della necessità.
Questi device sono collegati in rete solitamente attraverso comuni cavi ethernet (RJ45) e dispongono di una enorme capacità di archiviazione.

Vantaggi:
- Enorme capacità di archiviazione (spesso > 1 Tb.)
- Facilità di gestione
Svantaggi:
- Estremamente lento
- Acquisto e manutenzione costosi

3) DAS: Un Directly Attached Storage non è altro che un hard diskdirettamente collegato al PC. L'hard disk che tutti noi abbiamo all'interno del nostro PC è un DAS.
Ma un DAS può anche essere esterno pur restando direttamente collegato al PC mediante un cavo USB, FireWire o in fibra ottica. In questo modo si ottiene un supporto staccabile e trasportabile.

Vantaggi:
- Estremamente veloce
- Bassa fallibilità
Svantaggi:
- Contenuta capacità di archiviazione
- Inadatto al network-backup

4) NAS: Un Network Attached Storage è un disco (o un array di dischi) collegati ad un PC che funge da file server e che condivide lo spazio di quei dischi in rete. Non è molto diverso dalla condivisione di un disco da un normale PC.

Vantaggi:
- Tutti i PC della rete lo possono usare contemporaneamente
- Elevata capacità di memorizzazione
Svantaggi:
- Costi leggermente superiori alla media
- NON-trasportabilità dei backup

5) SAN: Una Storage Area Network è uno strumento estremamente complesso che espone ai PC della rete dischi collegati direttamente al network come se fossero locali, "guidabili" mediante protocolli SCSI. Gli array di dischi sono collegati in rete mediante particolari
switch studiati ad-hoc per le SAN.

Vantaggi:
- Capacità di memorizzazione infinitamente espandibile
- Velocità e affidabilità
- Adatta all'uso in WAN mediante iSCSI e FCIP
Svantaggi:
- Costo elevato

- Complessità elevata

+ RAID ??????????????????????????????????

11. Autenticazione e autorizzazione, concetti complementari ma distinti. In che modo questi strumenti sono funzionali alla protezione e quindi all'aumento della sicurezza di un sistema informativo? Quali strumenti tecnici possiamo utilizzare per implementare questi controlli?

Nel campo della sicurezza informatica si definisce autenticazione il processo tramite il quale un computer,un software o un utente, verifica la corretta identità, tramite opportuni controlli, di un altro computer,software o utente che vuole comunicare attraverso una generica "connessione". Se l'autenticazione andrà a buon fine,il sistema rilascerà all'utente determinate autorizzazioni, cioè una "lista" di azioni che solo quell'utente autenticato precedentemente potrà o non potrà compiere sulla parte di sistema interessata; l'autorizzazione dipende strettamente dall'autenticazione.
L'autorizzazione viene praticata solo dopo che l'utente è stato opportunamente autenticato, in un momento che è sia logicamente che fisicamente distinto da quello dell'autenticazione.
I controlli opportuni che il processo di autenticazione esegue vengono raggruppati nel campo del CONTROLLO DEGLI ACCESSI.
Il controllo degli accessi è un campo vastissimo che prende in considerazione tutti i casi nei quali è necessario fornire un accesso selettivo a informazioni o a locazioni specifiche.
Indipendentemente dal metodo utilizzato e dall'oggetto (o dato) sul quale vogliamo imporre un accesso controllato, lo scopo del controllo degli accesso è sempre lo stesso: fare in modo che solo chi è autorizzato possa accedere.

L'autorizzazione riguarda qualunque risorsa, dalla concessione o negazione di un determinato privilegio di accesso ad un file (lettura/scrittura/modifica/cancellazione) alla concessione o negazione dell'accesso ad una specifica risorsa di rete (un disco condiviso, un sito web, …).
Garantisce che le regole imposte dall'amministratore di sistema sugli oggetti appartenenti al sistema stesso vengano rispettate.

Un primo esempio di controllo degli accessi è davanti ai nostri occhi ogni giorno quando accendiamo il PC.
Per poter utilizzare il nostro sistema operativo ci viene chiesto di verificare la nostra identità inserendo username e password. In base allo username e alla password il sistema operativo è in grado di associare a noi il nostro profilo utente e di consentirci l'accesso a tutte quelle risorse e quei dati cui abbiamo diritto. Questo metodo fu uno dei primi "rimedi" per aumentare la sicurezza nei sistemi, tuttora è il più diffuso ma al tempo stesso il più insicuro, possibili attacchi di sniffing, snooping, guessing ecc…
E' quindi evidente che l'obbligo di inserire username e password può non essere sufficiente a garantire un corretto e sicuro controllo degli accessi ai sistemi operativi. Per ovviare a questo problema sono stati predisposti metodi alternativi, alcuni anche molto complessi:

SMARTCARD
TOKEN-KEYS
BIOMETRIC TOKEN-KEYS

Recentemente si sta sviluppando sempre più un tipo di autenticazione basata su caratteristiche fisiche di una persona, intrinsecamente affidabili perché uniche di quell'individuo. Queste caratteristiche vengono trattate dalla BIOMETRIA,che fornisce alcuni metodi di autenticazione:

IMPRONTE DIGITALI
STRUTTURA VENOSA DELLA MANO
GEOMETRIA MANI/VOLTO
SCANSIONE IRIDE/RETINA
TIMBRO VOCE

12. I file system sono nati come strumenti logici atti a rappresentare i dati in forma strutturata. Oggi però sono divenuti qualcosa di più

complesso. Effettuare una disamina comparativa tra i vari tipi di file system noti. Illustrare le differenze filosofiche e implementative relative alla crittografia del file system nei sistemi Windows e Linux.

Sopra NTFS infatti Microsoft ha implementato EFS (Encrypted File System) che però non garantisce un livello di sicurezza sempre adeguato.
The Encrypting File System (EFS) provides the core file encryption technology used to store encrypted files on NTFS volumes. EFS keeps files safe from intruders who might gain unauthorized physical access to sensitive, stored data (for example, by stealing a portable computer or external disk drive).
Users work with encrypted files and folders just as they do with any other files and folders. Encryption is transparent to the user who encrypted the file; the system automatically decrypts the file or folder when the user accesses. When the file is saved, encryption is reapplied. Users who are not authorized to access the encrypted files or folders transparently receive an "Access denied" message if they try to open, copy, move, or rename the encrypted file or folder. The exact message text may vary depending on application which tries to access the file, because it is related not to user rights for file but to ability of EFS to decrypt file using user's private key.
EFS has the following benefits over 3rd party encrypting applications:
It is transparent for user and any applications. There's no risk for user to forget to encrypt file and leave data unprotected. Once file or folder is marked as encrypted, it will be encrypted in background without interaction with user. User does not need to remember password to decrypt files.
Strong key security. In contrast to other solutions when keys are based on user entered pass-phrase, EFS generates keys which are tolerant to dictionary based attacks.
All encrypting/decrypting processes are performed in kernel mode, excluding the risk of leaving key in paging file, from where it could be possibly extracted.
EFS provides data recovery mechanism which is valuable in business environment, giving an organization an opportunity to restore data even if the employee who encrypted it left the company.
Encrypting File System (EFS) provides the core file encryption technology used to store encrypted files on NTFS file system volumes. Once you encrypt a file or folder, you work with the encrypted file or folder just as you do with any other files and folders.

Encryption is transparent to the user that encrypted the file. This means that you do not have to manually decrypt the encrypted file before you can use it. You can open and change the file as you normally do.

Using EFS is similar to using permissions on files and folders. Both methods can be used to restrict access to data. However, an intruder who gains unauthorized physical access to your encrypted files or folders will be prevented from reading them. If the intruder tries to open or copy your encrypted file or folder he receives an access denied message. Permissions on files and folders do not protect against unauthorized physical attacks.

You encrypt or decrypt a folder or file by setting the encryption property for folders and files just as you set any other attribute such as read-only, compressed, or hidden. If you encrypt a folder, all files and subfolders created in the encrypted folder are automatically encrypted. It is recommended that you encrypt at the folder level.

LINUX: (LOOP-AES) Loop devices are block devices that don't store any data directly, but rather redirect all reads and writes to an underlying block device or file, possibly encrypting or decrypting data in the process.

9. I sistemi Windows e quelli della famiglia Linux sono basati su modelli di sicurezza assai differenti. Illustrare le difformità filosofiche e implementative relative alla crittografia del file system nei sistemi Windows e Linux, sia dal punto di vista del file system stesso (nativo) che da quello di eventuali software (add-on) di terze parti.

1.La protezione del dato è al centro del concetto di sicurezza di sistema. Illustrare nel dettaglio quali problematiche di gestione della sicurezza attengono all'ambito specifico della protezione del dato quando esso è residente su supporti di memorizzazione non volatili.