Un virus informatico è un programma auto-replicante che opera senza il consenso dell'utente. Si diffonde attaccando una propria copia a qualche parte di un programma come Word o Excel.
I virus possono anche attaccare il *boot sector* o il *Master Booot Record*, che contengono informazioni vitali per l'avvio del computer.
Alcuni virus semplicemente si replicano, altri visualizzano messaggi; alcuni però, contengono un *payload* cioè una porzione di codice progettata per corrompere programmi, cancellare files, formattare hard-disk, mettere in ginocchio reti aziendali, riuscendo a volte ad intaccare il bene più prezioso: le informazioni immagazzinate in anni di lavoro.

VIRUS SEMPLICI
Un virus che replica sé stesso è piuttosto facile da individuare. Se un utente lancia un programma infetto, il virus prende il controllo del computer e aggancia una propria copia ad un altro programma. Dopo essersi diffuso, il virus trasferisce il controllo al programma ospite il quale, a questo punto, funziona normalmente. L'infezione farà sempre un'esatta copia di sé stessa, indipendentemente dal numero di volte che ha infettatati un file o un supporto di memorizzazione.
I software anti-virus devono solo compiere una scansione dei files alla ricerca di una ben precisa serie di bytes, chiamata *signature*, che si trova all'interno del virus.

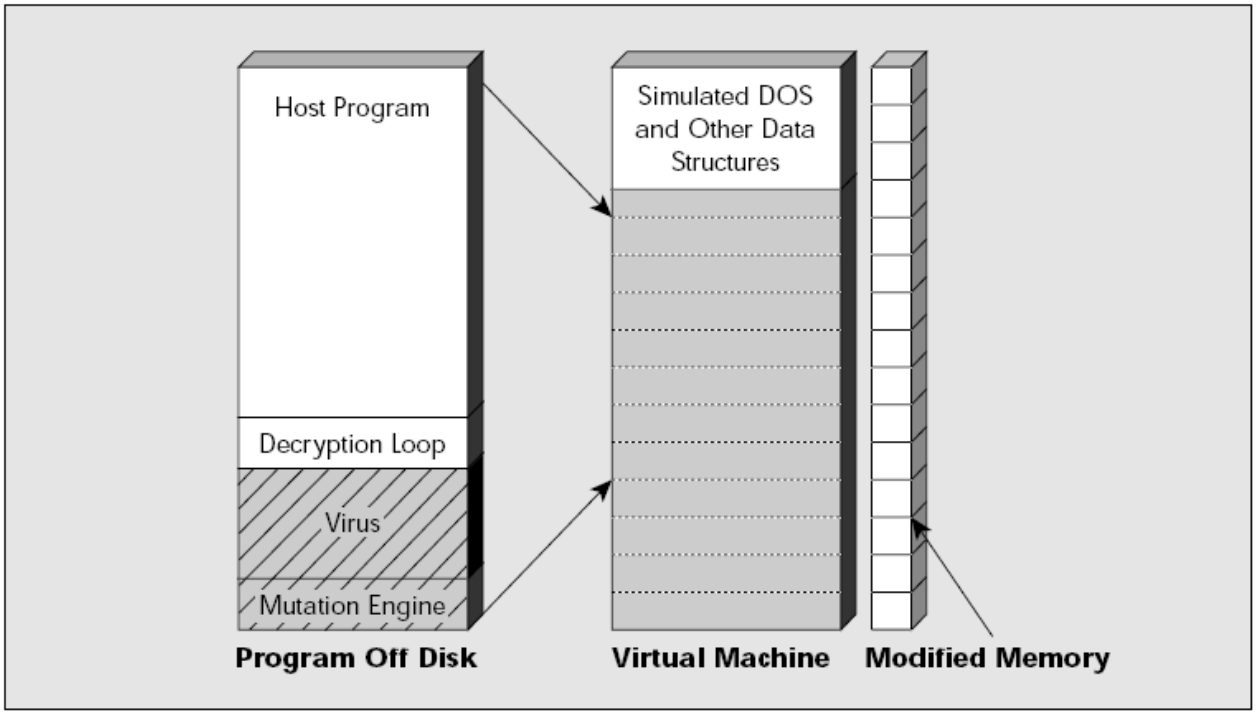
VIRUS CIFRATI
L'idea è quella di cifrare il codice dei virus in modo da nascondere la *signature* alla scansione degli anti-virus. Un tale virus si compone di una *routine* di decifratura e del corpo vero e proprio del virus (naturalmente cifrato). Se un utente lancia un programma infetto, la routine di decifratura del virus dapprima prende il controllo del computer, e poi decifra il corpo del virus. Quindi, la routine di decifratura trasferisce il controllo del computer al virus decifrato.
Un virus cifrato infetta i programmi ed i files nello stesso modo in cui lo fanno i virus semplici. Ogni volta che infetta un nuovo programma, il virus fa una copia del corpo decifrato e della relativa routine di decifratura, cifra la copia e li aggancia entrambi al file vittima.
Per cifrare una nuova copia, il virus utilizza una chiave di cifratura che viene cambiata ad ogni infezione, rendendo così il virus differente di volta in volta. Questo fa sì che sia estremamente difficile per l'antivirus rilevare il virus attraverso la ricerca della *signature*.
Tuttavia, la routine di cifratura rimane la stessa durante le varie infezioni e questo torna a tutto vantaggio degli anti-virus. Infatti, invece di cercare la s*ignature* del corpo del virus, verrà ricercata la *signature* della routine di decifratura.

VIRUS POLIMORFICI
Come un virus cifrato, anche un virus polimorfico include un corpo cifrato e una *routine* di decifratura che si occupa di decifrare il corpo del virus e di trasferirgli il controllo del computer. Tuttavia, un virus polimorfico aggiunge a queste due componenti un terzo fattore, ovvero un motore di mutazione che genera *routines* di decifratura in modo pseudo-randomico in modo che cambino ad ogni nuova infezione. Inoltre, sia il motore di mutazione che il corpo del virus sono cifrati
Quando un utente lancia un programma infetto, la *routine* di decifratura prende il controllo del computer, decifra sia il corpo che il motore di mutazione e poi trasferisce il controllo al virus che localizza un nuovo file da infettare. A questo punto, il virus fa una copia di sé stesso e del motore di mutazione e li pone in RAM, invoca il motore di mutazione il quale genera una nuova *routine* di decifratura differente da quelle usate in precedenza. Quindi, il virus cifra la nuova copia di sé stesso e del motore di mutazione e li aggancia, assieme alla nuova *routine* di decifratura, al file vittima.
Quando non è possibile procedere alla ricerca di una precisa *signature* o di una precisa *routine* di decifraura, le infezioni successive di uno stesso virus risultano come differenti, complicando moltissimo il lavoro di un software anti-virus.

VIRUS POLIMORFICI – INDIVIDUAZIONE

I ricercatori hanno cercato di sviluppare alcune *routines* in grado di individuare i virus polimorfici in modo specifico, scansionando linea per linea il loro codice cifrato e confrontandolo con quelli generati da un certo tipo di motore di mutazione conosciuto. Ma questo approccio si rivela inefficiente e costoso, in quanto ogni nuovo virus necessita di una nuova routine di individuazione. Inoltre, alcuni motori di mutazione possono produrre miliardi e miliardi di varianti, vanificando questo tipo di soluzione.

Esistono scanner di virus polimorfici che adottano come tecnica di scansionare il file da verificare solo dopo averlo caricato in una macchina virtuale creata in RAM, in modo da limitare gli eventuali danni di un virus polimorfico. In questo modo, il software antivirale ha a disposizione il corpo del virus decifrato per confrontarlo con le *signatures* disponibili e può cercare di identificarlo gestendolo in un'area relativamente protetta come quella di una macchina virtuale.



Questo tipo di approccio presenta però un problema in termini di tempo che può portare facilmente alla sua non applicabilità nel mondo reale. Infatti, il virus potrebbe non rivelarsi pienamente nella finestra temporale che l'antivirus dedica alla scansione del file e questo produrrebbe come effetto un falso negativo.

Per risolvere tale problema, si ricorre ad un approccio *euristico*, cioè l'utilizzo un insieme di regole per identificare un virus a partire dal suo comportamento. Ad esempio, un antivirus potrebbe notare azioni particolari (elaborazione di calcoli poi non utilizzati) da parte di un file e percependole come diversivi potrebbe decidere di dedicare maggiore tempo alla scansione del file in questione.

Inizialmente, si assume che ogni file abbia una probabilità del 10% di essere infetto, probabilità che aumenta o diminuisce a seconda del comportamento del file, secondo una serie di regole predefinite.

Il seguente schema riporta un'esemplificazione di alcune regole per una scansione con approccio euristico.

- Regole Pessimistiche
  - Se si riscontra nel codice una istruzione NOP, allora la probabilità di infezione cresce del 0.5%
  - Se il contenuto di un registro viene distrutto prima di essere usato, la probabilità di infezione cresce di 1.2%
- Regole Ottimistiche
  - Se il programma genera degli Interrupt DOS, la probabilità diminuisce del 15%
  - Se il programma non realizza scritture in memoria entro 100 istruzioni eseguite, la probabilità di infezione diminuisce del 5%

In conclusione, gli approcci sopra desctitti sono validi ed efficaci solamente se il software antivirus ha a disposizione il codice decifrato del virus caricato in memoria in un ambiente virtuale.
Ma gli autori di virus potrebbero creare infezioni che rimangono in uno stato dormiente fino ad un certo evento. Ad esempio, si immagini un file che necessita di una particolare sequenza di tasti per avviarsi; nel caso un tale file sia infettato, la sequenza di tasti eseguirà il file ed anche il virus. Purtroppo nell'ambiente virtuale approntato dall'antivirus,mancando l'interazione con l'utente umano, questo evento scatenante non avverrà ed il codice virale rimarrà cifrato vanificando l'azione antivirale.