Il problema dei virus informatici: tecniche di difesa e soluzioni anti-virus

Stefano Coeli

Relatore: Francesco Bergadano

Università di Torino
Facoltà di Scienze M. F. N.
Dipartimento di Informatica
T 27 - 99/2000

Ai miei genitori,
che hanno creduto in me,
e a Cristo, sale della mia vita

Indice

0 - Presentazione

Nel presente lavoro si intende trattare il problema dei virus informatici in modo globale, prendendo in esame i vari aspetti del problema, le cause scatenanti, le conseguenze e le contromisure intraprese, le fonti di informazione. Viene anche introdotto il problema più ampio del software dolosamente dannoso, o malware. Una particolare attenzione è rivolta all'aspetto tecnico del problema e delle contromisure, analizzando le più recenti soluzioni ideate per rilevare ed eliminare la presenza di software dannoso. Viene preso in considerazione l'aspetto economico secondo due differenti aspetti: quello dei danni prodotti e quello delle nuove industrie sorte per contrastare il fenomeno. Si tenta di dare una spiegazione delle cause psicologiche e sociali che hanno fatto nascere il problema e continuano ad alimentarlo, al punto da sostenere una crescita del numero di nuovi software dannosi quasi esponenziale. Si analizzano le prese di posizione etiche e professionali di alcuni importanti ricercatori del campo, riguardo alle linee di tendenza future ed alle norme di sicurezza necessarie per operare un'efficace prevenzione, con un'attenzione particolare alle ricerche effettuate sulle tecnologie anti-virus, all'analisi delle nuove soluzioni e delle possibili future evoluzioni. Vengono esaminati e presentati i principali prodotti anti-virus presenti sul mercato, con un'analisi del supporto e delle informazioni fornite attraverso la rete Internet, assieme ad alcune note sulle compagnie produttrici e ad una critica delle attuali tecniche di valutazione dei prodotti. Si fornisce inoltre una descrizione degli istituti e delle organizzazioni preposte allo studio e al monitoraggio del fenomeno, attraverso l'analisi dei rispettivi siti presenti sulla rete Internet e, quindi, del materiale messo a disposizione del pubblico.

1 - Introduzione

In questo capitolo si vuole dare un'idea generale del problema dei virus e dei vari tipi di software progettato per danneggiare o comunque penetrare in sistemi altrui in modo nascosto e non lecito, fornendo un'introduzione ai principali aspetti del problema e alla loro terminologia; infine, una storia della nascita e dell'evoluzione di questi software, dagli inizi fino al periodo odierno.

1.1 - Descrizione del problema

La definizione di virus elaborata da Fred Cohen[1], pioniere dello studio di questo problema, è la seguente: un virus è un programma capace di infettare altri programmi, modificandoli in modo da includere una copia possibilmente evoluta di se stesso. Il fenomeno odierno dei virus informatici riguarda programmi definiti come sopra, creati da un programmatore per motivi diversi e distribuiti attraverso diversi canali in modo occulto, al fine di infettare più programmi possibili su diversi sistemi, senza che gli utilizzatori se ne possano rendere conto. Si può creare un'analogia con la situazione seguente: il programmatore di virus è come un ingegnere genetico che crea un virus inteso in senso biologico, dopodiché cerca dei canali occulti, come potrebbe essere l'acquedotto, per infettare il maggior numero di individui a loro insaputa, in modo da iniziare un'infezione che possa propagarsi il più possibile.

Si distinguono diversi tipi di virus, secondo il metodo usato per riprodursi e del tipo di codice che è infettato. Un virus inoltre può contenere un certo numero di *payload,* termine utilizzato per descrivere azioni intraprese dal virus diverse da quelle necessarie alla riproduzione. I payload sono di diversi tipi e possono essere solo rappresentativi come estremamente distruttivi. Spesso sono in uno stato di quiescenza e si attivano in momenti particolari o sotto condizioni particolari.

Con il termine virus spesso si intende un problema più generalizzato che attualmente è definito con il termine *malware*. Il malware è un software che penetra in un sistema in un

[1] Riferimento [8], disponibile presso: http://all.net/books/virus/top.html.

modo nascosto e ha effetti indesiderati, più o meno gravi, che possono andare dal semplice spreco di risorse ad effetti distruttivi.

Una suddivisione generale del malware odierno è la seguente:

- Virus: malware capace di riprodursi secondo la definizione di Cohen. Infetta il codice eseguibile.

- Boot virus: particolare tipo di virus, che infetta il codice di boot di un sistema. Si riproduce attraverso memorie di massa rimovibili (es. floppy disk) utilizzabili per il bootstrap del sistema, che quindi contengano il codice di boot.

- Worm: un programma che copia ed attiva se stesso in altri sistemi attraverso un canale di comunicazione.

- Trojan horse: software con una funzione apparentemente lecita che contiene altre funzioni nascoste ed indesiderate.

- Logic bomb: un particolare trojan horse contenente funzioni nascoste che si attivano in determinate condizioni, ritardate rispetto all'esecuzione del programma portatore.

- Backdoor: un software nascosto che permette ad un utente estraneo al sistema di attraversare le funzioni di protezione ed eseguire dall'esterno azioni non lecite.

Vi sono poi malware che si riconducono a una combinazione di questi tipi.

1.2 - Storia dei virus

I primi virus sono nati su grandi sistemi. Già nel 1949 Von Neumann[2] aveva ipotizzato la realizzabilità di programmi in grado di riprodursi. Alla fine degli anni '50 un gruppo di programmatori dei Bell Laboratories sviluppò un gioco chiamato "Core Wars". Il

[2] J. Von Neumann, Theory and application of complicated automata, 1949.

gioco consisteva nel programmare una serie di “organismi” che dovevano riprodursi nella memoria centrale del computer e cercare di distruggere gli organismi dell’avversario. Vinceva chi dopo un certo lasso di tempo, aveva più organismi attivi in memoria.

Sempre sui mainframe, nel 1970 iniziò a trovarsi sulla rete ARPAnet il programma CREEPER. Questo programma annunziava la sua presenza con il messaggio “I’M THE CREEPER... CATCH ME IF YOU CAN”. Esisteva anche il programma REAPER, programmato in modo simile ma con lo scopo di cercare e cancellare CREEPER. Si trattava dunque di programmi virus con poca capacità distruttiva; erano giochi scappati di mano agli autori o distribuiti con uno scopo goliardico. Gli incidenti verificati erano effetti non previsti e involontariamente causati dagli autori.

Nel 1980 c’è stato il caso dell’IBM Trojan. Un programma cavallo di Troia veniva distribuito ai computer IBM 4341. Alle 7:30 dell’11 aprile 1980 il cavallo di Troia si attivò e bloccò tutti i 4341.

Il primo virus come viene inteso oggi, può considerarsi Elk Cloner, apparso a cavallo tra il 1981 e il 1982. Si trattava di un virus per Apple II che infettava i dischi di boot. Prevedeva alcuni payload: la visualizzazione di una presentazione in versi, effetti grafici sul testo visualizzato, ecc. Per prendere il controllo di un computer, il virus infettava alcuni comandi del sistema operativo dell’Apple II che operavano sui dischi; una volta riuscito in questo intento si diffondeva su tutti i dischi che fossero stati utilizzati.

Nel 1983 F. Cohen presentò ad un seminario sulla sicurezza dei computer un lavoro in cui si presentava per la prima volta il concetto di virus. Il nome gli fu suggerito dal dottor Len Adleman, suo insegnante. Cohen preparò una serie di esperimenti sui computer della sua università, consistenti in una serie di attacchi di virus. Il risultato fu così impressionante che gli amministratori del sistema vietarono ogni successivo esperimento, precauzione che si è rivelata totalmente inutile.. Negli anni successivi il dottor Cohen pubblicò altri lavori sul rischio dei virus informatici, come “Computer Viruses – Theory and Experiments”, 1984, che viene considerata la prima pubblicazione scientifica sul problema dei virus; inoltre alcuni studi sull'aspetto computazionale dei virus.

Nel 1986 compare il virus BRAIN; per la prima volta un fatto simile viene denunciato sui mass media di tutto il mondo. Si tratta di un virus del settore di boot per Personal Computer, non distruttivo, che si riproduceva su tutti i dischetti inseriti nel lettore

modificando l'etichetta del disco in "(C) Brain". A differenza di altri casi, gli autori volevano essere conosciuti; infatti inserirono nel virus i loro dati completi addirittura con numero telefonico. Si trattava di due fratelli del Pakistan, titolari di una software house a Lahore, il Brain Computer Service, i quali avevano scritto questo virus per "punire" i pirati informatici, ai quali loro stessi vendevano programmi copiati. Nel Pakistan a quei tempi la legge sul copyright del software non era chiaro, e quindi i due fratelli potevano copiare senza paura diverso software commerciale. Anche gli stranieri presenti a Lahore, attratti da questo apparente affare, iniziarono a recarsi presso il Brain Computer Service. I due fratelli però erano a conoscenza delle diverse leggi sul copyright all'estero e consideravano i clienti stranieri come dei pirati; per questo ad essi vendettero dei dischetti infettati. L'infezione arrivò rapidamente nelle maggiori università americane, dove venne denominato "BRAIN" ed in seguito "PAKISTANO". Nonostante il virus non avesse payload distruttivi, la sua presenza provocava malfunzionamenti nei sistemi. Per queste cause iniziò la prevenzione del "virus threat", minaccia dei virus. I dischi venivano protetti in scrittura e gli utenti venivano informati dei rischi e responsabilizzati riguardo ai propri dati personali; questo a partire dall'University of Delaware dove il virus venne scoperto.

Pochi giorni prima del Giorno del Ringraziamento del 1987, nella Lehigh University di Bethlehem, Pennsylvania, USA, iniziarono a manifestarsi dei malfunzionamenti nei dischi utilizzati nelle aule computer. Vennero fatte delle indagini approfondite e i tecnici notarono che all'interno del file "COMMAND.COM", file fondamentale del sistema operativo MS-DOS per P.C., vi era una parte lunga circa 300 bytes che il sistema MS-DOS utilizzava come spazio per la memorizzazione temporanea, riempita con il valore zero. Ora questo spazio in alcuni dischetti era alterato e conteneva una porzione di codice assembler. Si trattava di un virus, che alterava l'indirizzo del salto iniziale del file, che è un normale file eseguibile, in modo da passare il controllo alla porzione aggiunta in quello spazio di 300 bytes "vuoto". Quando il file veniva caricato assieme al sistema operativo, prendeva il controllo di alcune routine del sistema che si occupano di svolgere servizi per i file. Le routine venivano modificate in modo da verificare, ad ogni chiamata, se un disco di sistema non infetto era stato inserito e, in questo caso, infettare il disco, incrementando un contatore. Dopo un certo numero di contagi si attivava un payload distruttivo. Risale anche a questa scoperta il primo sviluppo di un anti-virus, FIX, che veniva distribuito gratuitamente agli studenti e dipendenti che ne facessero richiesta; il programma FIX scandiva il file COMMAND.COM e, se rilevava la presenza del virus, lo cancellava riscrivendo la sequenza originale di zeri nella porzione dove il virus si annidava.

Immediatamente i virus iniziano a moltiplicarsi sempre più e a presentarsi come problema generale. In questo stesso periodo un programmatore tedesco, Ralf Burger, pubblicò un libro dal titolo "Computer virus, una malattia ad alta tecnologia", che conteneva il sorgente quasi completo di un virus noto come Vienna. Immediatamente dopo la pubblicazione del libro iniziarono a diffondersi molti virus simili al Vienna, in termini tecnici delle *varianti;* questo fatto rendeva noti alcuni meccanismi psicologici, di cui si parlerà ampiamente nel successivo capitolo, che spiegano la diffusione dei virus.

Parallelamente alla costruzione di microcomputer a grande diffusione, abbiamo già incontrato il P.C. IBM, iniziano presto a comparire virus in grado di infettarli. Nel 1987 compaiono, infatti, virus sui computer Commodore Amiga e Apple Macintosh.

Comunque, data la sua maggiore diffusione, il sistema più colpito da virus è sempre stato il P.C.. Proprio in questo periodo ci sono altri virus da menzionare per la loro grande diffusione, STONED, noto anche come MARIJUANA, e JERUSALEM, noto come Venerdì 13, dal giorno della sua attivazione. Quest'ultimo, scoperto nel dicembre 1987 nella Hebrew University di Gerusalemme, è noto per le voci che furono diffuse dai media al suo riguardo; alcuni giornali parlarono di attacco terroristico ad Israele tramite questo virus, altri insinuavano che il virus potesse infettare i mainframes e che quindi i computer del ministero della difesa Israeliano fossero tutti infetti. Le voci risultarono prive di ogni fondamento.

Il virus Ping-Pong risale a questo periodo. Questo virus è stato realizzato in Italia, si dice dagli studenti del Politecnico di Torino. Alcuni autori di quel periodo ricordano che il fatto venne accolto in un modo alquanto strano: sembrava un vanto per l'Italia che i nostri informatici fossero arrivati ad un livello tecnico sufficiente per scrivere un virus. Un atro virus molto conosciuto in quel periodo e molto dannoso era DATACRIME. Si trattava di un virus che in caso di attivazione, il 13 e il 31 ottobre, avrebbe formattato la traccia zero del disco fisso, rendendo così inutilizzabili i dati memorizzati su di esso. Queste date furono attese con paura, ma il virus arrecò meno danni del previsto.

Nel 1990 un programmatore americano realizzò alcuni virus in grado di cambiare aspetto, tecnicamente polimorfi. Le sue creazioni non vennero mai divulgate, ma l'idea venne accolta e iniziarono a comparire i primi virus polimorfi. In particolare si può citare TEQUILA e MALTESE AMOEBA, comparsi nel 1991. L'autore di quest'ultimo, noto con lo pseudonimo Dark Avenger, l'anno successivo distribuì il "Mutation Engine" o "MtE"; si tratta di un tool che consente a programmatori anche non esperti di generare virus polimorfi.

Sempre nel 1990 si ha notizia di primi virus in grado di infettare una rete LAN; il primo caso analizzato, in grado di infettare una rete Novell NetWare, è stato analizzato da un gruppo di diversi ricercatori e studiosi di sicurezza informatica nel giugno 1990. [33]

Negli anni immediatamente successivi viene individuato il virus Michelangelo, un virus considerato pericolosissimo poiché, il giorno del compleanno di Michelangelo Buonarroti, cioè il 6 marzo, sovrascriveva il disco rigido con caratteri nulli. In realtà il virus arrecò molti meno danni di quanto veniva paventato dai media, infatti, solo pochi P.C. erano stati infettati. Questo si deve anche all'industria del software che iniziò subito, fino dai primissimi momenti della scoperta del problema, a produrre software anti-virus sempre più sofisticato, fino a delineare una vera industria del software anti-virus.

Da notare che lo sviluppo dei virus ha un andamento quasi esponenziale; questo veniva rilevato da un grande esperto quale il dottor Bontchev, già nel 1994. Nel 1992 al NCSA di Washington, USA, venne formato un comitato incaricato di ridurre la confusione nella assegnazione dei nomi ai virus, che erano ormai migliaia. Il comitato era formato da Fridrik Skulason, editore di bollettini tecnici sui virus, Alan Solomon, dell'industria anti-virus S&S international, e lo stesso Vesselin Bontchev, ricercatore dell'università di Amburgo. La convenzione scelta prevede fino nei minimi particolari le regole per l'attribuzione dei nomi, che hanno il seguente schema:

nome_famiglia.nome_gruppo.variante_maggiore.variante_minore

Il "nome famiglia" è un nome attribuito ad un genere di virus con caratteristiche generali comuni, es. i macro virus capaci di infettare Word 97. Il "nome gruppo" è invece il nome di un particolare virus, inclusivo di tutte le sue varianti; esempio "Datacrime". La "variante maggiore" definisce una particolare variante del virus citato, normalmente consiste in un numero che rappresenta la lunghezza in byte della variante stessa. Infine la "variante minore" distingue varianti molto simili, con lievi differenze, ed è normalmente una lettera alfabetica.

Esempio: VBS.NewLove.1234.A
Descrizione:

VBS: famiglia dei virus scritti in Visual Basic Script

NewLove: gruppo delle varianti del virus NewLove

1234: variante lunga 1234 bytes

A: variante minore "A"

Una grande rivoluzione nel mondo dei virus avviene con la comparsa di Windows 95. Questo software è un sistema operativo completo, con una diversa gestione delle periferiche rispetto al DOS; inoltre in quel periodo iniziano a comparire i CD-ROM, che diventano molto presto uno standard per il trasporto dei dati. Per questi motivi i virus del settore di boot smettono di moltiplicarsi. Invece Windows 95 può essere facilmente infettato da virus di file e da virus polimorfi. Inoltre compaiono i primi virus in grado di infettare nuovi tipi di file introdotti con il nuovo sistema operativo: i file VxD, contenenti i device driver virtuali che servono per accedere alle periferiche, e i file Portable Executable o PE, nuovo formato di file eseguibili. Il primo virus in grado di infettare un file PE è stato BOZA. Nel 1998 apparve un altro virus, sempre specifico per PE, di nome Chernobyl o Win.CIH, che produsse gravissimi danni in tutto il mondo. Il virus sfruttava una nuova caratteristica del BIOS, la parte del sistema operativo residente su ROM, che serve per la procedura di accensione del P.C. e per iniziare la procedura di boot. Nel 1998 le case produttrici iniziarono a costruire schede madri con BIOS scritti su chip in grado di essere aggiornati via software. L'autore del virus Win.CIH venne in qualche modo a conoscenza delle specifiche riservate per l'accesso al BIOS e inserì un payload che lo cancellava. Quando questo virus si attiva su una di queste nuove macchine, questa diventa inutilizzabile poiché non più in grado di eseguire la procedura di boot. Inoltre per il ripristino della funzionalità è necessario smontare la scheda madre e sostituire fisicamente il chip del BIOS, con grave perdita di tempo e denaro.

Questi ultimi virus sono stati molto diffusi e lo sono ancora, data la loro capacità di infettare sistemi operativi moderni e le loro tecniche di *stealth*, cioè le loro capacità tecniche di nascondersi all'interno dei file in modo da non modificarne le dimensioni (cercando lunghe zone del file riempite di zeri, oppure, in alcuni casi, suddividendo il codice in frammenti disposti nelle zone con valori nulli anche breve). In effetti, si assiste ad una sorta di selezione naturale dei virus, che non sono più in grado di replicarsi a causa del cambiamento delle caratteristiche del sistema operativo. Per esempio, i virus di boot hanno grandi problemi a riprodursi in un ambiente come quello di Windows 95 che utilizza particolari driver a 32 bit per l'accesso alle periferiche.

Contemporaneamente alla diffusione dei virus, dei quali abbiamo citato alcuni esempi significativi, abbiamo la nascita e la diffusione dei *tool* per lo sviluppo dei virus. Abbiamo già parlato del "Mutation Engine" o "MtE" che è un tool che permette di generare virus polimorfi. Il Mutation Engine comprende un file, MTE.OBJ, che contiene il programma di crittografia, inoltre comprende un programma sorgente di un virus dimostrativo, che si può utilizzare per provare il tool. Si conoscono almeno trentasei virus che usano MtE, che comunque viene rilevato dagli odierni scanner senza problemi.

Nel 1992, ispirandosi al Mutation Engine, un olandese con lo pseudonimo di Masud Khafir inizia a distribuire il "Trident Polymorphic Engine" o "TpE". L'autore, conosciuto come autore di virus importanti, ha distribuito quattro versioni successive del suo programma di crittografia, correggendo via via gli errori trovati nelle precedenti versioni. Si conoscono almeno nove virus molto diffusi che utilizzano TpE, inoltre si è notato che alcuni scanner faticano a rilevarlo.

Il "NuKe Encryption Device" è un altro programma di crittografia per codice eseguibile realizzato nel 1992. È stato realizzato da Nowhere Man, pseudonimo dell'autore, membro del gruppo di scrittori di virus "NuKe", che ha anche realizzato nello stesso periodo il "Virus Creation Lab", un tool completo per lo sviluppo dei virus.

Un altro classico generatore di codice polimorfo è il "Dark Angel's Multiple Encryptor" o "DAME". Esso compare nel 1993 come programma sorgente su una rivista elettronica reperibile su Internet, ad opera di Dark Angel, uno scrittore di virus membro del gruppo "Phalcon/SKISM", anche autore del tool di sviluppo virus PS-MPC. I virus contenenti DAME vengono facilmente rilevati dai programmi anti-virus.

Questi tool per lo sviluppo dei virus si moltiplicano con il passare del tempo, ne possiamo citare alcuni, come "Guns'n'Roses Polymorphic Engine", la cui documentazione è scritta solo in cinese, il "MutaGEN", il "Dark Slayer Mutation Engine", tra i programmi per cifrare il codice; poi "NRLG", "Instant Virus Production", tra i tool completi per lo sviluppo dei virus. Alcuni di questi tool hanno una interfaccia grafica professionale e l'utente può inserire facilmente diversi tipi di payload, dai più innocui ai più distruttivi, oppure selezionare le caratteristiche del virus da generare, come "boot virus", "bomba logica", "cavallo di Troia", crittografato o no, ecc.

Un ultimo tool da menzionare è il "Simulated Metamorphic Encryption Engine" o "SMEG" scritto da Black Baron, autore di virus inglese. Quando comparve, nel 1994, venne utilizzato in alcuni virus considerati molto pericolosi e difficili da rilevare. Infatti SMEG

contiene, oltre ad un programma per cifrare il codice eseguibile, un particolare sistema generatore di codice casuale e privo di utilità; questo codice generato viene inserito nel codice del virus in modo casuale, rendendo difficilissima una ricerca di signature fisse. Infatti non tutti i prodotti anti-virus sono in grado di rilevarlo realmente.

Nel 1995 si presenta un problema completamente nuovo: compare il primo macro virus, "Word.Concept". Si tratta di un virus scritto nel linguaggio delle macro delle prime versioni di Word, il WordBasic, in grado di replicarsi e agire come un comune virus scritto in Assembler. Le macro sono brevi insiemi di operazioni che vengono eseguite sui dati di un programma di gestione testi o un foglio elettronico; col passare del tempo, in alcuni software, questa funzione si è evoluta fino ad avere la possibilità di usare un linguaggio per la scrittura delle macro. In particolare alcuni prodotti Microsoft danno la possibilità di scrivere macro in alcuni linguaggi simili al Visual Basic.

Già nel 1989 il professor Harold Highland aveva parlato in una sua pubblicazione [10] del rischio di macro virus, presentando il risultato di esperimenti da lui eseguiti sul linguaggio di macro del Lotus 123. Nessuno aveva dato però importanza a questo monito, infatti la casa produttrice del Lotus aveva escluso un simile rischio.

Il virus Word.Concept trova un ambiente totalmente impreparato a contrastarlo e diventa velocemente uno dei virus più diffusi al mondo, grazie anche all'assenza di payload dannosi. Inizia quindi la diffusione di macro virus, compaiono virus in grado di infettare altri programmi Microsoft, come Laroux che colpisce Excel, e Tristate che può infettare tutti i prodotti Microsoft: Word, Excel e Power Point. La nuova versione di Word presente in Office 97 contiene un diverso linguaggio di macro, il VBA, Visual Basic for Applications. Word 97 comprende un sistema automatico per la conversone delle macro da WordBasic a VBA, al momento della esecuzione di un file di una precedente versione. Alcuni macro virus superano la conversione, altri no, però nascono subito dei macro virus scritti nel nuovo linguaggio, come Melissa e Explore.ZIP.

Si può notare, con l'evoluzione delle possibilità tecniche dei P.C., una evoluzione parallela delle tecniche utilizzate nei virus.

Finora abbiamo parlato della famiglia dei virus relativa ai P.C., con processore della famiglia Intel o compatibili, che è divenuta la più diffusa. In realtà anche altri sistemi vengono bersagliati, in misura minore, spesso per il breve periodo della loro diffusione come il Commodore Amiga. Negli archivi dell'università di Amburgo si trovano bollettini

riguardanti virus per Commodore Amiga e Atari, oltre che per i conosciuti Apple Macintosh e sistemi Unix[3]. Vi è anche menzione di un malware di tipo particolare, che viene definito in modi diversi ma si può considerare come un worm che si propaga tramite la posta elettronica. Si tratta di CHRISTMA.EXEC Chain Letter, che infetta i mainframe VM/CMS IBM 370, 3080/3090, ES8900, ES9000. È scritto in linguaggio "REXX" un linguaggio di alto livello ed interpretato. Il worm si propaga mandando una copia di se stesso ad altri computer trovati in una sorta di file rubrica del sistema e, in gennaio e dicembre visualizza un grafico a forma di albero di natale con alcune scritte. La data di rilevamento è luglio 1993. [4]

Per quanto riguarda i sistemi Unix, uno degli episodi più classici e famosi è quello dell'"Internet Worm". Il 2 novembre 1988 un programma venne lanciato su più computer collegati ad Internet, questo programma era in grado di rilevare informazioni su computer, reti ed utilizzatori, inoltre era in grado di moltiplicarsi violando altre macchine connesse, usando dei "bug" nella sicurezza di Unix. Il programma era in grado di agire soltanto su macchine Sun Microsystem 3 e su VAX con sistema operativo Unix BSD 4.0, nonostante questo si moltiplicò rapidamente gettando nella confusione gli amministratori di sistema. Si trattava di un programma diviso in due parti: un vettore, che veniva usato per prendere controllo di un altro computer collegato tramite rete, e una parte principale, che veniva trasferita in seguito dal programma vettore e eseguiva diversi compiti per prendere controllo di altre macchine. Il programma worm cercava di scoprire le password di utenti di computer collegati direttamente con il computer infetto; per fare questo aveva un piccolo vocabolario allegato e inoltre tentava le password "classiche" come nome utente, nome account, ecc. Queste informazioni venivano reperite attraverso l'uso del programma finger, una utilità di Unix che riporta i dati degli utenti di altri computer collegati in rete.[5] Si riporta che il worm infettò circa seimila computer sparsi in tutte le nazioni. Secondo le intenzioni dell'autore, il worm avrebbe dovuto replicarsi senza farsi notare, continuando fino a toccare tutte le macchine possibili collegate alla rete. Purtroppo un errore di programmazione faceva si che

[3] Da notare che erroneamente si considera un virus come progettato per funzionare su un determinato tipo di computer; invece è il sistema operativo, oltre che il computer che lo supporta, a ricevere o meno una determinata infezione.

[4] Tratto da un bollettino presente in forma elettronica presso: Virus Test Center, Faculty for Informatics, University of Hamburg; autori: Prof. Dr. Klaus Brunnstein, Vesselin Bontchev, Simone Fischer-Huebner, Wolf-Dieter Jahn; disponibile all'indirizzo ftp.informatik.uni-hamburg.de/pub/virus/texts/catalog/.

[5] Una descrizione estremamente accurata dei dettagli tecnici del worm si trova in [35], disponibile in: ftp.informatik.uni-hamburg.de/pub/virus/texts/tests/catalog/worm_repo.ps.

il programma si replicasse anche all'interno delle macchine che infettava, portandole al collasso; infatti circa 3000 computer collegati ad Internet si bloccarono.

L'autore del worm venne individuato in Robert Morris, studente della Cornell University, figlio di un grande esperto in sicurezza dei computer che ricopriva un ruolo dirigenziale presso il National Computer Security Center. L'esame del codice del worm evidenziava l'assenza di volontà distruttiva dell'autore. Si può supporre che Morris volesse dimostrare, con un atto sensazionale, la gravità dei difetti del sistema operativo Unix; comunque il worm conteneva meccanismi per la violazione delle password e si introduceva in altri computer senza averne il permesso. Morris fu condannato.

Viene riportato[6] che, come conseguenza di questo episodio, e di un altro episodio di pirateria accaduto pochi giorni dopo, durante il quale un computer della rete militare MILNET veniva penetrato da un hacker collegato ad ARPANET, la rete militare MILNET venne disconnessa da ARPANET. Inoltre DARPA istituì, presso il Software Engineering Institute alla Carnegie Mellon University, il CERT – Computer Emergency Response Team, il cui scopo iniziale era quello di fare da quartiere generale per la gestione delle emergenze sui computer della rete ARPANET e MILNET.

Gli errori nella gestione della sicurezza del sistema operativo Unix sono stati cercati e corretti, ma questo problema dei worm torna ad affacciarsi nel periodo odierno, per le nuove caratteristiche tecniche dei computer *home.* In particolare ultimamente si è notato un grande incremento di un tipo relativamente nuovo di malware, si tratta di un worm con alcune caratteristiche del cavallo di Troia. Il primo caso che ha ottenuto menzione da parte dei media è stato "I love you", tecnicamente LoveLetter, e le sue varianti. La posta elettronica consente di allegare ad un messaggio di testo uno o più file, siano essi eseguibili, documenti o di qualunque altro tipo. Il meccanismo di questo malware è di inviare, attraverso gli allegati di un messaggio di posta elettronica, un file eseguibile o un file di testo contenente macro, apparentemente innocui. Il programma contenuto in essi, quindi un cavallo di Troia, prende il controllo del sistema di posta elettronica dell'utente che lo ha eseguito (o aperto nel caso di documenti), e invia una copia di se stesso a tutti gli indirizzi presenti nel file agenda del P.C. infetto, più eventuali payload. (Per una descrizione tecnica consultare 3.2)

[6] E. Spafford, *Crisis and Aftermath,* CACM, Vol. 32, N. 6, June 1989, P.678-687, incluso in [4].

2 - Fattori che determinano la diffusione dei virus

In questo capitolo si tenta di dare una spiegazione dei diversi fenomeni che hanno generato il problema e che ne hanno dato impulso allo sviluppo e alla diffusione.

2.1 - Aspetti psicologici del programmatore e del collezionista di virus

Esistono due figure interdipendenti, di cui si vuole fare un'analisi psicologica e motivazionale, quella del programmatore di virus e quella del collezionista.

2.1.1 - Il programmatore di virus

Per quanto riguarda i primi tempi del fenomeno virus, secondo uno studio effettuato da Sarah Gordon [49], non esiste un generico tipo psicologico dell'autore di virus. L'autrice presenta uno studio su quattro categorie di persone legate allo sviluppo dei virus, che dichiarino di averne almeno scritto uno, poi rilevato "in the Wild" [7]: adolescenti, universitari, adulti, ex scrittori di virus. Essi risultano differire per età, livello sociale, luogo di provenienza, capacità di inserimento sociale, livello culturale, preferenze, comunicatività.

Riguardo alle motivazioni, nessuno pare avere come obiettivo il governo o l'apparato militare, anzi, non ci sono obiettivi specifici. Per le categorie più giovani il nemico non esiste, mentre per le altre il nemico è una raffigurazione in parte astratta della società. La presenza femminile è praticamente assente, non si esclude però che possa nel futuro aumentare, seguendo modelli di sviluppo delle forme di devianza comportamentale giovanile conosciuti. Ci sono inoltre analogie tra la distribuzione di virus a persone inconsapevoli e forme di delinquenza giovanile, riguardo all'evoluzione del disturbo comportamentale attraverso gli anni. Come per il fenomeno della delinquenza non si sa per quale ragione inizi, per quale ragione prosegua in alcuni fino a formare criminali professionisti, e per quale ragione in altri receda. Riguardo alla prevenzione molti autori

[7] Ci si riferisce a "The WildList International Organization", www.wildlist.org, organizzazione internazionale che rileva i virus diffusi e in stato di riproduzione attiva. Una descrizione dettagliata in 5.5.3.

propugnano la teoria della deterrenza e auspicano la presenza di leggi molto severe, che puniscano gli scrittori di virus con pene molto severe. Comunque non è certo che questa sia la migliore strategia, infatti, vi sono autori che affermano che una disapprovazione sociale da parte di amici, oppure di parenti, sia un deterrente maggiore di quello delle pene previste dalle leggi.[8] Inoltre il sociologo Jack Katz afferma che la seduzione del crimine è la prima motivazione degli atti anti-sociali.[9]

Il supporto legislativo è comunque irrinunciabile. Il dottor Bontchev, nella sua pubblicazione "The Bulgarian and Soviet Virus Factory" descrive la situazione bulgara nel periodo attorno agli anni '90; non esiste in quel periodo una legge che vieti di produrre né di distribuire programmi atti a danneggiare i sistemi informatici. Non esiste inoltre nemmeno una legge che tuteli o definisca la proprietà dei dati e del copyright dei programmi. Il risultato, come descritto dal dottor Bontchev, è che la Bulgaria è stata in quegli anni la culla della maggior parte dei virus del periodo. Anzi, la pubblicazione citata afferma che in Bulgaria la produzione di virus era "uno sport nazionale", e questo era dovuto alla scarsa valorizzazione del lavoro intellettuale del programmatore, quindi vi era una sorta di frustrazione e di desiderio di rivalsa contro la società; per non parlare ai problemi dovuti alla mancanza totale di una disciplina sul copyright, perciò un programma non poteva vendere più di un paio di copie. Scattarono poi dei fenomeni psicologico sociali, che portarono a considerare in certi ambienti persone come "Dark Avenger", famigerato autore di molti virus, una sorta di eroi. La mancanza di leggi permise poi la fondazione di diverse BBS private che scambiavano in perfetta libertà sorgenti di virus. Una situazione analoga, anche se molto meno marcata, era presente in Russia.

In effetti, si nota che, dal punto di vista legislativo, molte nazioni hanno dovuto adeguarsi e evolversi per comprendere una vasta gamma di reati informatici pochi anni prima inesistenti. In generale è considerato un reato quasi ovunque distribuire programmi capaci di danneggiare sistemi informatici. Non in tutte le nazioni, invece, è illegale detenere o creare tali programmi. In Italia, riguardo ai virus, l'articolo 615-quinquies del codice penale recita: "Chiunque diffonde, comunica o consegna un programma informatico da lui stesso o da altri redatto, avente per scopo o per effetto il danneggiamento di un sistema informatico o telematico, dei dati o dei programmi in esso contenuti o ad esso pertinenti, ovvero l'interruzione totale o parziale, o l'alterazione del suo funzionamento, è punito con

[8] Dalla bibliografia di [49]: Journal of Criminal Law and Criminology, Van den Haag.
[9] Dalla bibliografia di [49]: Jack Katz, *Seduction of Crime*, Basic Books, New York, 1988.

la reclusione fino a due anni e con la multa sino a lire venti milioni". Non si accenna alla detenzione, però per il resto la legge è abbastanza restrittiva: sono esclusi siti dove si possano conservare e distribuire virus. (Nonostante questa legge, all'incirca dal maggio 2000 si trova, anche nei news server di alcuni grandi ISP italiani, il gruppo di discussione "free.it.codice.virale", vedere 5.2). Negli USA invece, per effetto della legge sulla libertà di espressione, vi sono molte organizzazioni come quelle citate, che pubblicano liberamente sorgenti di virus su siti e riviste elettroniche, newsgroups ecc. [45]

Si tratta quindi comunque di persone che rischiano con il loro operato di rovinarsi la vita avendo problemi legali non indifferenti. Al riguardo uno studio afferma che, del numero degli scrittori di virus, l'85% sono fondamentalmente delle persone tranquille e non vogliono correre rischi, il rimanente 15% è distribuito uniformemente a partire da chi è disposto a correrre solo rischi minimi fino a chi non si cura minimamente delle conseguenze del proprio operato. [16]

La situazione degli autori di virus si è comunque evoluta. In un secondo studio, sempre della Gordon, intitolato "The Virus Writer II", si analizza la situazione del "New Age Virus Writer". Si può dare una delimitazione più precisa dell'età che risulta in media di 20-21 anni, inoltre si nota una formazione di una cultura dell'autore di virus, una sorta di "anarchici elettronici", molto simili al cyberpunk, fenomeno per cui si tende a trasformare il fenomeno hacker in una cultura o "underground"- cultura. Gli scrittori di virus sono spesso riuniti in gruppi, come "NuKE", "VLAD", e altri; inoltre la zona di produzione si è spostata, almeno riguardo alla massa, dall'America, Bulgaria, Canada alla Svezia, Norvegia, Australia. Secondo le interviste fatte dalla Gordon a questa nuova generazione di scrittori di virus, non c'è più il giovane annoiato o alla ricerca di fama, ma un giovane arrabbiato con la società, che si sente in lotta con essa ed esprime la sua lotta in questa maniera. Inoltre, si tratta di persone più consapevoli della loro capacità tecnica e consapevoli della necessità di avere questa capacità, per creare virus capaci di sostenere la lotta contro gli anti-virus. Di questo poi si parlerà nel capitolo tre, comunque l'autore di virus deve essere in grado di saper maneggiare oggi avanzate tecniche di programmazione, e lo considera un aver raggiunto una meta importante, che richiede tempo e studio. La possibilità poi di avere moltissimo codice sorgente a disposizione attraverso i siti web (vedere paragrafo successivo), sempre secondo la Gordon, porta a demistificare e a legalizzare, almeno dal punto di vista psicologico, il fenomeno dei virus. Inoltre è scomparso il problema dello scambio del codice tra scrittori di virus, che una volta usavano le BBS segrete conosciute

solo da loro per scambiarsi virus sorgenti. Oggi invece il virus deve solo essere scaricato dal sito più aggiornato. Molto interessante è poi quanto riportato da Secure Computing riguardo al virus Boza: si riporta un'intervista effettuata ad uno degli autori che parla di verifica del virus con beta-tester, amici che provano il virus su loro computer per verificarne la validità.[10] Si tratta quindi di virus molto avanzati, scritti da persone in grado di lavorare e guadagnare molto con le loro capacità. Per fare un esempio, il virus WM.Concept, un virus di macro molto diffuso, segue la Hungarian naming convention, un insieme di dettagliate istruzioni per attribuire il nome alle routines e alle variabili. Se da una parte i vecchi scrittori di virus hanno avuto la loro evoluzione, il loro apice e stanno smettendo di produrre per aver finito il proprio periodo fecondo, questi nuovi autori non presentano la caratteristica di andare oltre i limiti di età, poiché sono già uomini adulti e professionalmente completi.

Per interrompere questa tendenza, anche se non si vuole introdurre censure, è realmente auspicabile che il codice sorgente dei virus non si possa trovare disponibile sui siti del Web. Per ottenere questo si possono introdurre nuove legislazioni, sensibilizzare o ri-sensibilizzare la comunità informatica riguardo all'illiceità e alla pericolosità di questi comportamenti. D'altra parte, al giorno d'oggi, esistono associazioni che combattono strenuamente contro siti che pubblicano fotografie di pornografia infantile. Questo può dare un'idea della mancanza di etica di alcuni gestori di siti odierni, specialmente in alcune nazioni.

Un altro comportamento da notare e sconsigliare è quello di molti tecnici che al giorno d'oggi giocano con il sorgente dei virus, a rischio di generare nuove varianti di virus. L'impatto di un comportamento di questo genere può essere devastante per una compagnia che si occupi di informatica.

2.1.2 - Il collezionista di virus

Dei collezionisti si parla pochissimo e sono una realtà relativamente ignorata. Per descriverla meglio analizziamo la situazione dei siti Internet dedicati ai virus, che si occupano di questo argomento da un punto di vista diverso da quello dei ricercatori e produttori di anti-virus e di sicurezza informatica. Esiste una vasta gamma di siti

[10] Riferimento a "Secure Computer, *BOZA or Bizatch, the Hype Continues*, Giugno 1996" contenuto in [50].

underground di questo genere. Vi sono siti che contengono tools per creare virus, tools per generare virus polimorfi, codici sorgenti di virus e file binari con virus pronti all'uso. Questo in modo più o meno esplicito e approfondito a seconda della severità della legislazione del paese che ospita i siti. Praticamente nessuno di questi siti però afferma di voler diffondere infezioni o incita a questo. Si parla invece di diffondere conoscenza informatica, una conoscenza specifica sulla produzione dei virus, ma fine a se stessa. Una conoscenza tipica dell'*hacker*, non inteso come esperto di informatica, personaggio che cerca di apprendere tutti i segreti dei sistemi di sicurezza dei computer per poterli violare, al solo scopo di dimostrare a se stesso ed agli altri la sua capacità, quindi senza nessun intento distruttivo. Invece i *lamer,* dall'inglese to lame, azzoppare, sono coloro che usano i virus e, in generale, la loro conoscenza informatica per danneggiare.

I collezionisti in realtà non commettono nessun atto rivolto a danneggiare altri individui, infatti, si leggono nei loro siti dichiarazioni di scarico di responsabilità del tipo "il materiale presente in questo sito è finalizzato solo alla ricerca". Spesso vi sono anche avvisi che invitano gli utilizzatori ad informarsi sulla legislazione vigente nel proprio paese d'origine, e a disconnettersi nel caso la legislazione locale proibisse la detenzione di codice sorgente virale. Finalizzati a questo motivo sono sorti anche molti newsgroup di collezionisti, tra cui il già citato free.it.codice.virale, ma ve ne sono molti altri. Una analisi dettagliata dei siti web e dei newsgroup dei collezionisti di virus è presente nel paragrafo 5.2.

2.2 - Provocazione

Tornando ai collezionisti, il problema è la tendenza psicologica all'emulazione di alcuni individui e la possibilità di trovare codice dannoso in siti accessibili. Si è già segnalato il caso della pubblicazione del codice sorgente del virus Vienna (Cap. 1) da parte di Ralf Burger nel libro "Computer virus, una malattia ad alta tecnologia". Il risultato fu la comparsa di moltissime varianti del virus, segno questo che molte persone ricompilarono il codice, realizzarono il virus e, alcuni, lo distribuirono. Viene, infatti, considerato estremamente dannoso, presso i ricercatori del settore, pubblicare codice sorgente di virus e

altro malware. In Italia, poi, visto il testo della legge, suscettibile di diverse interpretazioni, una pubblicazione simile potrebbe risultare illegale.

Se in passato la diffusione del sorgente dei virus poteva essere considerata una azione estremamente difficile e operata nel nascondimento, al giorno d'oggi vi sono moltissime pagine web come quelle descritte nel paragrafo precedente, che riportano, per ragioni di "diffusione della conoscenza", sorgenti di virus in grande quantità (anche migliaia) e tools per generarne altri. Ora, chiunque desideri prelevare un virus sorgente, può farlo liberamente. Poi lo può compilare, infatti, spesso i file sorgenti di virus contengono informazioni sul tipo e versione dell'assemblatore o compilatore da utilizzare. Inoltre contengono frequentemente una descrizione dettagliatissima di come il virus operi, con il commento tecnico di ogni linea di codice. Quindi si può prelevare virus, si può compilarli e generare il programma eseguibile "portatore", cioè un file eseguibile contenente solo la routine del virus, si può esaminare i dettagli tecnici e capire come realizzare altri virus. Il virus Win.CIH, per esempio, si trova sotto forma di sorgente e, all'interno, si trova ben evidenziato il codice in grado di cancellare il BIOS dei computer recenti. Analogamente, pochissimi giorni dopo che i media parlarono dell'infezione da parte del worm "I love you", sul newsgroup "alt.comp.virus.source.code" ne era già disponibile il sorgente integrale e funzionante. Si rilevò, a breve, un aumento significativo di varianti di tale worm.
Con questo non si vuole assolutamente proporre censure, ma si desidera evidenziare che, se in passato poteva essere molto difficile e complesso trovare codice sorgente di virus da modificare, oggi non è più così. Non si può certamente suggerire di considerare un reato il collezionare o il programmare virus "per uso personale", i ricercatori delle società anti-virus al contrario devono avere una collezione la più completa possibile ed anche una profonda conoscenza dei meccanismi di funzionamento dei virus. È però necessario evidenziare che la reperibilità del codice sorgente rende facile il riciclaggio di codice virale per creare un proprio personale virus, come la reperibilità di tool di sviluppo virus completi, quali NRLG, IVP, PS-MPC, ecc. (vedere Cap. 1). Per coloro che sono interessati, oltre che allo studio dei meccanismi virali, alla diffusione dei virus, come resistere alla tentazione di utilizzare tutto questo?

2.3 - Aspetti economici

La letteratura riporta pochissimi casi in cui si sia dimostrato che la diffusione di virus avesse in qualche modo a che fare con vendette, estorsioni, tentativi mirati di danneggiare una azienda o qualunque altra azione dolosa rivolta verso un obiettivo specifico.

D'altra parte uno degli effetti indotti dalla diffusione dei virus, che si può ritenere iniziata intorno alla metà degli anni ottanta, è stato quello di fare nascere molte associazioni di ricerca sul fenomeno e di fare sorgere un industria prima totalmente inesistente, quella degli anti-virus. Il dottor Bontchev, nelle sue pubblicazioni, in particolare in [45], descrive le difficoltà necessarie a produrre un software anti-virus di buona qualità. Le conclusioni sono che, al giorno d'oggi è quasi impossibile per una azienda anti-virus iniziare da zero; in realtà, è problematico per una azienda anti-virus, anche riuscire a mantenersi attiva sul mercato. I problemi sono molteplici, principalmente mantenersi aggiornati sui nuovi virus e mantenere la velocità di scansione del proprio prodotto competitiva. Si tratta di difficoltà non indifferenti, poiché il numero attuale di virus riconosciuti da uno scanner per signature è di circa 45000 (luglio 2000), questi virus non sono in realtà tutti attivi e diffusi, esiste il problema dei virus estinti e dei cosiddetti "Zoo virus". Come si è già accennato e si tratterà ampiamente nei successivi capitoli, esiste una lista, aggiornata in continuazione da volontari, detta la WildList, che contiene i nomi dei virus più diffusi al momento. I virus non presenti nella lista possono certamente essere presenti in qualche parte, ma statisticamente la probabilità di incontrarli è molto minore e tende col passare del tempo ad essere nulla; infatti, dopo un certo periodo dalla comparsa nella lista, dopo che non vi siano più state segnalazioni, un virus viene cancellato. Si può considerare che un gran numero di virus in passato denunciati e attivi, siano ora estinti. Inoltre vi possono essere virus che sono stati presenti solo in ristrettissime zone. I cosiddetti Zoo virus, sono poi virus che sono esistiti solo nei laboratori dei ricercatori, ricevuti magari da qualche smanioso di fama, ma non sono mai stati realmente diffusi. Nonostante questo gli scanner comprendono tutti i virus possibili, per una sorta di necessità di immagine secondo cui un prodotto è migliore se rileva più virus di un altro. In realtà vedremo che non è così.

Riguardo alle implicazioni economiche intese come danni prodotti dai virus, vi è un articolo della APB News [60], che segnala le perdite portate dai virus nel solo 1999 a 12 miliardi di dollari. In effetti, abbiamo visto che i virus non sono più il problema marginale di alcuni anni fa. Si tratta di un problema molto serio che richiede una prevenzione particolare, che si attua anche con campagne e corsi di istruzione, dove si trasmettono le precauzioni per prevenire i "contagi". Alcune università hanno preparato dei corsi appositi di sensibilizzazione sul problema dei virus, anche a fini di prevenzione; la ricerca si è poi occupata di stilare linee guida preventive indirizzate alle organizzazioni e alle corporazioni. Si veda al riguardo il paragrafo 3.2, oltre a: [28], [4]: p. 508, "Teaching Students About Responsible Use Of Computers;"; [55], [51].

Uno dei pochi casi in cui vi sono state prove di un uso di virus a scopo di vendetta contro un'azienda è quello di Donald Gene Burleson, dipendente dell'United Services Planning Association Inc., il quale svolgeva mansioni di programmatore e addetto alla sicurezza del sistema informatico. Licenziato senza giusta causa, il 18 settembre 1985, inserì un programma autoriproducente nel sistema informatico aziendale che avrebbe effettuato diverse operazioni distruttive ad intervalli di tempo. Il programma venne rilevato dopo i primi malfunzionamenti e Burleson fu incriminato e condannato al pagamento dei danni e a sette anni di libertà vigilata.

Vi sono altri episodi sporadici di denuncie contro individui che avrebbero infettato appositamente un sistema per vendetta. In rare occasioni, sono state inserite all'interno del virus stringhe con intenti di vendetta; le stringhe riportate di seguito, per esempio, sono state rilevate in un virus.[11]

«BEN VENUTI ALLA FIERA DI MILANO GIUGNO This virus for Professori Francesco Gardin and the University of Milano June 9, 1989»

Infine, un fatto relativamente recente è quello della vendita di virus per posta o in altra maniera. Ne abbiamo un resoconto sia in [45], che in una pubblicazione di Sarah Gordon. [50]

[11] [1], p.40.

3 - Aspetti tecnici

In questo capitolo si descrivono le caratteristiche tecniche dei principali tipi di malware, i mezzi attraverso cui le infezioni si diffondono e le precauzioni che si possono prendere per ridurre il rischio di infezione, infine, i meccanismi utilizzati dai virus per nascondersi ed eludere i programmi anti-virus.

3.1 - Meccanismi di funzionamento dei virus

Si distinguono molti tipi di virus, o meglio di malware, in base al sistema di riproduzione o di diffusione e alle caratteristiche tecniche. In questo capitolo si prendono in esame, eccetto poche eccezioni, solo virus riguardanti Personal Computer. La grande diffusione dei P.C. ha portato un forte incremento dei relativi virus; il numero è tale da permetterne, oltre all'analisi tecnica, una suddivisione in categorie. La stessa cosa non avrebbe senso per sistemi quali il Macintosh, o meno ancora per il malware indirizzato a sistemi Unix/Linux o mainframe. In ogni modo anche questi si possono inserire nelle categorie presentate.

Uno specchietto riassuntivo può essere il seguente:

Tipi di Malware

Virus

1. suddivisione secondo i meccanismi di infezione:
   - Virus del settore di boot / MBR
   - Virus dei file:
     - Eseguibili
     - Overlay
     - Device
     - File di sistema
     - Macro
2. suddivisione secondo le caratteristiche tecniche:
   - Polimorfici (auto-cifranti)
   - Polimorfici (auto-modificanti)

- Residenti in memoria
- Stealth
- Progettati per il s.o. Dos
- Progettati per il s.o. Windows

Worm

- Worm trasmessi via Trojan contenuto in attachment E-mail sotto forma di:
  - Eseguibile
  - Documento con macro
- Worm trasmessi con script IRC
- Worm trasmessi via comunicazione diretta Internet

Trojan o cavalli di Troia

- Bombe logiche
- Bombe a tempo
- Backdoor

Non si desidera dare un inquadramento rigido dei tipi di virus, o di malware, con questa suddivisione, ma piuttosto un inquadramento generale. Molti malware corrispondono ad un insieme di categorie piuttosto che ad una sola; le diverse categorie, inoltre, non si sono presentate contemporaneamente, ma piuttosto c'è stata un'evoluzione parallela del malware, dei sistemi operativi, dei mezzi di calcolo e degli anti-virus.[6]
Si noti come il software sviluppato per la rilevazione e rimozione del malware sia sempre definito anti-virus; il termine "virus", nato per definire un particolare tipo di software dannoso, ne è diventato il sinonimo generale.

3.1.1 - Boot virus[12]

Il primo tipo di virus analizzato, ed anche il primo comparso, è quello del settore di boot. Molti tipi di computer hanno una parte del sistema operativo residente su ROM, su chip dedicati, un'altra parte invece è residente su memoria di massa, come dischetti o dischi rigidi. Questo per permettere un aggiornamento ed un'intercambiabilità delle versioni del sistema operativo. È comunque necessario indicare alla porzione di sistema operativo residente su ROM dove deve prelevare il resto del sistema operativo stesso. Per fare questo, molti computer di piccole dimensioni, come i Personal Computer, utilizzano la tecnica del

[12] La bibliografia principale per quanto riguarda il seguito del capitolo si trova in [68], [69], [70], [57], [58], [59].

bootstrap: la ROM ricerca su un'unità di memoria di massa, oppure su più unita in sequenza, un particolare codice sul primo settore del disco. Dove questo codice sia presente, esso è accompagnato da una breve porzione di codice eseguibile che si occupa di caricare il resto del sistema operativo, andando a cercare i file che lo contengono. Questo codice eseguibile si chiama "codice di boot", dal nome della tecnica usata per il caricamento del sistema operativo. Un disco contenente il sistema operativo ed in grado di avviare questa operazione si chiama "disco di avvio". Una procedura analoga avviene nel caso di unità a disco rigido: il settore zero contiene, invece del settore di boot, il "Master Boot Record" o MBR, che quando è eseguito, legge ed esegue a sua volta il primo settore della partizione attiva dell'unità.

Questo genere di virus modifica il codice di boot, o l'MBR, e vi si inserisce. Può usare tecniche diverse: per quanto riguarda il codice di boot, semplicemente il virus si sostituisce al codice di boot e si occupa anche di caricare il sistema operativo. Oppure può spostare il codice di boot in un altra parte del disco e sostituirvisi, questo è tipico dei virus capaci di infettare settore di boot e MBR; infatti, esso contiene dati che non sono standard ma variano in funzione delle caratteristiche dell'unità. In questo caso il virus si sostituisce all'MBR o al codice di boot e memorizza quest'ultimo in un altro punto dell'unità. Il virus Stoned, per esempio, memorizza il settore a cui si sostituisce nell'ultimo settore dello spazio del disco destinato alla directory radice. Altri, come il Ping Pong, spostano il contenuto del primo settore in un settore libero del disco e poi lo marcano nella FAT come settore danneggiato.[13] Altri ancora memorizzano il settore spostato nell'ultimo settore del disco, nella speranza che non venga mai raggiunto. Anche se sono stati citati solo virus relativi ai P.C., il meccanismo descritto vale per ogni tipo di computer che usi la tecnica del "disco di avvio" o "disco di boot".

I virus di questo tipo sono composti da diverse parti:

- Riproduzione: il virus controlla se vi sono altre unità collegate, oltre quella in cui esso è residente, e modifica per ognuna di esse il settore di boot.

[13] La FAT è una sorta di mappa del disco che, tra le altre cose, tiene traccia dell'utilizzo di ogni settore e indica con un codice speciale se il settore è stato rilevato come difettoso in fase di formattazione.

Nota: non è necessario che il virus verifichi la presenza del sistema operativo, infatti, ogni unità che può essere utilizzata per il bootstrap riserva il primo settore a questo fine. Se si tenta di fare un bootstrap da una unità che non contiene il codice di boot o che non contiene i file del sistema operativo viene visualizzato un messaggio di errore e si interrompe la procedura in attesa dell'intervento dell'operatore. Il virus quindi infetta qualunque unità abbia a disposizione; se questa era utilizzata per il bootstrap da quel momento essa propaga anche l'infezione. Se non era preposta a questo, il settore virale che vi è stato trasferito potrebbe comunque essere eseguito per errore: basta che, nel caso dei dischetti, un dischetto venga dimenticato nell'unità in fase di accensione. La porzione di s.o. residente su ROM, che nel P.C. si chiama BIOS (Basic Input Output System), segnalerà l'errore, ma nel frattempo il virus avrà già infettato ogni altra unità collegata.

- Installazione in memoria: questa fase non è sempre presente. Il virus si installa in memoria e diventa residente, in questo modo esso può infettare ogni disco che viene inserito nell'unità anche non in fase di boot. Le caratteristiche tecniche di questo tipo di attacco verranno descritte in seguito.

- Caricamento del sistema operativo: per ultimo il virus legge il settore che è stato nascosto nell'unità e lo esegue; ovvero, nel caso non abbia memorizzato il settore in qualche posto, procede esso stesso alla lettura ed esecuzione dei file del sistema operativo, come un normale codice di boot.

3.1.2 - Virus dei file di codice.

In questo paragrafo si vogliono trattare tutte le tipologie (conosciute) di virus che sono in grado di infettare un file contenente codice eseguibile, eccetto i macro virus che verranno trattati nel paragrafo successivo.

Questo genere di virus è molto vario, come già si è potuto notare nello specchietto iniziale, sia per i tipi di file bersaglio che per le tecniche utilizzate. Si noti che la descrizione

che segue riguarda in particolare il sistema operativo DOS e Windows su P.C., ma molte delle tecniche descritte di seguito possono essere utilizzate da altri sistemi.

Si tenga presente il seguente specchietto riassuntivo delle caratteristiche dei diversi tipi di file contenenti codice eseguibile (tralasciando il caso delle macro),

- COM: file non superiore a 64 KB di lunghezza, privo di particolari informazioni di memorizzazione, inizia subito con il codice.

- EXE: file di codice eseguibile, inizia con l'EXE Header, che contiene informazioni dettagliate tra cui valori da assegnare ai vari registri in fase di inzializzazione, indirizzo di inizio, indirizzo dello stack, indirizzo dell'area dati, ecc.

- OVL: file di overlay, per evitare errori di memoria esaurita i file eseguibili di notevole lunghezza in certe versioni del DOS venivano spezzati in un file EXE principale e una o più parti richiamate al bisogno, OVL, che contengono in generale delle subroutine.

- SYS: file device, si tratta di file contenenti un programma in codice eseguibile che generalmente viene mantenuto residente in memoria e fornisce delle subroutine utilizzate dal sistema operativo per scopi particolari, come la gestione di un dispositivo esterno, es. mouse.

- EXE (Portable Executable): file specifico per l'ambiente Windows, permette l'interfacciamento con le funzioni di libreria di Windows. Contiene una intestazione e molti dati aggiuntivi organizzati in forma complessa.

- DLL: dynamic link library, file specifico dell'ambiente Windows, si può considerare un analogo del tipo OVL.

- VXD: file di periferica virtuale, contiene codice utilizzato per interfacciare il sistema con le periferiche, siano esse fisiche o virtuali.

- OCX: file specifici dell'ambiente Windows, possono contenere controlli Activex ed altro software.

Il primo tipo apparso in ordine di tempo, ed anche il più semplice, è il virus del file COM. I virus Vienna e Cascade sono i primi di questo tipo. Normalmente il virus si inserisce in coda al file e aggiunge in testa al file una istruzione di salto alla prima istruzione del virus. Si dà, di seguito, una descrizione schematica ma precisa del meccanismo di infezione.

In figura 1 si dà una rappresentazione schematica di come viene alterato il contenuto del file, da un punto di vista fisico, invece in figura 2 da un punto di vista logico.

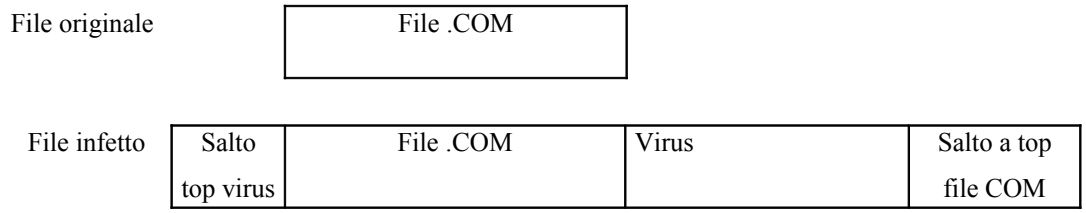

Figura 1

contenuto originale	Dopo l'infezione
	-salta a prima istruzione virus
-prima istruzione programma	**-prima istruzione programma**
-seconda istruzione programma	**-seconda istruzione programma**
-terza istruzione programma	**-terza istruzione programma**
.....	**.....**
.....	**.....**
-ultima istruzione programma	**-ultima istruzione programma**
	-prima istruzione virus
	-seconda istruzione virus
	.....
	-salta a prima istruzione programma

Figura 2

Un programma infetto, al momento della esecuzione, esegue una ricerca nella directory attuale, oppure in alcuni casi anche in altre directory, rileva ogni file compatibile con l'infezione, lo apre e, normalmente, verifica se il file è già stato infettato; nel caso che non lo sia:

- apre il file in scrittura
- valuta la lunghezza del file e aggiunge in testa una istruzione di salto all'offset immediatamente successivo a quello dell'ultima istruzione
- si posiziona in coda al file e aggiunge ad esso una copia della propria routine virale
- aggiunge in coda alla routine virale una istruzione di salto all'offset immediatamente successivo a quella della prima istruzione, aggiunta dal virus stesso.

Un file EXE contiene sempre del codice eseguibile, ma il suo formato è più complesso. Infatti, il primo virus in grado di infettare un file di questo genere è posteriore agli infettori di file COM, e a sua volta anteriore alla successiva evoluzione, quella cioè di codice virale in grado di contagiare sia file EXE sia file COM.

Il primo virus in grado di infettare un file EXE è stato YANKEE, in seguito sono apparsi dei virus in grado di infettare entrambi i tipi di file eseguibile.

Le operazioni aggiuntive che il codice virale deve eseguire sono le seguenti: in fase di infezione, deve leggere l'EXE Header del file bersaglio e modificare per prima cosa l'indirizzo della prima istruzione da eseguire, in modo da puntare al virus che, anche in questi casi, generalmente, è inserito in coda al file. Inoltre, spesso il virus crea un suo ambiente per la gestione dei dati alla fine del codice, nello heap, al momento in cui viene caricato in memoria, per evitare di intaccare i dati originali del programma infettato; quindi è normalmente necessario, per il virus, modificare alcuni registri di sistema, dopo aver salvato in qualche parte della memoria i valori originali.

Dopo aver eseguito le procedure di infezione, il virus provvederà a ripristinare i valori dei registri in modo da permettere al programma originale di funzionare. Inoltre, a differenza del caso del file di tipo COM, non viene effettuato un salto alla seconda o n-esima istruzione del file, ma viene eseguito un salto all'indirizzo di inizio del programma originale , che normalmente si trova nell'EXE Header, e quindi il virus deve aver provveduto, in fase di infezione, a memorizzarlo in qualche parte del file, normalmente un area dati aggiuntiva del virus.

In figura 3 si dà uno schema molto semplificato di un file normale e uno infetto.

EXE Header	EXE Header (infetto)
prima voce header	prima voce header
seconda voce header	seconda voce header
.....	.....
indirizzo inizio programma (XXXX)	indirizzo inizio programma (YYYY)
.....	.....
Fine EXE Header	Fine EXE Header
Inizio codice	Inizio codice
istruzione codice	istruzione codice
istruzione codice	istruzione codice
.....	.....
.....	.....
XXXX: istruzione codice (inizio programma)	XXXX: istruzione codice (inizio programma)
.....	.....
Fine codice	Fine codice
Area dati	Area dati
.....	.....
Fine area dati	Fine area dati
Fine file	Inizio codice virus
	YYYY: prima istruzione virus
	.....
	YYYY+n: n-esima istruzione virus
	Fine codice virus
	Area dati virus
	.....
	indirizzo inizio originale XXXX
	.....
	Fine virus
	Fine file

Figura 3

Questa è una possibile tecnica. Ci sono altri virus che sovrascrivono la prima parte del file e memorizzano da un altra parte, tipicamente in coda, la parte iniziale sovrascritta. Al momento dell'esecuzione il virus, dopo avere eseguito le sue operazioni, ripristina la parte iniziale e la esegue normalmente.

Per quanto riguarda i virus in grado di infettare sia file COM sia EXE, il virus si comporta diversamente secondo il tipo di file che lo ospita ed esegue operazioni diverse secondo il tipo di file bersaglio.

Come esempio si esamina il funzionamento del virus Dark.Apocalypse:

- infezione COM: aggiunge in testa al file un insieme di linee di codice che saranno eseguite per prime; esse, in fase di infezione, modificano dei particolari campi, all 'interno del file, per indicare che il tipo di file infettato è COM. In seguito vi è un salto al fondo del file, all'inizio della routine virale principale. Essa rileva i valori inseriti nei campi di verifica, ed in base a questi differenzia il suo comportamento.
- infezione EXE: l'insieme di linee di codice non viene posto all'inizio ma si trova alla fine del file; l'EXE Header viene modificato per eseguire per prima cosa la routine virale principale. Vengono salvate le informazioni relative ai registri originali dell'EXE Header.

In seguito, con la comparsa del sistema Windows 95 e Windows NT, è nato un nuovo tipo di file eseguibile, il file EXE Portable Executable. Esso contiene delle particolari tabelle e un header particolare che permette l'interfacciamento con le funzioni di Windows. Infatti, Windows per interfacciarsi con le periferiche non usa le funzioni del BIOS, ma alcuni device driver detti periferiche virtuali. Essi hanno una forma di protezione, in particolare su Windows NT, e non permettono ad un programma utente di accedere direttamente alle periferiche. Esistono alcuni virus, come per esempio Chernobyl (tecnicamente Win95.CIH), che riescono, usando delle caratteristiche non documentate di Windows a ottenere i cosiddetti "privilegi Ring0". Il nucleo del sistema operativo Windows NT, detto kernel, è composto di anelli (ring) di privilegi per accedere alle funzioni del sistema operativo; un programma ha di solito privilegi "Ring3", può quindi richiamare funzioni per accedere alle periferiche ma non può accedervi direttamente. Ottenendo il privilegio Ring0, il virus è in grado di accedere in modo diretto alle periferiche e di eseguire delle operazioni riservate al sistema operativo. Il virus che abbiamo citato, Chernobyl, riesce, in questo modo, ad attivare le procedure per l'aggiornamento del BIOS, nelle schede madri di ultima generazione, e a cancellarlo.

Esistono altri due tipi di virus, dalle caratteristiche tecniche non particolarmente rilevanti, i quali si possono considerare come un particolare tipo di virus di codice eseguibile. Vengono trattati brevemente nel seguente specchietto.

- Virus overwrite. Normalmente un virus di codice eseguibile si occupa di trasformare il file bersaglio per creare uno spazio in cui memorizzare la copia di se stesso, e poi esegue l'infezione. Il virus overwrite sovrascrive semplicemente la parte iniziale del file bersaglio con il suo codice. In questo modo il programma infettato non sarà più in grado di svolgere le funzioni originali.

- Virus companion. Si basano su un meccanismo del DOS dei personal computer. Se nella directory attuale esistono due file eseguibili con lo stesso nome, ma uno con estensione COM e uno con estensione EXE, viene data la precedenza al file con estensione COM. Questi virus, quindi, creano una copia di se stessi con il nome uguale a quello del file EXE da infettare, ma con estensione COM; possono poi rendere nascosto il file originale, in modo da non destare sospetti. Infine il virus può richiamare, come ultima istruzione, il programma originale infettato.

3.1.3 - Virus residenti e di sistema operativo

I tipi di virus visti finora infettano altri file nel momento dell'esecuzione del virus, un brevissimo istante prima di eseguire il programma infettato. Quindi non è possibile per un virus infettare molti file in una volta sola; certo, con diverse esecuzioni l'infezione si espande esponenzialmente, ma alcuni tipi di virus hanno cercato un sistema ancora migliore. Sono nati cosi virus residenti in memoria. Quando questi virus vengono eseguiti, si installano in memoria, alterando qualche funzione di sistema del DOS, per esempio quella che gestisce la apertura dei file, oppure quella che gestisce il caricamento ed esecuzione di un file eseguibile. Ogni volta che un file eseguibile viene letto, quando questi virus sono in memoria, esso viene anche infettato. In questo modo, anche dopo una sola attivazione di un file contenente un virus, tutto il contenuto del disco può venire infettato. Uno dei primi virus di questo tipo è il Dark Avenger, e la tecnica è risultata molto più infettiva delle tecniche precedenti, infatti, questo genere di virus è stato denominato "fast infectors".

Una naturale evoluzione di questa tecnica è stata quella di cercare di infettare anche i file del sistema operativo, tipicamente command.com. In questo modo il virus inizia subito a riprodursi, senza bisogno di aspettare che venga eseguito un file infetto. In questa categoria, ci sono virus che puntano subito a cercare il COMMAND.COM e dopo ad infettare gli altri file; altri invece non seguono uno schema fisso ma infettano "a caso" i file che incontrano, a differenza però di altri virus, che evitano di infettare il file COMMAND.COM, questi lo infettano. Un famoso virus, Lehigh, infetta per l'appunto il file COMMAND.COM, inoltre esso si installa in una parte del file riempita con zeri, che viene utilizzata come area dati temporanea. In questo modo il file infetto non varia la sua lunghezza. (Vedi paragrafo 3.3)

3.1.4 - Infettori dei file di overlay e dei device driver

Si tratta di file particolari, che contengono un insieme di routine in codice eseguibile. I file di overlay contengono normalmente un insieme di funzioni aggiuntive, che vengono utilizzate da file di programmi come libreria. I device driver contengono invece un insieme di funzioni che debbono essere installate in memoria e verranno utilizzate per interfacciare con il sistema le periferiche esterne.

Le estensioni più comuni di questo tipo di file sono: per i device, SYS per il DOS e VxD per Windows; per gli overlay, OVL per il DOS e DLL per Windows.

I virus in grado di infettare questo genere di file, normalmente sono anche in grado di infettare file di altri tipi, i quali fanno da veicoli di infezione (non è comune infatti scambiare file di overlay o device driver). I file driver invece, essendo spesso eseguiti all'inizio, dal sistema operativo, rendono l'infezione residente in memoria da subito. In questo modo i virus prendono il controllo del sistema e possono infettare molti più file, come descritto nel paragrafo precedente.

Il virus Navrhar, per esempio è un virus in grado di infettare sia documenti con macro (vedere paragrafo seguente) che file VxD. I file VxD hanno il formato interno LE EXE (Linear Executable), qualcosa di analogo ai PE EXE ma con molte differenze. Questo

tipo di file è estremamente complesso e non è possibile descriverlo dettagliatamente in queste pagine. Il virus scandisce la tabella interna delle funzioni, la aggiorna in modo da aggiungere una funzione che fa da "loader" per il virus, mentre il codice del virus è inserito in coda al file. Quindi, l'infezione si propaga attraverso i file di documenti con macro, ma essa è attivata automaticamente tramite i file VxD, dei quali alcuni vengono eseguiti immediatamente all'accensione del computer, come una parte del sistema operativo.

3.1.5 - Macro virus

Il tipo di virus qui presentato è relativamente recente e presenta caratteristiche tecniche totalmente differenti da quelle viste finora. Un linguaggio di macro è un linguaggio, dalla sintassi normalmente molto semplice, presente in alcuni tipi di programma per elaborazione di testi o per la gestione di fogli elettronici e database. Viene utilizzato per creare delle brevi procedure, per gestire o rappresentare i dati in maniera particolare. I primi linguaggi di questo tipo erano molto semplici e non avevano le caratteristiche necessarie per essere utilizzati per scrivere virus. In seguito i linguaggi si sono evoluti fino ad avere una capacità di interagire con il sistema operativo e con le memorie di massa molto avanzata, la capacità di alterare i file presenti nell'ambiente e in particolare le macro presenti in altri documenti.

Una definizione di macro virus, fornita dal Bontchev, è la seguente: "Un macro virus è un insieme di una o più macro, il quale è capace di replicarsi ricorsivamente".[46] Un primo monito riguardo alla possibilità di una minaccia da parte di questi tipi di virus provenne da J. Highland [10], il quale descrisse degli esperimenti eseguiti con il linguaggio delle macro del Lotus 123, riuscendo a produrre una sorta di virus.
Quando questo problema si è presentato, la comunità informatica era impreparata e i primi macro-virus, come WM.Concept, hanno avuto una diffusione grandissima.

Il tipo di macro-virus che viene descritto in seguito, è in grado di infettare solo i documenti creati con i programmi della suite Microsoft Office, in particolare Word, un elaboratore di testi. Esistono comunque virus scritti nel linguaggio di macro di altri programmi, ma sono una minoranza. La grande diffusione della suite Office rende i virus scritti nel linguaggio di macro dei suoi principali elementi (Word, Excel, Access)

praticamente un problema a se. Si inizia con l'esaminare il programma Word, che avendo caratteristiche comuni a tutti gli altri rende possibile una descrizione più ampia.

In un documento Word si possono utilizzare diversi tipi di macro. Ognuna di esse ha un nome che viene assegnato dall'utente. Il linguaggio utilizzato per le macro è stato fino alla versione 7.0 di Word il WordBasic, un linguaggio molto simile al Visual Basic[14]. Poi, con l'Office 97 è stato introdotto il VBA, o Visual Basic for Applications, una evoluzione di questo linguaggio, però compatibile con macro scritte in Word Basic attraverso un meccanismo di conversione automatico. Queste macro possono essere eseguite in differenti modi: vi sono le cosiddette "auto macro", delle macro con un nome particolare, che vengono eseguite automaticamente assieme ad alcune normali operazioni del programma. Vi sono macro che sono rese disponibili in una libreria, infine esse possono essere associate ad un evento come la pressione di un bottone o di un elemento dei menu. Il file NORMAL.DOT contiene un elenco di macro rese disponibili a tutti i documenti che vengono aperti. Di seguito l'elenco di alcune macro automatiche.:

AutoOpen: viene eseguita ogni volta che il documento che la contiene viene aperto. Se si trova in normal.dot, viene eseguita ad ogni apertura di documento.

AutoClose: viene eseguita ogni volta che il documento che la contiene viene chiuso. Analogo ad AutoOpen se si trova in normal.dot.

AutoExec: viene eseguita ogni volta che il programma (Word, Excel, ecc.) viene aperto. (Per essere eseguita la macro deve trovarsi necessariamente in normal.dot).

AutoExit: viene eseguita ogni volta che il programma viene chiuso. (Per essere eseguita deve necessariamente trovarsi in un documento aperto al momento della chiusura del programma o in normal.dot.)

I primi virus utilizzavano le macro automatiche per essere eseguiti immediatamente e senza che l'utente se ne potesse accorgere. La prima operazione che esegue una macro virale, sarà di convertire i documenti infettati nel formato Template, il quale può contenere

[14] Quindi molto più facile da utilizzare dell'assembler o del C,linguaggi in cui sono normalmente scritti i virus. Questo ha reso la programmazione di virus accessibile a molte più persone; questo è uno dei motivi della grandissima diffusione dei macro virus.

macro. In seguito, anche per una forma di protezione dagli anti-virus[30], i quali eseguivano una scansione principalmente sulle macro automatiche, sono state intraprese altre strade. La prima strada alternativa a questa è quella delle system macro, che sono macro che eseguono delle operazioni di base del programma, come "FileSaveAs", per esempio. Il dottor Bontchev, nella sua pubblicazione[15], ne cita 122 e le elenca tutte in una tabella. Ognuna di esse può essere riassegnata da una macro scritta dall'utente; un virus può quindi essere inserito in una macro così ridefinita e poi richiamare le operazioni desiderate dall'utente. In questo modo ogni qualvolta un utente esegue una operazione che è stata così intercettata, in realtà avvia il virus associato. Un altro sistema è di inserire delle macro che intercettino comandi dei menu del programma: ogni voce di menu può essere eliminata e sostituita con un altra voce con lo stesso nome ma che esegue una macro. In questo modo un virus può essere attivato da una semplice selezione di voce di menu e, in seguito, usando le macro di sistema, eseguire anche l'operazione richiesta dall'utente. Una operazione analoga può essere eseguita ridefinendo le icone nelle barre di comando del programma. Ancora un altro sistema, che sfrutta un meccanismo analogo a quello degli ultimi due, è quello di associare delle macro virali ai cosiddetti "key shortcuts" o scorciatoie dei tasti. Si tratta di un meccanismo per cui si può associare l'esecuzione di una macro alla pressione di un tasto o una combinazione di tasti, normalmente una lettera e una combinazione dei tasti Control, Shift, Alt.

Un meccanismo diverso da questi, utilizzato per aumentare l'infettività dei macro virus, è quello di infettare il file NORMAL.DOT, che contiene macro disponibili in modo globale, all'avvio del programma, in ogni documento che viene aperto. I virus, comunemente, infettano questo file, così alla successiva esecuzione di Word non sarà necessario leggere un documento infetto per attivare il virus. Esiste anche un altro sistema per avere lo stesso risultato: esiste una particolare directory, detta di STARTUP, che viene esaminata da Word ad ogni avvio; in questa directory vi possono essere dei particolari file, DOT o WLL, che contengono macro o istruzioni da eseguirsi per prime. Un virus può inserire in questa directory un file contenente una copia di se stesso, ottenendo un risultato identico a quello ottenuto infettando NORMAL.DOT. Esistono altri meccanismi più particolari, che saranno esaminati nel paragrafo 3.3, utilizzati dai macro virus per contrastare l'effetto degli anti-virus.

[15] Vedi nota 6.

Sono stati riportati dettagli tecnici relativi ai macro virus relativi al programma Word, ma vi sono macro virus in grado di infettare Excel e Access, come già accennato. Il linguaggio di macro è lo stesso del Word, il VBA, e i meccanismi sono in gran parte gli stessi, eccetto alcuni dettagli, relativi per lo più ai file di startup e ai file globali, come NORMAL.DOT del Word, per intenderci. Esistono virus in grado di infettare indifferentemente Word ed Excel, ed addirittura tutti e tre i programmi citati, come il virus TriState; abbiamo inoltre già visto un virus capace di infettare documenti di Word e file eseguibili contenenti device driver virtuali di Windows, il Navrhar. Esso infetta i documenti Word, inserendo una macro automatica, AutoOpen, la quale non è che il "loader" del virus, che viene inserito come overlay nel documento infetto, alla fine dei dati. Notare che i documenti della suite Office hanno un formato interno molto particolare, detto OLE2, che considera il file come una unità a se, includendo una tabella di allocazione e una tabella delle entry. Il virus Navrhar scandisce e modifica queste tabelle, come già modificava le tabelle interne dei file VxD. Si tratta, quindi, di un virus che comprende in se tecniche molto avanzate, per la gestione di file con formato interno molto complesso.

3.1.6 - Worm

I worm sono virus di tipo particolare, essi si riproducono non tramite i file o tramite i dischi, ma tramite i canali di comunicazione diretta, in parole povere le reti geografiche. Il primo caso rilevato è quello del worm programmato da Robert Morris, di cui si è trattato nel primo paragrafo. Una analisi tecnica del suo programma rivela che esso è stato reso possibile da alcuni errori e dimenticanze nella gestione della sicurezza sul sistema operativo Unix BSD versione 4. Possiamo inoltre rilevare, dalla sua descrizione tecnica,[16] quali sono le caratteristiche principali di un worm attivo su Internet e quali sono i requisiti che deve avere un sistema per renderlo funzionale.

- Il worm deve poter accedere ad una rete di computer. Infatti, come già accennato, esso non si replica eseguendo copie di se stesso su altri file, che *in seguito*, possono venire trasportati su altre macchine e quindi propagare l'infezione. Il worm si trova su una certa macchina e si riproduce solo su un'altra macchina con cui è entrato in contatto diretto.

[16] Vedi nota 6.

- Il worm deve poter conoscere uno o più nomi di utenti della macchina bersaglio, oppure deve essere in grado di insinuarsi in un programma o documento che sarà prelevato da un qualche utente di un'altra macchina. Esso penetra nel computer bersaglio sotto forma di file eseguibile o di macro, ovvero di allegato ad un messaggio di posta elettronica o attraverso una pagina web, ecc. Può avere bisogno di un destinatario o di un login, eventualmente di cui scoprire la password, per aprire una sessione telnet o simili.
- Devono essere presenti degli errori nella gestione della sicurezza del sistema operativo, tali da permettere al worm di eseguire delle operazioni non previste e non volute.

Il worm scritto dal Morrison utilizzava gli applicativi *fingerd* e *sendmail*, due parti del sistema operativo, con cui riusciva prima a penetrare nella memoria della macchina bersaglio, inviando un vettore, e poi a prenderne il controllo. In questo modo utilizzava la rete Internet creando dei *socket* per la trasmissione dei comandi che erano eseguiti da una shell, attivata sulla macchina bersaglio grazie ad un errore nei programmi indicati. In seguito veniva attivato un decifratore di password per cercare di prendere completamente possesso della macchina. Infine il procedimento si ripeteva.

Il worm che abbiamo descritto era totalmente automatico, perché gli errori nella sicurezza rilevati erano tali da permettere al worm di prendere autonomamente il controllo dell'area di un altro utente. I worm odierni non utilizzano questo sistema automatico, ma sfruttano l'ingenuità o la buona fede degli utenti, con degli stratagemmi. Il motivo principale è che la gestione della sicurezza nelle reti è ora molto curata, proprio per il moltiplicarsi di individui interessati per passione o altro a sfidarne la robustezza.

I worm odierni sono principalmente di tre tipi:

- Via posta elettronica.
- Via HTML
- Attraverso programmi funzionanti in rete e non sufficientemente protetti (es. programmi per accedere ad IRC od ICQ)

I worm di posta elettronica sono i più diffusi e rappresentano una reale minaccia. Una delle tecniche è la seguente: viene inviato un messaggio di posta elettronica, con un allegato contenente un file eseguibile o un documento, con qualche frase capace di stimolare

la curiosità del ricevente ed indurlo ad aprire l'allegato. Il file allegato, un eseguibile o un documento contenente una macro, è il vettore del virus. Esso, appena entrato in esecuzione, normalmente cerca nel sistema un file che contenga l'agenda dell'utente, poi prende il controllo del programma che gestisce la posta ed invia una copia di se stesso a tutti gli indirizzi trovati nell'agenda. Infine può contenere dei payload che si attivino subito oppure a distanza di tempo, dopo che il worm ha preso il controllo del sistema, normalmente modificando od alterando alcuni file del sistema operativo.

Un'alternativa a questa tecnica è di modificare il programma di posta elettronica, facendo in modo di aggiungere il virus in allegato ad ogni messaggio che viene inviato, avendo trovato un soggetto in grado di stimolare la curiosità del ricevente, che viene indotto ad aprire il programma allegato.

Esempi di questi worm sono VBS.Freelink, che contiene un allegato scritto usando il VisualBasicScript, un linguaggio simile al Visual Basic, installato sul sistema operativo Windows 98 e successivi, per fornire a tutto il sistema operativo un meccanismo simile a quello delle macro di Office. Il soggetto del messaggio che lo contiene è "Have fun with these links, bye", che significa "divertiti con questi link, ciao". Infatti, il worm, dopo essere stato eseguito presenta una finestra di dialogo che promette di creare un link ad un sito per adulti e, se l'utente seleziona il tasto yes della finestra di dialogo, lo crea realmente. Inoltre il ricevente vedrà che il mittente del messaggio è una persona conosciuta, poiché il suo nome era presente nella sua rubrica, e questo lo tranquillizza e lo invoglia a verificarne il contenuto. Il worm prende il controllo di Outlook, programma per la gestione della posta elettronica, ed invia una copia di se stesso agli indirizzi trovati nel file della rubrica. Il worm prende il controllo della macchina alterando alcuni file del sistema operativo e i registri di Windows, in modo da entrare in esecuzione ad ogni avvio; inoltre modifica alcuni file dei programmi mIRC e PIRCH32, programmi che vengono utilizzati per mettere in contatto la Internet Relay Chat, ovvero un insieme di server sparsi un po' dovunque ed utilizzati come chat-line. Quando i programmi sono stati infettati, al momento dell'esecuzione, appena entrati in una sessione, inviano a tutti gli utenti collegati un messaggio contenente il worm e con soggetto uguale a quello della mail.

Un altro worm diffuso è VBS.LoveLetter, che ha ottenuto un grande rilievo da parte dei mass-media. La grande diffusione del worm VBS.LoveLetter è dovuta all'astuzia

dell'autore, infatti, il soggetto del messaggio, "Love Letter For You" (Lettera d'amore per te), è molto accattivante. Il worm prende il controllo della macchina in modo analogo al worm VBS.Freelink, alterando alcuni file del sistema operativo e i registri di Windows, in modo da entrare in esecuzione ad ogni avvio; inoltre intercetta alcune funzioni del programma mIRC, modificando il file SCRIPT.INI di mIRC e, appena il programma viene eseguito, si auto-invia come file contenente un messaggio per il ricevente, in formato HTML, con titolo uguale a quello delle mail. Quando si è attivato, il worm contatta un sito nelle Filippine da cui scarica un programma, dal nome "Win-Bugsfix", che in realtà cerca alcuni file contenenti password e li invia al sito di provenienza; anch'esso viene inserito in un registro di Windows, per essere attivato ad ogni avvio. Il worm sovrascrive tutti i file VBS e VBE con il suo codice; inoltre crea un nuovo file con estensione VBS, contenente il suo codice, per ogni file con estensione JS, JSE, CSS, WSH, SCT, HTA, JPG, JPEG, dopo cancella tutti i file originali. Nasconde inoltre tutti i file con estensione MP2 e MP3, creando per ognuno un nuovo file con lo stesso nome ed estensioneVBS, contenente il suo codice.

Vi sono poi worm analoghi a questi, che però utilizzano un allegato scritto in codice eseguibile, piuttosto che in VBS, come Happy99. Comunque, i worm descritti, più che utilizzare degli errori nella gestione della sicurezza, sfruttano l'ingenuità degli utenti. Il worm VBS.Bubbleboy, invece, sfrutta un errore nella gestione della sicurezza del programma Outlook associato ad Internet Explorer 5. Questo worm è apparso nel dicembre 99 "in the Wild"[17], e ancora presente nella WildList di giugno 2000. Il worm Bubbleboy arriva sotto forma di messaggio di posta elettronica. Grazie all'errore presente in Outlook, uno script VBS inserito nel corpo della mail, ma invisibile, viene attivato automaticamente e inserisce un file, di nome UPDATE.HTA, nella directory di avvio automatico, nel menu principale di Windows. Questo diventerà attivo al successivo avviamento del computer e avvierà un programma, in codice VBS, che verifica l'esistenza di una voce nel registro di Windows, per evitare di ripetere l'infezione più di una volta; se la voce non esiste viene creata, altrimenti il worm termina. Poi il worm invia una copia di se stesso, usando il programma Outlook, a tutti gli indirizzi contenuti nell'agenda. Infine, visualizza un messaggio camuffato da messaggio di sistema, che segnala un errore di sistema e richiede la cancellazione del file UPDATE.HTA. La Microsoft ha prontamente preparato un aggiornamento scaricabile da Internet[18], per correggere gli errori presenti nel programma

[17] Nella lista dei virus attivi "WildList", vedere paragrafo 5.4.
[18] http://microsoft.com/security/Bulletins/ms99-032.asp

Outlook, comunque il worm non risulta molto diffuso. Con lo stesso meccanismo funziona il worm VBS.Kak.Worm, apparso intorno al 15 agosto 2000.

Un ultimo tipo di worm, ancora non esistente ma di cui si teme la comparsa, è un worm che sfrutti le pagine WEB per diffondersi. Questo timore nasce con la comparsa di alcuni linguaggi per inserire delle brevi procedure, quasi delle macro, all'interno della pagina WEB, come il Javascript, le applet Java e l'Activex. Il Java è un linguaggio con sintassi simile al C, basato sugli oggetti, dotato di grande portabilità. Esso è compilato in un formato particolare, chiamato bytecode, che viene interpretato da una macchina virtuale, la Java Virtual Machine. Questa è presente nei principali browser per pagine WEB, come Netscape o Internet Explorer; è quindi possibile creare una routine in Java, detta applet, con caratteristiche particolari che la rendono adatta ad operare in una pagina WEB. La Java Virtual Machine presente nei browser ha una elevata sicurezza, infatti, non permette alcun contatto con il sistema attorno ad esso; però si teme che esistano delle falle nel sistema di sicurezza e che queste possano essere scoperte ed utilizzate. Un tale worm potrebbe alterare le pagine web di un sistema che lo riceva e, quindi, infettare siti WEB, inoltre potrebbe attivare altri tipi di riproduzione in loco; infine, contenere payload dannosi. Al momento, non si rileva nulla del genere, vi sono però dei cosiddetti *Java exploit*, che sono dei meccanismi particolari per effettuare operazioni illecite, usando alcune vulnerabilità dei sistemi. Uno di questi problemi, descritto purtroppo da un sito di cui non si può garantire la credibilità, parla di una vulnerabilità nel meccanismo delle aste virtuali effettuate da una certa compagnia. Attraverso questa vulnerabilità sarebbe possibile inserire, nella descrizione dell'oggetto da vendere all'asta, un codice Java in grado di sottrarre il nome e la password di accesso usato dagli utenti per accedere alle aste; con questi dati terze persone potrebbero fare delle offerte nelle aste a nome dell'utente piratato. Il nome di questo exploit è Ebayla.

I controlli Activex sono una sorta di linguaggio di macro per pagine WEB realizzato dalla Microsoft, attivo su molti programmi come Internet Explorer, Outlook, ecc. I problemi posti da questo linguaggio sono gli stessi delle applet Java, con una differenza: i controlli Activex non sono delimitati ai programmi che li ricevono ma possono interagire con il sistema. Per risolvere questo problema la Microsoft ha dotato i controlli Activex di una firma digitale, che presenta all'utente, al momento dell'attivazione, il nome dell'autore e il nome della autorità di certificazione che ha rilasciato la firma all'autore. Prima di eseguire un controllo Activex ricevuto via Internet, il browser presenta una finestra di dialogo che

visualizza le informazioni rilevate sulla firma del controllo, se un controllo scaricato non ha una firma di un autore ritenuto degno di fiducia, oppure non possiede una firma, l'utente può negare il permesso di essere eseguito al controllo. Il worm di cui si è parlato più sopra, VBS.Bubbleboy, utilizza proprio un errore nella gestione della sicurezza dei controlli Activex di sistema[19], infatti, la macchina virtuale Activex considera come "sempre sicuro" un controllo in grado di interagire con il sistema esterno. Quindi, se tale controllo viene richiamato, esso viene eseguito senza richiedere l'autorizzazione all'utente. Come si è già evidenziato, la Microsoft ha subito preparato un upgrade per correggere questo problema. Naturalmente esiste la possibilità di non esserne a conoscenza del problema, oppure ancora di dovere ri-installare il sistema operativo e di scordarsi di ri-installare anche l'upgrade correttivo; infatti, il worm Bubbleboy risulta, stando alle informazioni della WildList, ancora diffuso.

3.1.7 - Altri tipi di virus

Vi sono alcuni tipi di virus, che si possono riunire in un paragrafo generale. Nei paragrafi precedenti sono stati descritti dei meccanismi virali applicabili a diversi tipi di sistema operativo, ma specificamente rilevati su personal computer con sistema operativo Microsoft. La grande diffusione di questo sistema permette di considerare i relativi virus come una categoria a se, anche se in realtà esistono virus che si riproducono sul sistema operativo Linux, per esempio, con una tecnica simile agli infettori dei file eseguibili e dei file script, di cui si tratta nel seguito del paragrafo.

Anche il sistema Macintosh è terreno fertile per virus, in proporzione molto ridotta. Presso il sito del CIAC[20], si trovano alcuni bollettini tra cui il Virus Information Update, contenente diverse tabelle complete sui più diffusi malware riguardanti Personal Computer, Macintosh e altri microcomputer. La Macintosh Computer Virus Table aggiornata al maggio 1998 conteneva un centinaio di voci, tra cui per esempio il virus nVIR, che infetta il sistema operativo e ogni file eseguibile che viene aperto, con una tecnica simile a quella

[19] Si tratta di controlli già presenti nella libreria del programma interessato, che non debbono essere scaricati ma solo attivati indicandone il nome.
[20] Computer Incident Advisory Capability, (informatore sui possibili incidenti dei computer) è realizzato dal Computer Security Technology Center dei Lawrence Livermore National Laboratory, U. S. Department of Energy,

degli infettori dei virus COM del DOS. Esiste inoltre, sempre nello stesso documento, una tabella dei diversi tipi di virus esistenti in quel periodo, con una descrizione dettagliata dei tipi di virus relativi al Macintosh, di cui si riporta di seguito la traduzione:[63]

- Bogus CODE resource.

  Il virus è aggiunto come un nuovo segmento CODE e la tabella dei salti viene alterata per puntare al nuovo segmento. Per esempio, quando una applicazione viene infettata con il virus nVIR, esso aggiunge una risorsa CODE 256 alla fine dell'applicazione e modifica la risorsa CODE 0 (la tabella dei salti) per eseguire la risorsa CODE 256 prima di eseguire l'applicazione. Molti virus per il Macintosh sono di questo genere, per esempio Scores, nVIR, INT29.
- Patched CODE resource.

  Il codice del virus è aggiunto al termine del segmento di codice principale; la prima istruzione del programma, oppure la tabella dei salti, vengono corrette per puntare al codice del virus.
- Bogus INIT.

  Una chiamata di sistema INIT è eseguita al momento del boot prima che il controllo passi al sistema operativo. È possibile modificare il sistema e cambiare questa funzione, e questa opportunità può essere sfruttata da un virus.
- Bogus resource.

  Questi tipi di virus installano una versione modificata delle risorse standard di sistema nella catena delle chiamate tra un programma ed il sistema. Quando un programma necessita di una risorsa, prima esamina l'ultimo file aperto e poi prosegue fino al primo file aperto (il sistema operativo) finchè trova la risorsa di cui ha bisogno. L'ultimo file aperto è normalmente un documento, seguito da applicazioni, dal file di desktop, dal finder e dal sistema. Una risorsa virale posta in qualsivoglia di questi file, verrà utilizzata al suo posto nel sistema.

Un altra fonte interessante di materiale è il testo Viruses and the Mac FAQ [75], disponibile su usenet nel newsgroup comp.virus, un newsgroup moderato da un noto ricercatore del campo anti-virus, in cui sono disponibili solo documenti di un certo spessore.

Il seguente specchietto presenta alcuni particolari tipi di malware, che non sono stati descritti nei paragrafi precedenti.

- Trojans o cavalli di Troia. Sono programmi apparentemente innocui, come giochi, compressori di file, utilità di gestione del sistema. In realtà essi contengono al loro interno delle funzionalità non previste e non volute. Possono contenere dei payload distruttivi, come formattazione di Hard Disk, ovvero delle bombe logiche, in grado di effettuare danni dopo un certo periodo.

- Backdoor. Si tratta di programmi spesso contenuti in altri programmi vettori, secondo il modello dei cavalli di Troia, con una funzionalità particolare. Essi entrano in contatto con altri utenti, quando il computer è connesso alla rete Internet, e danno ad essi la possibilità di effettuare operazioni illecite sui computer in cui si sono installati, come trasmettere informazioni riservate, quali password, contenuti di documenti, directory, oppure modificare file, generare finestre di dialogo, ecc. I più comuni sono Back Orifice e Netbus. Back Orifice, per esempio, si può scaricare da alcuni siti di hacker o di collezionisti di virus; viene distribuito come un pacchetto contenente server, il programma per prendere il controllo del computer infettato, e del client, che può essere nascosto in un altro programma.

- Virus sorgente. Si tratta di virus che cercano un particolare tipo di file sorgente, scritto in un determinato linguaggio. Essi vi inseriscono un insieme di linee che contengono il sorgente del virus, e modificano il sorgente infettato in modo da richiamarne la funzione virale aggiunta. Quando il programma viene compilato ed eseguito, il virus si attiva.

- Batch virus. Si tratta di virus o trojans scritti usando il linguaggio di gestione delle shell, sia di Unix che di DOS. Un esempio conosciutissimo consiste in un file script contenente la seguente linea: "echo + +>~/.rhosts"; esso crea un file, nella home directory di un area Unix, che imposta un'opzione per cui, su certi sistemi, chiunque può accedere all'area senza conoscere la password.

- Hoax. Vengono inclusi nell'elenco dei virus, anche se in realtà non si tratta di programmi, ma di falsi allarmi. Alcune volte essi vengono diffusi attraverso una catena di mail, contenenti la descrizione di un improbabile virus, normalmente definito

pericolosissimo; altre volte queste notizie vengono diffuse dai media. Il danno prodotto da queste false notizie è spesso di tipo psicologico: immotivate apprensioni o applicazioni di misure preventive ; questo, però, in ambito corporativo si può tradurre in una perdita reale. Il "motore di replicazione" di questo genere di virus è proprio la persona umana, con la sua curiosità e il suo desiderio di comunicare le notizie.

Un caso interessante accaduto di recente è il caso dei file della ditta Radiate/Aureate, che si occupa di pubblicità su Internet. Alcuni software gratuiti, distribuiti attraverso la rete, presentano una finestra che visualizza immagini pubblicitarie di siti Internet. La ditta Radiate si occupa di fornire un server per visualizzare queste immagini o banner, che viene interrogato da alcuni file di libreria dinamica, distribuiti assieme ai programmi gratuiti, (il più comune è advert.dll) che si attivano al momento di eseguire i programmi; essi prelevano da un sito le immagini da visualizzare nei banner e segnalano al server quali immagini sono state selezionate dall'utente. Ad un certo punto sono stati pubblicati articoli on-line che denunciavano la presenza, nei file della famiglia di advert.dll, oltre che dei dispositivi atti a trasmettere le immagini pubblicitarie e a rilevarne la selezione da parte dell'utente, anche di routine in grado di trasmettere dati personali dell'utente via Internet al server della Radiate. Il seguente articolo è stato pubblicato da una fonte di un certo rilievo, quale il sito de "La Repubblica":

«Fate un piccolo esperimento: cercate sul disco rigido del vostro computer un file denominato "advert.dll". Probabilmente ne ignoravate l'esistenza, né sapete come sia arrivato lì. E soprattutto a cosa serva. Ecco le risposte: quel file si piazza sull'hard disk quando installate alcuni dei più popolari software gratuitamente scaricabili dalla Rete (la lista ne contiene centinaia). E ogni volta che vi connettete a Internet invia alla società che distribuisce i programmi - a vostra insaputa - una lunga serie di informazioni su vostro conto.»

Tratto integralmente da "La Repubblica.it, tecnologie ed Internet", 20 luglio 2000, autore "G. Mol."
http://www.repubblica.it/online/tecnologie_internet/track/advert/advert.html

In effetti, questi programmi entrano in funzione non appena viene attivato il browser per la navigazione in Internet, piuttosto che all'attivazione dei programmi distribuiti gratuitamente di cui si è parlato. La Radiate, dal canto suo, ha pubblicato sul suo sito alcune pagine informative che negano questo addebito; la Radiate afferma che nella documentazione allegata ai software gratuiti è presente la descrizione di questo software e la sua attività. Inoltre, ha reso disponibile un programma in grado di eliminare i file in questione, segno che molti utenti debbono aver contattato la Radiate con messaggi di richiesta informazioni e protesta. Sono anche comparsi dei programmi che si presentano

come “soluzione al software-spia”, del cui corretto funzionamento si può fortemente dubitare. Dal punto di vista legale, in Italia, un comportamento come quello addebitato alla Radiate sarebbe vietato, ammettendo che le voci fossero vere. A questo punto, salvo che qualcuno presenti una denuncia a qualche organo giudiziario e questo non disponga una perizia sui file oggetti della disputa, per mezzo di un reverse-engineering adeguato, la questione è destinata a cadere nell’oblio, ovvero ad essere annoverata nell’elenco dei falsi allarme Internet (hoax). Il problema, infatti, è stato rilevato e registrato come un hoax già dal sito della compagnia anti-virus NAI, alla pagina http://vil.nai.com/vil/ve98516.asp.

3.2 - Mezzi di infezione e di trasmissione

Si prenderanno ora in esame i vari mezzi di infezione e la loro evoluzione, parallelamente all'evoluzione tecnica dei virus.

Il virus del settore di boot necessita, per l'attivazione, che un disco infetto sia in una unità collegata al sistema nel momento del bootstrap e l'unita venga interrogata dal sistema per il boot. Non è necessario che il disco infetto contenga anche il sistema operativo; in fase di interrogazione, il sistema esegue il settore di avvio di qualunque disco si trovi nell'unità, per poi eventualmente segnalare che i file del sistema operativo non sono stati trovati; questo è sufficiente affinché l'infezione si attivi. Questo significa che, se non è stato infettato un disco di avvio, la possibilità di riproduzione è abbastanza bassa, però non nulla; è necessario un errore dell'utente. Nei primi sistemi, privi di disco fisso, questo poteva accadere abbastanza frequentemente, poi con la presenza dei dischi fissi, che normalmente sono resi di avvio, è necessaria una dimenticanza di un disco infetto nell'unità al momento dell'avvio. Si noti però che, a differenza dei sistemi privi di disco fisso, in cui era quindi necessario inserire il disco di avvio ad ogni accensione, nei sistemi a disco fisso non è più necessario effettuare questa operazione. Quindi è più facile scordare un disco nell'unità in fase di accensione. In ogni caso, l'infezione non ha probabilità certa di manifestarsi, anche ricevendo un disco infetto che sia utilizzato normalmente.

I virus dei file eseguibili, invece, diventano attivi dopo la semplice esecuzione del file. È sufficiente utilizzare un programma contenuto in un file infetto, per diffondere subito

l'infezione nel proprio sistema. Quindi in caso di utilizzo di materiale infetto, la probabilità di contrarre l'infezione, salvo errori di programmazione del virus o incompatibilità del sistema operativo, è del 100%. Inoltre il mezzo di diffusione non è più "fisico" (un dischetto) ma "logico" (un file); quindi l'infezione si può contrarre attraverso qualunque canale di trasmissione dati, tipicamente BBS ed Internet.

Come si è visto nei paragrafi precedenti, i virus di macro hanno avuto inizialmente una diffusione imprevista. Questo è dovuto al fatto che non si prevedevano virus capaci di utilizzare i documenti di testo come canali di infezione. Anzi, spesso sono apparsi degli *hoax*, cioè dei falsi allarmi diffusi da mitomani o pseudo-spiritosi, che paventavano una minaccia virale per mezzo di documenti di testo. È quindi di vitale importanza, per la riproduzione dei virus, rendere occulto ed imprevisto il più possibile il canale di trasmissione scelto. In effetti, la massima diffusione viene ottenuta da virus che utilizzano canali e tecniche rivoluzionarie, rispetto ai propri predecessori, in quanto essi possono riprodursi molto rapidamente e massicciamente nel periodo immediatamente successivo alla loro iniziale distribuzione, perché gli anti-virus ancora non sono in grado di rilevarli e gli utenti di computer ancora non conoscono la nuova minaccia.

Il vero agente di infezione, trascurando i rari casi dei worm, è quindi l'uomo. È risultato perciò necessario educare e informare gli operatori e gli utilizzatori su una "profilassi" informatica, dando delle norme precauzionali per evitare danni prodotti da infezioni virali. Si riporta di seguito il riassunto delle norme prescritte, per la prevenzione aziendale ed industriale delle infezioni virali, da un noto ricercatore sui virus come H. J. Highland: [28]

- Usare solo programmi reperiti da un fornitore affidabile.
- Non usare mai programmi prelevati da una BBS o da Internet.
- Stabilire delle procedure che proibiscano ai dipendenti di trasportare programmi da casa al posto di lavoro.
- Usare hard disk rimovibili per ridurre l'infezione.

  La giustificazione di questa norma si intuisce poiché, così facendo, non è più necessario utilizzare un gran numero di dischetti, quando si debba trasferire software

o dati all'interno dell'azienda. Attualmente esistono soluzioni più efficienti, come i CD-ROM.

- Usare stazioni prive di dischi in una LAN.
- Verificare ogni nuovo software.
- Creare copie di backup con cadenza regolare.
- Riporre e conservare i dischi originali in un luogo sicuro.
- Eseguire regolari check-up delle reti locali.
- Non permettere agli utenti di una LAN di accedere a BBS esterne o ad Internet.

  Questa norma è certamente efficace ma poco attuabile nel periodo odierno, eccetto casi in cui sia richiesta una grande sicurezza.
- Ridurre i rischi di infezione stabilendo procedure operative sicure.
- In caso di infezione isolare i sistemi infettati o sospetti.
- In caso di infezione rivolgersi ad assistenza specializzata.

  L'autore sconsiglia le soluzioni empiriche da parte di dipendenti poco esperti, che possono, ad esempio, cancellare un file considerato sospetto e proseguire il lavoro senza un adeguato check-up dei dischi rigidi.

Le indicazioni fornite dagli studiosi del fenomeno si orientano su una gestione della protezione anti-virus corporativa organizzata secondo un modello gerarchico lineare o piramidale, in cui vi sia un solo responsabile della gestione degli anti-virus, inteso come una sola persona o un solo team. Il responsabile deve analizzare gli eventi virali, riconoscere eventuali epidemie in corso, effettuare statistiche ed, in base ai dati analizzati, intraprendere adeguate contromisure. Quindi le corporazioni richiederebbero la possibilità di delegare la configurazione dei software anti-virus ad un solo server, posto in una LAN, una rete locale oppure una rete geografica, di qualunque genere siano gli elementi di essa [51], [55].

3.3 - Meccanismi per eludere gli anti-virus

Da subito, molti sviluppatori hanno iniziato a produrre software per cercare ed eliminare i virus, di questi software daremo una trattazione ampia nel capitolo successivo. In questo paragrafo si vuole descrivere le tecniche sviluppate dagli autori di virus per cercare di rendere i propri virus non rilevabili o più difficilmente rilevabili dai programmi anti-

virus. Come si può rilevare dal paragrafo precedente e da alcuni studi, di cui uno dal titolo, molto significativo, "Computer virus - Anti-virus coevolution", di C. Nachemberg [6], una delle principali caratteristiche di un virus "valido" è di agire all'insaputa dell'utilizzatore del computer, fino al momento del'attivazione del payload, se presente. Un virus capace di eludere i programmi anti-virus avrà una diffusione ed efficacia maggiore. La pubblicazione di Nachemberg esprime che vi è stata una evoluzione parallela degli anti-virus e dei virus, già dall'inizio; una rincorsa per migliorare le tecniche di rilevamento, riguardo agli anti-virus, e, parallelamente, per eludere le nuove tecniche di rilevamento.

In base alle tecniche utilizzate per evitare il rilevamento, nascono nuove categorie di virus che si possono riassumere nelle seguenti:[21]

- virus polimorfi
- virus stealth
- retrovirus

3.3.1 - Virus polimorfi

La prima categoria è anche la più importante e diffusa, ha avuto una interessante evoluzione: consiste nell'introdurre all'interno del codice del virus una o più routine auto-modificanti. Nel successivo capitolo si vedrà come gli anti-virus usino, tra l'altro, delle tecniche a scansione: ogni file viene aperto e letto alla ricerca di virus conosciuti o di meccanismi che lo rendano sospetto, per esempio una ricerca di file COM ed EXE. Inoltre, viene letto il settore di boot dei flloppy disk e il MBR delle unità a disco fisso, anche in questo caso per cercare virus sconosciuti o meccanismi sospetti. Per effettuare il primo tipo di ricerca, i programmi anti-virus eseguono la cosiddetta scansione di *signature* (vedi 4.1.1) in cui si ricerca una sequenza di dati caratteristica per ogni differente virus conosciuto. Se il virus è in grado di modificare se stesso, questa ricerca sarà molto più difficile, richiedendo tecniche aggiuntive particolari.

[21] Si tratta di una analisi rivolta, nuovamente, ai virus diffusi sui sistemi MS-DOS e Windows; le tecniche descritte comunque si possono applicare ad altri sistemi.

Il modo più semplice ed efficace con cui viene realizzato il polimorfismo è la crittografia. Poniamo di avere un brevissimo virus in tabella 1 e, a fianco, la sua codifica in codice eseguibile.

Inizio	00
Istruzione 1	A0
Istruzione 2	35
Istruzione 3	37
Istruzione 4	19
Istruzione 5	E3
.....	...
Istruzione n	...
Fine	FF

Tabella 1

Label	Codice Sorgente	Codice Operativo
Top:	Inizio	00
Routine:	Routine crittografica	...
TopVir:	Istruzione 1	A0
	Istruzione 2	35
	Istruzione 3	37
	Istruzione 4	19
	Istruzione 5	E3
	.....	...
BottomVir:	Istruzione n	...
	Fine	FF

Tabella 2

Questo significa che un file infettato da questo virus contiene in qualche punto del suo corpo la stringa di valori S = {A0, 35, 37, 19, E3}. Il caso esaminato è puramente esemplificativo: infatti una stringa di cinque valori è troppo breve per essere significativa; le signature utilizzate hanno una dimensione minima di 15 - 20 byte. Un programma anti-virus cerca quindi in tutti i file da esaminare la stringa S.

Consideriamo ora il virus in tabella 2. Sia la "routine di crittografia" una routine che esegue, per tutto il corpo del virus, lo XOR dei valori con un numero determinato:

```
Routine di Crittografia
Integer I , Key:=FF;
        for I := indirizzo (TopVir) to indirizzo (BottomVir)
                mem[I]:=xor( Mem[I], Key);
End Routine di Crittografia
```

È sufficiente eseguire questa routine in fase di infezione, quindi di replica del virus, aggiungendo in testa ad esso una routine simile, in grado di decifrare il virus subito prima che venga eseguito. Il valore con cui viene eseguito lo XOR, ovvero la chiave di codifica, è 255 (in esadecimale FF) per la prima volta; la routine di duplicazione contenuta nel virus dovrà provvedere a cambiare la chiave ad ogni duplicazione, normalmente con dei valori random. Si noti come risulta semplice modificare la routine mostrata, in modo da utilizzare due o più bytes di seguito come chiavi, ovvero modificare il meccanismo di crittografia dallo XOR a una qualunque combinazione di operatori matematici e logici.

Di seguito alcuni esempi del risultato (routine di crittografia basata su XOR):

Signature	Generazione	Chiave
A0, 35, 37, 19, E3	programma infettore	----------
5F, CA, C8, E6, 1C	Prima	255 (FFh)
68, FD, FF, D1, 2B	Seconda	200 (C8h)
18, 8D, 8F, A1, 5B	Terza	184 (B8h)

Tabella 3

Una routine basata sullo XOR ed una chiave di 8 bit, moltiplica già il numero delle signature che devono essere riconosciute dal software di rilevazione per 255. Se si utilizzano procedure più complesse il fattore di moltiplicazione può aumentare in modo esponenziale. In questo modo una rilevazione per signature non è più efficace.

Un altra tecnica molto efficace con cui si può ottenere il polimorfismo è la modifica del codice. Questa tecnica consiste in un insieme delle seguenti azioni:

- Inserimento di codice inutile (junk code): si tratta di inserire del codice privo di effetto all'interno del codice del virus in un modo casuale.

Questo si può realizzare con l'aggiunta di istruzioni NOP (nessuna operazione), di combinazioni di push e pop in eguale numero ed in sequenza, dello stesso registro, di operazioni su un registro non utilizzato, ecc.
In tabella 4 si dà un esempio in assembler.

- Scambio del codice: il codice viene suddiviso e vengono rilevate le parti non necessariamente sequenziali, poi queste vengono permutate in modo casuale. Il codice può anche venire spezzato, per essere inserito nel file da infettare in punti diversi.

- Trasformazione del codice: consiste in una sequenza delle seguenti operazioni: istruzioni attigue non sequenziali scambiate, singole istruzioni sostituite da insiemi di istruzioni con lo stesso risultato, ovvero sostituite da chiamate di funzioni nuove.

- Modifica dei registri: i registri utilizzati dalla routine da modificare vengono cambiati. Un esempio in tabella 4.

La particolarità dello spezzare il codice è spesso presente nei virus dedicati al nuovo formato di Windows PE EXE, che comprende al suo interno vari spazi riempiti di valori uguali, spesso zeri. I virus di questo tipo cercano questi spazi e possono suddividere il loro codice in essi; il virus poi può, a tempo di esecuzione, ripristinare gli spazi originali nell'immagine caricata in memoria del file; può anche correggere il file direttamente sul disco prima di restituire il controllo al programma infetto e ri-infettarlo quando il programma termina la sua esecuzione.

Riguardo ancora alla modifica del codice, in realtà l'operazione è più complessa: si noti, riguardo all inserimento di junk code, che una iterazione del procedimento descritto sopra porta ad un allungamento indefinito del virus, cosa assolutamente indesiderata. Per ovviare a questo inconveniente possono esistere dei punti fissi nel codice, che possono o meno contenere del junk code; inoltre il junk code in uno stesso punto si può fare variare da una generazione all'altra, quindi il motore polimorfico deve poter rilevare il suo stesso junk code. È quindi una tecnica estremamente complessa da implementare.

Codice originale	Tipo di modifica	Codice modificato

shl bx,cl
mov si,100h
mov bp,si
add cx,cx
mov [di+1],cx
mov ds,dx
mov [di+2],ax
movsw
	Modifica registri:
	
	bx ⇒ ax
	cx ⇒ bx
	dx ⇒ cx
	ax ⇒ dx
	di ⇔ si
		shl ax,bl
		mov di,100h
		mov bp,di
		add bx,bx
		mov [si+1],bx
		mov ds,cx
		mov [si+2],dx
		movsw

	Scambio del codice: A:shl bx,cl B:mov si,100h B:mov bp,si C:add cx,cx C:mov [di+1],cx D:mov ds,dx D:mov [di+2],ax E:movsw	 C:add cx,cx C:mov [di+1],cx B:mov si,100h B:mov bp,si D:mov ds,dx D:mov [di+2],ax A:shl bx,cl E:movsw

	Junk code:
	Registri ax e bx messi e tolti dallo stack
	
	Operazione nulla
	
	Operazione nulla
	
	Salto alla istruzione successiva
		shl bx,cl
		push ax
		push bx
		pop bx
		pop ax
		mov si,100h
		mov bp,si
		nop
		add cx,cx
		nop
		mov [di+1],cx
		mov ds,dx
		jump $+1
		mov [di+2],ax
		movsw

Tabella 4

A suo tempo vi è stata una denuncia, da parte del dottor Bontchev, della possibile minaccia che può rappresentare una simile tecnica evoluta, per esempio implementata in un motore polimorfico. [45]

La tecnica della modifica del codice è implementata più che altro nelle routine polimorfiche descritte precedentemente; l'MtE, per esempio, produce questo risultato. Riguardo alla tecnica della crittografia, sia essa ottenuta con tool appositi o direttamente, rimane sempre una parte non cifrata, cioè la routine di decifrazione; questa sezione del codice è quella che viene ricercata dai programmi anti-virus. La routine di decifrazione richiede tecniche alternative per non essere rilevata, che possono consistere nel rendere la routine più generica possibile, nell'inserire artifici che rendano difficile la sua simulazione da parte di un anti-virus, nell'implementare su di essa la tecnica di modifica del codice, l'utilizzo di diverse routine crittografiche con medesimo risultato, l'utilizzo di diversi algoritmi con analogo risultato. (cfr. paragrafo 4.1).

Vi sono stati numerosi esempi di tools sviluppati dagli autori di virus in grado di fornire routine di crittografia da collegare alle routine virali per renderle polimorfiche. Il primo è stato il già citato MtE, poi il TpE, il NED, lo SMEG, tra i principali[22]. Si tratta di routine già compilate da unire al proprio codice virale, richiamate mediante una chiamata di funzione. I virus che le utilizzano devono avere determinate caratteristiche per non produrre conflitti con la routine di crittografia; inoltre la dimensione di un virus deve essere rilevata in maniera abbastanza precisa ed essere conosciuta a priori, cosa resa più difficile dalla presenza di un modulo linkato dopo l'assemblaggio del virus. Il risultato è che i virus di questo tipo possono effettuare spesso un insieme limitato di operazioni, pena la possibilità di crash del sistema che esegue il codice virale. Questo problema si può superare con l'utilizzo di motori polimorfici di buona qualità. Lo SMEG, per esempio, che risulta utilizzato solo in circa tre virus, e non è mai stato molto diffuso, presenta un'ottima affidabilità; inoltre, è in grado di implementare, oltre la classica crittografia, una certa forma di modifica del codice attraverso l'inserimento di junk code. Per questo motivo viene considerato molto pericoloso. [73], [analisi dell'autore].

3.3.1.1 - Tecniche speciali relative ai macro virus

La rilevazione di un virus di macro è molto differente dalla rilevazione di un virus in codice eseguibile, o assembler. Il VisualBasic for Applications ed il Word Basic utilizzano

[22] In realtà ne esistono moltissimi, una delle tante mode collegate al mondo dei virus; un sito ucraino, esaminato nel paragrafo 5.2, ne contiene più di 200.

dei formati interni particolari, applicati ad un tipo di file, l'OLE2, molto complesso ed articolato, in grado di contenere un insieme di differenti formati di dati uniti assieme. Per questo è più facile che il virus possa produrre alcune piccole variazioni nel codice, capaci di ostacolarne la rilevazione. Il Basic presenta, tra l'altro, la possibilità di inserire dei commenti all'interno del codice, quindi una forma di polimorfismo molto valida può essere quella di modificare il numero ed il contenuto di questi commenti. Un esempio è dato da WM.Marker, un virus di macro che, ad ogni infezione di un nuovo sistema, inserisce, al fondo del proprio codice, un commento nuovo contenente le informazioni utente del sistema infettato.

Di seguito, alcune tecniche anti-rilevazione dei macro virus, basate sulla modifica del codice:

- Inserire dei commenti random tra gli operatori della macro.
- Rinominare le variabili usate nella macro.
- Inserire tra gli operatori degli altri operatori inutili e privi di efficacia.
- Rimpiazzare alcuni operatori con altri che eseguano le stesse funzioni.
- Scambiare operatori non necessariamente sequenziali.
- Rinominare le macro virali, con nomi diversi oppure generati in modo random.
- Cifrare il codice ASCII di alcune macro virali.

Ognuna di queste tecniche è in grado di produrre dei macro virus equivalenti agli originali ma differenti dal punto di vista morfologico. Una combinazione di queste tecniche può essere usata da un macro virus per implementare un polimorfismo efficiente. Può, per esempio, esistere un virus comprendente macro automatiche, contenenti solo delle chiamate di altre macro e del codice non virale; le macro richiamate potrebbero modificare autonomamente il proprio nome in modo random ad ogni esecuzione/riproduzione e contenere il corpo del virus. In questo modo, le macro automatiche, quindi più sospette, sarebbero esaminate con attenzione; inoltre, il nome delle macro virali, cambiando ogni volta, non sarebbe un elemento valido per il rilevamento.

Una tecnica più complessa consiste nel creare una catena di macro nuove ad ogni riproduzione del virus, le quali apparirebbero e scomparirebbero in modo differente ad ogni

riproduzione o esecuzione. Si potrebbe implementare anche una sorta di chiamata a catena delle macro; in questo modo se qualche macro fosse rimossa, per effetto di un anti-virus, le macro rimaste potrebbero accorgersi ed eseguire qualche payload dannoso [30].

Una tecnica per implementare la crittografia, piuttosto che l'auto-modifica, è fornita in effetti dallo stesso ambiente Office: la possibilità di cifrare i documenti. Un macro virus può implementare la cifratura del documento, cifrandolo al volo al momento della chiusura e decifrandolo al volo al momento dell'apertura. In questo modo, la password di apertura sarebbe conservata dal virus in qualche punto a lui solo conosciuto. L'utilizzatore potrebbe accorgersi dell'accaduto, nel caso in cui trasferisse un suo documento su un altro computer. Per ridurre questa probabilità, il virus potrebbe implementare una sorta di tecnica random, cioè solo alcuni documenti vengono cifrati; oppure lo stesso documento a fasi alterne. In quest'ultimo modo, la presenza di un documento cifrato verrebbe considerato un errore di trasferimento dati, dal momento che, con buone probabilità, la volta successiva l'operazione avrebbe successo. Inoltre si può designare una macro come di "sola esecuzione", il che viene attualizzato dall'Office nel cifrare il corpo della macro con una tecnica a XOR con un solo byte di chiave. Il documento, poi, potrebbe essere totalmente convertito in una stringa ed inserito nel corpo di una macro che lo rigeneri ad ogni apertura. In questo modo, oltre che proteggere il virus stesso, la tecnica provvede una sorta di "vendetta" in quanto la rimozione del virus rende inutilizzabile i documenti disinfettati. La stessa cosa vale per la tecnica di cifrare l'intero documento [30].

Una tecnica considerata *stealth* (cfr. paragrafo seguente), cioè in grado di modificare il sistema al fine di impedire la rilevazione del virus, è quella della sostituzione della voce di menu "Tools/Macro" o nella versione Italiana "Strumenti/Macro", la quale permette di esaminare i tipi di macro contenuti all'interno di un documento o di un ambiente. Come si è già esaminato nella sezione 3, le piattaforme Office presentano la possibilità di sostituire le operazioni avviate dalle voci del menu di sistema con delle altre inserite in apposite macro. La tecnica citata consiste nel sostituire a questa voce di menu una macro che visualizzi una finestra vuota, o che intercetti il risultato al fine di non visualizzare i nomi delle macro virali inserite nel documento.

3.3.2 - Virus stealth

Questo nome deriva da un tipo di aereo da caccia, lo stealth, che contiene degli apparati elettronici in grado di renderlo invisibile ai radar. I virus di questo tipo includono una sezione residente in memoria; queste routine residenti si interpongono alle routine del sistema operativo che vengono utilizzate dagli anti-virus per scandire le unità di memorizzazione ed impediscono che i programmi anti-virus possano accedere ai dati del virus. Possono anche impedire che alcuni particolari sospetti siano rilevati dall'utente, come le variazioni della lunghezza dei file.

Si presenta ora un insieme di tecniche utilizzate da alcuni virus:

- Tecniche applicate al settore di avvio.

Questi virus alterano il contenuto del settore di boot inserendo una copia di se stessi e, in certi casi, copiando il settore originale in un settore vuoto del disco infettato. Poi inseriscono una routine che "aggancia" la funzione del sistema operativo incaricata di leggere un determinato settore di un disco, funzione denominata nei sistemi operativi Microsoft INT 13h. Al momento della chiamata, in realtà, il sistema operativo dà il controllo alla routine inserita dal virus, la quale esamina la richiesta e, normalmente, restituisce il controllo al seguito della funzione reale; se invece rileva la richiesta di leggere il settore zero, la routine virale non trasferisce il controllo alla normale routine del sistema operativo ma fornisce essa stessa la risposta, utilizzando il settore originale salvato sul disco in precedenza, oppure un'immagine corretta ricostruita ad hoc. In questo modo tutto appare normale, non è possibile rilevare il virus né con programmi anti-virus né con una lettura diretta dei settori del disco.

I virus che utilizzano questa tecnica sono diversi: AntiEXE, Bye, Bones, Michelangelo, per citarne alcuni. Il virus Civil_Defense è, inoltre, un virus *multipartito*, che è quindi in grado di infettare sia settore di avvio sia file. Esso, quando si trasmette via file, infetta in un sistema pulito il settore di avvio, dopodiché si cancella dal file infettore; fino al successivo avvio non viene installato in memoria, poi non può più venire rilevato e riprende ad infettare i file. Il virus fa sfuggire al controllo anche 129 settori che sono utilizzati dal virus per il settore originale ed il corpo del virus. Vi sono ancora dei virus che memorizzano

parti di se stessi o il settore di avvio originale in zone del disco non accessibili normalmente, alterando i parametri dell'unità; questa tecnica è utilizzata, per esempio, dal virus Neuroquila, che formatta una traccia extra sui dischetti per memorizzare il corpo del virus.

- Tecniche applicate ai file.

Le tecniche applicate ai file consistono normalmente nel prendere possesso della system call INT21h che gestisce i file. I virus possono effettuare diverse operazioni: possono in caso di lettura del file da parte di uno scanner disinfettare il file al volo, come il virus Frodo. Il virus Byway, invece, utilizza un sistema molto interessante: esso, al momento di infettare un sistema pulito, crea un file nascosto di nome chklist#.ms#, con # che rappresenta il carattere ASCII 255, contenente il codice principale del virus; poi, ad ogni file che deve essere infettato, viene spostato il primo cluster di dati nei byte riservati della voce di directory e la FAT è alterata, in modo che il primo cluster del file sia il cluster di chklist#.ms#. In fase di esecuzione, il controllo viene passato al codice contenuto in questo file, che installa il virus in memoria; poi da questo momento ogni chiamata al file infetto viene ridiretta al cluster memorizzato nell'area delle directory, che è anche cifrato. Il file risulta così corretto e funzionante ed è impossibile rilevare l'infezione mediante scansione. Inoltre anche le unità floppy vengono infettate nello stesso modo; la riproduzione di Byway avviene, infatti, tramite scambio di dischetti infetti. Altri virus, come Pieck, agganciano le system call di lettura dei file ed impediscono che il virus sia rilevato in scansione, correggendo il risultato fornito dalle system call di lettura.

Gli anti-virus possono essere in grado di rilevare la presenza del virus in memoria, però non sono in grado di effettuare altre operazioni finché il virus è residente. Per questo motivo, se il programma anti-virus rileva in memoria le tracce della presenza di un virus, oppure che le funzioni del sistema operativo o la tabella di queste funzioni sono state alterate in modo anomalo e sospetto, ovvero rileva la presenza di una funzione virale residente in memoria, normalmente non tenta di correggere il problema ma richiede all'utilizzatore di spegnere il computer e fare avviare il sistema operativo da un disco originale o di riserva, non infettato.

3.3.3 - Retrovirus

Con il termine retrovirus si identificano virus che eseguono delle operazioni finalizzate ad attaccare direttamente uno o più anti-virus. Questo attacco si concretizza in due modi: 1. il virus può cercare e cancellare i file di controllo degli anti-virus, che contengono le dimensioni dei file o i loro checksum (vedere paragrafo 4.1.3 sui sistemi di verifica d'integrità); può anche cancellare direttamente il programma anti-virus, però questo è più raro e molto meno efficace. 2. il virus cerca le parti residenti degli anti-virus, detti behavior blocker, e tenta di disinstallarle dalla memoria. Queste routine sono quelle che scandiscono i file in fase di lettura o apertura e vigilano sulla esecuzione di azioni sospette, come modifica di un file eseguibile, modifica del settore di avvio di un disco, ecc.

Il virus Tremor, per esempio, cerca e disinstalla dalla memoria l'anti-virus residente Vsafe e la parte residente di MSAV (Microsoft Anti-Virus); il virus Vampiro utilizza delle system call non documentate con le quali tenta di disinstallare gli anti-virus residenti Vsafe, Vwatch, Pctools V8+. Il virus Groove, invece, cancella i database di checksum dei programmi anti-virus MSAV, CPAV (Central Point Anti-Virus), cancellando anche i file programma VSAFE.COM e CPAV.EXE. I programmi CPAV e MSAV utilizzano la tecnica di inoculazione, memorizzando i dati dei checksum in file presenti nelle directory esaminate. Il virus Neuroquila attacca numerosi anti-virus, se rileva che VIRSTOP o DOSDATA.SYS (un programma di utilità della suite QEMM) vengono caricati da CONFIG.SYS, il virus impedisce che vengano caricati in memoria; inoltre esso tenta di modificare i programmi TBDRIVER, TBDISK, VSAFE e -D mentre sono in memoria, al fine di impedirne il funzionamento; modifica anche il settore di protezione generato da TBUTIL e, inoltre, è capace di bloccare i messaggi di errore prodotti da Windows, in seguito a scrittura diretta sul disco, quando è attivo il sistema di gestione dei dischi a 32 bit, uno scoglio insuperabile per molti virus [63].

4 – Programmi anti-virus

In questo capitolo si trova un'analisi delle tecniche principali utilizzate dai programmi anti-virus, assieme all'analisi di alcune innovazioni. Vengono anche presentate le fonti di rilevazione dei nuovi virus. Si analizzano, infine, le caratteristiche dei principali prodotti esistenti.

4.1 – Funzionamento di un anti-virus

È necessario evidenziare, al fine di una corretta comprensione della struttura del capitolo, che gran parte delle informazioni necessarie sono conservate gelosamente dai ricercatori delle aziende, quali segreti industriali. La concorrenza tra diversi prodotti, nell'industria anti-virus, è particolarmente evidente e critica: attualmente, secondo diverse fonti, sono stati rilevati più di 50000 virus ed ogni anti-virus comunica all'utente quale è il numero di virus rilevati, per non menzionare il problema della rilevazione dei virus polimorfici, dei virus stealth e dei virus ancora sconosciuti. Vi è quindi una forte concorrenza per la produzione di un prodotto in grado di rilevare più virus dei concorrenti e in grado di effettuare questa operazione in un breve lasso di tempo.

Molte delle informazioni presentate in questo paragrafo sono frutto di comunicazioni personali con il dottor Vesselin Bontchev, ricercatore presso la Friskies international e membro del team di ricerca sui virus dell'università di Amburgo, autore di numerosi contributi di carattere scientifico sull'argomento.

È noto che alcuni problemi di programmazione sono indecidibili. Un esempio è il cosiddetto problema "dell'alt", che si può enunciare, a grandi linee, nel modo seguente: "si dimostra che il problema di scrivere una funzione calcolabile g capace di determinare se un'altra funzione calcolabile f, dato un input x, è determinata, è indecidibile", ovvero non è possibile scrivere un programma capace di rilevare se un altro programma, dato un certo input, darà la soluzione o si bloccherà. Una conseguenza dell'indecidibilità del problema della fermata è l'indecidibilità di altri problemi correlati, come il verificare se due programmi eseguono la stessa funzione. Da questo deriva che non è possibile scrivere un

programma in grado di determinare se un altro programma, fornito come input, ha una certa funzione o no, poiché i programmi, in generale, non si possono confrontare in modo algoritmico. La conseguenza che riguarda il campo anti-virus è che non è possibile scrivere un programma che sia in grado di osservarne un altro e determinare in modo certo se è o non è un virus. La dimostrazione è la seguente [8]: posto che P sia un programma da analizzare e D(S) sia la procedura che determina se S è un virus, allora si può introdurre in P una chiamata alla procedura D, in modo che P si comporti come un virus se D(P) è falso e non lo faccia se D(P) è vero, generando una contraddizione. È quindi necessario cercare soluzioni incrementali, cioè che vengono modificate per ogni particolare virus, in modo da comprenderne la rilevazione; oppure si utilizzano tecniche valide a meno di una certa possibilità di errore. Questo problema si rileverà in ognuna delle tecniche descritte; il suo principale effetto è di impedire lo sviluppo di soluzioni capaci di rilevare i virus in modo sicuro, senza possibilità di errori e di falsi allarmi.

Gli strumenti disponibili per contrastare il fenomeno dei virus appartengono ad una delle seguenti classi:

- Scanner: sono dei programmi che "scandiscono" le aree in cui si può trovare un virus, memoria RAM, settore di avvio, file, e con differenti tecniche cercano di rilevarne la presenza analizzando il codice scandito.
- Verificatori di integrità: si tratta di programmi che creano un database di codici di controllo dei file e dei settori di avvio delle unità monitorate, usando per esempio il Check Redundant Code o CRC, e periodicamente, in modo automatico o dietro richiesta, verificano se vi sono state modifiche non previste.
- Behavior blocker o monitor: sono programmi residenti in memoria che analizzano tutte le operazioni svolte per individuare operazioni tipiche dei virus o dei payload connessi, come accesso in scrittura ad un file eseguibile o accesso diretto al settore di avvio di un disco, sospendendone lo svolgimento e segnalando all'utente l'operazione ed il nome del programma che l'ha richiesta. Possono anche effettuare scansioni dei file di programma immediatamente prima che essi vengano eseguiti.
- Digital immune systems: si tratta di software innovativi, introdotti per contrastare la rapidissima crescita e diffusione dei virus utilizzando i potenziali delle reti di calcolatori. Essi sono capaci di rilevare possibili infezioni di virus sconosciuti con sofisticate tecniche euristiche ed inviarli in modo automatico a centri di calcolo dedicati, in grado

di verificare l'attendibilità della segnalazione e generare le necessarie contromisure senza intervento umano, eccetto casi particolari.

La tecnica fondamentale e più usata, tra quelle presentate, è quella della scansione, molto spesso unita ad alcune delle altre tecniche. Il principio di funzionamento di uno scanner anti-virus è, a grandi linee, semplice: si tratta di esaminare la memoria, il settore di avvio delle unità a disco ed i file presenti in esse alla ricerca di virus o di tracce di questi. Per effettuare questi compiti si utilizzano due tecniche: la ricerca di signature e la ricerca euristica: la prima orientata a rilevare virus, con relative varianti, già conosciuti ed analizzati; la seconda orientata a rilevare nuove varianti di virus sconosciuti e nuovi virus non ancora rilevati.

L'ultima azione importante è la rimozione del virus rilevato, o cleaning. Quasi tutti i prodotti anti-virus basati sulla rilevazione esatta dei virus includono l'opzione, ove possibile, di rimuovere i virus rilevati e ripristinare così la funzionalità iniziale del programma infettato.

Nei successivi paragrafi vengono analizzati in dettaglio le tecniche di scansione ed alcune delle tecniche di supporto presentate.

4.1.1 – Scansione attraverso la ricerca di signature

La parola signature significa, in inglese, firma; si tratta quindi di cercare la "firma" di un virus, che consiste in una stringa di dati, rilevata all'interno della parte fissa di un virus, che lo caratterizza. Un esempio è capace di evidenziare il concetto di signature: si ponga di avere il seguente frammento di codice assembler, con relativa codifica in linguaggio macchina, tratto da un programma completo:

```
Linguaggio    Assembler
Macchina

58            POP AX
354F21        XOR AX,214F
50            PUSH AX
254041        AND AX,4140
50            PUSH AX
5B            POP BX
```

```
345C          XOR AL,5C
50            PUSH AX
5A            POP DX
58            POP AX
353428        XOR AX,2834
50            PUSH AX
5E            POP SI
2937          SUB [BX],SI
43            INC BX
43            INC BX
2937          SUB [BX],SI
7D24          JGE 0140
45            INC BP
49            DEC CX
43            INC BX
41            INC CX
```

Se si leggono di seguito i valori esadecimali delle istruzioni tradotte in linguaggio macchina, si ottiene la seguente stringa di 32 dati in formato esadecimale:

```
58 35 4F 21 50 25 40 41 50 5B 34 5C 50 5A 58 35
34 28 50 5E 29 37 43 43 29 37 7D 24 45 49 43 41
```

Questa stringa è una "signature" relativa al codice da cui è stata tratta, in questo caso si tratta del test dell'Eicar per i programmi anti-virus. Posto di avere scelto questa signature per rilevare il virus "Eicar.test.file", essa verrà inserita in un database di stringhe; l'anti-virus che dovesse utilizzare questo database per una scansione di signature, al momento dell'esecuzione, inizierebbe a caricare in memoria la prima parte di dati da controllare, sia essa proveniente da un file, dal settore di avvio o dalla memoria, e a scandirla alla ricerca di una delle signature o stringhe del database. Se non viene rilevata nessuna signature il programma può considerare la porzione di codice analizzata sicura e proseguire con altre scansioni, oppure iniziare una ricerca euristica di codice virale; in caso contrario vi sono diverse scelte che dipendono da alcuni parametri oggettivi e soggettivi. Si noti inoltre che le signature possono contenere degli elementi variabili; si consideri il codice del seguente esempio :

```
B8 ?? ??          mov ax, offset ADDRESS
8B F0             mov si,ax
FF 24             jmp [si]
```

e si ponga che il dato memorizzato in ax possa variare in ogni diversa infezione. La signature diventa B8 ?? ?? 8B F0 FF 24, in cui il secondo e terzo valore non sono significativi.

La scelta della signature di un virus deve essere eseguita tenendo conto di alcuni fattori. I virus possono essere lunghi alcune centinaia e anche migliaia di byte, quindi, considerando che sono stati rilevati circa 50000 virus, la signature deve essere breve; è comunque estremamente sconsigliabile inserire porzioni significative del codice di un virus in un prodotto destinato alla distribuzione, per motivi di sicurezza. D'altra parte, una signature troppo breve ha una maggiore probabilità di trovarsi anche in altri frammenti di codice, relativi a programmi non virali, e dare quindi origine ad un falso allarme, o *falso positivo*. Deve, inoltre, essere scelta in una zona del virus significativa: tipicamente i ricercatori scelgono una porzione di codice che appaia inusuale e si trovi in una zona principale del codice virale, quindi di difficile modifica; questo sia per il problema dei falsi positivi che per ottenere una stringa valida anche in caso di varianti. La dimensione delle signature va dai 16 ai 32 bytes; considerando che una signature di 16 bytes ha una probabilità di presentarsi pari a $1/(256^{16})$ cioè dell'ordine di 10^{-38}, la probabilità di falsi positivi dovrebbe essere trascurabile. In realtà, questo computo è valido solo per dati grezzi; invece, le signature vengono tratte da codice eseguibile, quindi non tutte le combinazioni sono valide ed inoltre, anche tra le combinazioni valide, moltissime sono prive di senso. La probabilità reale non è per nulla trascurabile; per questo motivo la scelta della signature è una procedura critica. Il compito diventa particolarmente difficoltoso nel caso di virus scritti in un linguaggio di programmazione diverso dall'assembler, come il C o il Pascal; questo perché i compilatori inseriscono moltissimo codice preso verbatim dalle librerie, quindi una stringa significativa deve essere oltremodo lunga oppure cercata con grande attenzione. Nel paragrafo successivo viene presentato un algoritmo, creato in un centro ricerche dell'IBM, capace di estrarre automaticamente signature valide da file infetti, assieme ad una recensione di pubblicazioni su tecniche di scansione rapida.

Di per se, il semplice rilevamento di una signature non è sufficiente ad esaurire il compito di un anti-virus: perciò si analizzano ora le azioni seguenti la rilevazione di una signature. Una fondamentale selezione dipende dal luogo in cui si è rilevata la stringa:

- Memoria RAM: se una stringa virale si trova in memoria, all'interno di un programma residente oppure di una sezione del sistema operativo, come le routine di servizio degli interrupt, normalmente se ne deduce che è in corso un'infezione ed un virus è residente in memoria. In questo caso, al fine di prevenire le funzioni di stealth, spesso gli anti-virus evidenziano quanto rilevato e si fermano, evidenziando una procedura consigliata

consistente nello spegnere il computer ed avviarlo da un disco di avvio sicuro conservato separatamente, detto disco di ripristino, che permette all'anti-virus di funzionare con la memoria libera.

- Settore di avvio: una stringa rilevata in questa posizione, ammesso che sia relativa ad un virus del settore di avvio, segnala un'infezione di un virus particolare; normalmente i boot virus contengono delle sezioni che vengono installate in memoria, quindi anche in questo caso, prima di intraprendere azioni correttive, si evidenzierà il consiglio di effettuare un avvio da un disco sicuro.
- File: il rilevamento di una stringa in questa posizione viene accompagnata dalla segnalazione di quanto rilevato, dopo aver normalmente effettuato altri tipi di verifica, ed, eventualmente, dalla possibilità di intraprendere azioni correttive.

Quando si verifica il match di una stringa, è possibile e consigliabile eseguire altre verifiche prima di segnalare la presenza di un virus, in dipendenza anche del luogo di rilevazione. Esiste un problema, già in parte evidenziato, noto come quello delle varianti: possono esistere virus coincidenti a meno di lievi differenze oppure solo parzialmente coincidenti, originati da uno stesso virus, detti varianti. Una variante nasce per cause diverse: può essere lo stesso virus, solo compilato o assemblato dall'autore con differenti versioni o tipi di compilatore e assemblatore; oppure può essere il frutto di esperimenti successivi, eseguiti dall'autore o da altri, sul codice originale del virus. Come è intuibile, è indispensabile riconoscere l'esatta variante, per la rimozione del virus da un file. Per verificare se la variante rilevata corrisponde ad una di quelle conosciute si eseguono delle verifiche: si può calcolare un codice di controllo, come il CRC, della parte di virus non variabile conosciuta e confrontarlo con quello o quelli memorizzati nel database del programma anti-virus, per rilevare quale sia l'esatta variante e se si tratta di una variante conosciuta. Tra i sistemi utilizzati per rilevare, assieme ai virus, anche le loro varianti, vi è quello di permettere un match parziale della stringa, con una soglia di ammissibilità; in questo caso sarebbe comunque necessario effettuare una verifica, con CRC od altro, per rilevare se la variante è già tra quelle conosciute. Altri sistemi consistono nel conservare diverse signature per ogni virus, alcune principali ed altre, più brevi, utili per rilevare le varianti; questa viene considerata già una tecnica euristica. Nel caso di rilevazione di una variante sconosciuta, alcuni software effettuano comunque una verifica euristica sul corpo del presunto virus, per rendere minimo il rischio di falsi positivi; lo scopo di tale ricerca, come si vedrà nel paragrafo successivo, è di rilevare comportamenti classici di un virus, in

particolare la capacità di riprodursi e la capacità di compiere azioni dannose o distruttive (ricordiamo che con lo stesso procedimento vengono cercati anche i più diffusi cavalli di Troia e le backdoor, nonostante questi malware siano molto meno soggetti a mutazione dei virus e quindi facilmente rilevati dalla scansione per signature). È, comunque, di una certa utilità la capacità di rilevare una variante sconosciuta di un certo virus, con la relativa comunicazione all'utente, poiché è logico supporre che almeno una parte delle caratteristiche originali del virus si trovino nella sua variante. Conoscendo le caratteristiche della famiglia virale di appartenenza, si possono valutare azioni di profilassi e si può azzardare una valutazione della gravità dell'infezione, nell'attesa che il laboratorio di ricerca, normalmente contattato in questi casi, fornisca un software in grado di operare la disinfezione.

Una causa di variabilità molto importante, che necessita un'analisi separata, è quella introdotta dal polimorfismo. Una delle conseguenze della co-evoluzione dei virus e degli anti-virus è stata di introdurre dei sistemi per rendere differente il virus ad ogni sua riproduzione, totalmente o in parte, proprio per impedire la rilevazione di una signature valida all'interno del codice (paragrafo 3.3.1). Principalmente le tecniche seguite sono: la cifratura del codice, l'inserimento di codice inutile, la modifica del codice. I virus polimorfi che utilizzano la cifratura del proprio codice, debbono anche decifrarlo prima che esso venga eseguito; quindi vi sarà una routine di decifrazione in chiaro in testa al codice del virus. Questa routine permette certamente l'estrazione di una signature, però il procedimento è più complesso. Una routine di decifrazione è relativamente breve rispetto al corpo di un virus; inoltre, si tratta di operazioni svolte da moltissimi software per differenti motivi; quindi il rischio di falsi positivi cresce moltissimo, in modo inaccettabile. Una possibile soluzione è di utilizzare una tecnica denominata G.D. o general decryptor: quando viene rilevata una stringa caratteristica di una routine in grado di decifrare, il GD, che normalmente comprende una macchina virtuale in cui si può simulare e monitorare l'esecuzione del codice scandito, esegue la routine rilevata e verifica se essa sta modificando il codice. In questo caso, il GD decifra la parte cifrata e la scansione viene ripetuta sul codice decifrato, alla ricerca di eventuali signature; oppure si può eseguire una ricerca euristica, descritta in seguito, per rilevare la presenza di virus sconosciuti. Chiaramente, per la rilevazione di questo genere di virus è necessario eseguire l'estrazione della o delle signature dal codice in chiaro; questo presenta problemi che rendono più complessi gli algoritmi di estrazione automatica di signature, come quello descritto in

precedenza. Anche la tecnica di modifica del codice, come quella della rotazione dei registri, può essere neutralizzata: utilizzando un match parziale di signature, verificando se la parte di signature rilevata può rivelare una rotazione di registri (il caso più comune) ed eseguendo, nel caso, un algoritmo che rigeneri la situazione iniziale. In ogni caso, l'inserimento di codice inutile e la possibilità di permutare parti di codice non sequenziali, ovvero di utilizzare un insieme di routine di crittografia differenti con medesimo funzionamento, può portare un numero di varianti sufficiente a rendere non più vantaggiosa una ricerca per signature di queste routine. Viene effettuata da molti prodotti una ricerca mista per signature ed euristica delle routine di crittografia ed in caso di sospetti sufficienti le routine vengono eseguite in una macchina virtuale, ed il codice decifrato viene a sua volta scandito alla ricerca di virus; si tratta dunque di una tecnica GD con euristica. Utilizzando una tecnica analoga, anche se molto semplificata, viene eseguito il controllo sui file compressi; un buon prodotto deve poter eseguire verifiche anche su file compressi più volte ricorsivamente.

Una nota a parte richiedono i worm e i macro virus. Al riguardo dei macro virus, (ci si riferisce ai macro virus capaci di infettare i programmi della suite Microsoft Office) essi sono contenuti in un formato particolare di file denominato OLE2, il quale gestisce diversi tipi di dati assieme attraverso una gestione del suo spazio paragonabile a quello del file system di un'unità a disco. La grande possibilità di alterare una macro, che in realtà è un file di testo, rende difficile la verifica attraverso signature. Infatti, nel codice delle macro vi possono essere degli spazi aggiunti, volontariamente od involontariamente, da qualche esperimento di curiosi; anche la modifica di uno o più nomi, o la modifica dell'ordine di memorizzazione delle macro, se più di una, è estremamente semplice. Inoltre, lo stesso meccanismo di funzionamento delle macro prevede facili metodi per modificare autonomamente il codice. C'è anche da aggiungere che una signature significativa per un macro virus può essere molto lunga e comprendere gran parte del codice del virus, questo per un fattore intrinseco del linguaggio di programmazione delle macro. Per queste ragioni la scansione di un documento alla ricerca di macro virus richiede spesso tecniche aggiuntive di verifica, normalmente euristiche. Altri sistemi di scansione di macro virus consistono nella ricerca di macro con un certo checksum: in fase di scansione il software anti-virus calcola il checksum di ogni macro rilevata e lo confronta con quelli presenti in un database. Le macro possono essere precedentemente normallizzate, eliminando gli spazi multipli e rendendo minuscoli o maiuscoli tutti i caratteri, per esempio. Se vengono rilevati i

checksum di tutte le macro comprendenti un determinato virus, si può considerare con certezza il documento infetto. Se invece ne viene rilevato solo un sottoinsieme, si può segnalare la presenza di un macro virus variante, richiedendo per ulteriore conferma che la verifica euristica riveli meccanismi riproduttivi. Quest'ultimo metodo presenta il vantaggio di non richiedere grandi quantità di spazio per l'immagazzinamento delle signature e di essere indipendente dall'ordine di memorizzazione delle macro, inoltre non viene diffuso macro codice virale. Una tecnica analoga può essere utilizzata per gli script virus ed altro codice virale simile.

I worm, per la loro stessa natura, raramente si trovano nel file system, e comunque dopo essere divenuti attivi. Si trovano nelle principali aziende anti-virus dei software appositi che scandiscono la posta elettronica prima che essa sia rilevata dal client, e questo può anche essere fatto con una scansione di signature, ma il problema presenta caratteristiche particolari che rendono questa tecnica poco significativa; i worm in effetti hanno un tempo di diffusione estremamente rapido e il loro periodo di massima attività è estremamente più breve di quello dei virus. Una scansione di signature, quindi possibile dopo che il worm è stato rilevato e studiato, è utile a lungo termine, ma nel momento in cui il worm produce i maggiori danni, cioè i primissimi periodi di diffusione, non può essere effettuata. Questo problema, molto recente, viene affrontato con diverse tecniche innovative, come quella dei sistemi immuni, o con analisi comportamentali, quindi euristiche. È un problema destinato a crescere, poiché è molto probabile la comparsa di differenti software che gestiscano uno scambio di informazioni in rete, con la possibilità di trasmettere anche descrizioni di istruzioni, da svolgere in modo automatico.

4.1.1.1 – Algoritmi notevoli relativi alla ricerca di signature

Per quanto concerne il problema del rilevamento della signature, gli algoritmi sono quelli classici di ricerca di una sottostringa in una stringa, un problema noto nel campo dell'informatica e già affrontato in diversi modi. A differenza dei primi tempi, in cui il numero di virus conosciuti era limitato a poche decine, al più qualche centinaio, la velocità di esecuzione della scansione è un parametro fondamentale per i prodotti odierni. Si sono

verificati casi in cui tutto lo scanner di un prodotto ha dovuto essere riprogettato, per implementare algoritmi moderni di scansione rapida.

Un insieme di referenze a pubblicazioni sul tema può essere il seguente [41]:

Boyer R. S., Moore J.S., *A Fast String Search Algorithm*, Communication of ACM, ottobre 1977, p. 762-772.

Roger Riordan, *Polysearch: An Extremely Fast Parallel Search Algorithm*, proceedings of 5th Computer Virus and Security conference, New York, 1992, p. 631-640.

Sun Wu, Udi Manber, *Fast Text Searching With Errors,* Department of Computer Science, University of Arizona, Tucson, TR 91-11.

Il centro ricerche della IBM ha presentato nel 1994 un algoritmo per l'estrazione automatica delle signature da file infetti, in una pubblicazione, tra l'altro, molto ricca di dati sul problema delle signature in generale [35]. In questa pubblicazione si esamina l'algoritmo, che consente di automatizzare la procedura di estrazione di signature valide automatizzando le seguenti procedure: riproduzione di un virus in un ambiente protetto composto da diversi file "cavia", rilevazione degli esemplari infettati attraverso la verifica di integrità, mediante CRC o simili, spostamento in un ambiente separato e confronto dei file modificati per rilevare la posizione e il corpo dei virus, confronto delle sezioni aggiunte per rilevare le parti fisse e quelle variabili del virus fatto riprodurre, generazione di una o più signature, valutazione delle signature e scelta delle migliori. Il procedimento illustrato viene considerato dagli stessi autori non particolarmente rilevante, suscettibile di essere implementato in modo efficiente in molti altri metodi, compreso il rilevamento umano, se non per la parte relativa alla valutazione e alla scelta delle signature più significative. Il principio, è quello di considerare la probabilità della signature di occorrere all'interno di un vasto insieme di programmi, come una soglia significativa della bontà della signature stessa.

4.1.2 – Ricerca euristica

Il termine euristica deriva dal greco "heuristikein" che significa cercare, scoprire. Nel campo della computer science, si utilizza per descrivere una classe di algoritmi capaci di risolvere rapidamente problemi complessi, al prezzo di non essere in grado di trovare la soluzione ottimale o di introdurre una certa percentuale di errore; queste tecniche sono usate comunemente nel campo dell'intelligenza artificiale.

Nel campo anti-virus, la ricerca euristica, che si realizza con un insieme di tecniche diverse, consiste nel tentare di rilevare un virus sconosciuto, cioè i cui dati non sono presenti nel database dell'anti-virus che lo rileva. Questo non avviene soltanto nel caso di virus nuovi: come si è visto il numero attuale dei virus conosciuti è di circa 50000, molti però si considerano praticamente estinti o non realmente diffusi; la lista dei virus "In The Wild" (par. 5.4) ne comprende solo poche centinaia. Una strategia valida può essere quella di inserire nel database soltanto le signature ed i dati relativi ai virus realmente diffusi; d'altra parte la possibilità di rilevare un virus raro non è nulla. Si consideri, inoltre, che una collezione veramente completa è piuttosto rara, anche per le principali aziende anti-virus. È quindi una parte integrante della fase di rilevazione, la ricerca attraverso l'analisi del potenziale comportamento e dell'aspetto del codice. Si tratta di selezionare e rilevare i segnali che possano fare comprendere, o almeno sospettare con una certa probabilità, la presenza di un virus. Come abbiamo già osservato, la ricerca di un generico virus è un problema indecidibile; il problema viene affrontato con soluzioni che offrono una certa probabilità di successo e sono comunque suscettibili di aggiornamenti incrementali. Quanto detto si applica anche alle varianti sconosciute di virus noti. Una nuova variante può venire rilevata, dalla scansione iniziale per signature, come un match parziale; per decidere se si tratta di un falso positivo o una variante nuova è necessario verificare la presenza di comportamenti virali, proprio attraverso le tecniche euristiche.

Si analizza ora l'implementazione delle principali tecniche. A grandi linee si tratta di esaminare una porzione di codice e di verificare se vi sono descritti dei comportamenti comuni per un virus informatico. Alcuni esempi di comportamenti considerati sospetti sono:

- Comportamenti tipici di un codice auto replicante: ricerca di file, possibili bersaglio di infezione, nell'albero delle directory; apertura in scrittura e aggiunta di dati a possibili file bersaglio; sovrascrittura di porzioni di file bersaglio; presenza di codice che copia se

stesso in memoria, non usando le chiamate di sistema per rendere il codice residente, o in un file.

- Capacità di produrre danni o modifiche non evidenti ai dati memorizzati: accesso diretto ai settori del disco; chiamata di system call di formattazione; cancellazione di più file non generati dal codice; variazione dei time stamp dei file; sovrascrittura di file eseguibili.
- Meccanismi classici dei virus: alterazioni non comuni dello stack; tecniche di allocazione di memoria non standard; auto-modifica di porzioni del codice o dell'header del file; variazioni della tabella degli indirizzi delle system call; installazione di moduli, residenti in memoria, agganciati alle chiamate di sistema.
- Meccanismi tipici per contrastare la rilevazione: algoritmi di decifrazione; presenza di codice inutile; istruzioni non valide; costrutti di salto o di reperimento di dati inutilmente tortuosi; chiamata di system call non documentate.

Appare subito evidente che alcuni di questi comportamenti possono trovarsi in programmi perfettamente validi; vi possono poi essere dei comportamenti opposti, cioè che difficilmente si possono rilevare in un virus; se tali azioni vengono rilevate all'interno delle subroutine sospette, ne fanno decadere la probabilità di essere codice virale. Queste azioni possono essere, ad esempio, intensa interazione con l'utente oppure creazione di nuovi file non contenenti codice eseguibile. Viene quindi assegnato una sorta di punteggio ai dati esaminati, e si segnala una possibile infezione solo sopra una certa soglia. Tale soglia può essere innalzata dalla rilevazione di parti di signature; in questo caso si parla di verifica euristica di nuove varianti ed è quindi logico aumentare la probabilità minima di infezione.

L'analisi euristica del codice si può effettuare in due modi, statico e dinamico. L'euristica statica consiste nel cercare i comportamenti indicati attraverso un insieme di brevi signature. Si consideri per esempio il seguente frammento di codice:

```
8B F0            mov si,ax
FF 24            jmp [si]
```

il significato è il seguente: sposta il contenuto del registro AX nel registro SI, poi esegue un salto all'indirizzo indicizzato in memoria dal registro SI (questo vuole dire che all'indirizzo contenuto in SI si trova l'indirizzo a cui verrà effettuato il salto). Si immagini, puramente a titolo di esempio, che una simile tecnica di esecuzione dei salti sia spesso utilizzata dai virus, poniamo in fase di decifrazione del codice. Si nota subito che questo comportamento

è rilevato dalla stringa 8B F0 FF 24, che è una piccola signature. Un analizzatore euristico di tipo statico conterrà un database di piccole signature caratteristiche dei comportamenti sospetti, unito ad alcune regole finalizzate a determinare se i comportamenti rilevati, nella quantità e nell'ordine rilevato, sono sufficienti a segnalare un comportamento sospetto. Questa tecnica ha un limite notevole: sebbene si fornisca un ampio database di signature di comportamenti, uno stesso segmento di codice può essere trasformato in un segmento differente, ma con identico comportamento, infinite volte. Si può trasferire i dati in diversi registri, sommare e sottrarre ad un dato importante lo stesso valore, scambiare codice non sequenziale, aggiungere codice inutile, ecc.

L'analisi dinamica consiste nel simulare il codice da analizzare all'interno di una simulazione della CPU o della macchina virtuale specifica del codice (caso dei macro virus, virus dei file batch, virus di codice sorgente ecc.). In questi scanner si trova una simulazione della memoria in cui è possibile rilevare le modifiche e le operazioni eseguite dal codice. Vi sarà un insieme di regole di osservazione e di simulazione finalizzate a rilevare tutti i comportamenti sospetti in modo efficiente. Si ponga di voler verificare se il codice può modificarsi: si inizia a simulare l'esecuzione del codice sospetto e si esamina se esso cambia di aspetto in una zona dove normalmente questo non deve accadere; nel caso, si prosegue la simulazione fino a terminare la sequenza di modifica e poi si esamina se il codice che è stato decifrato viene o può venire eseguito. Le condizioni evidenziate, riguardo all'area non standard (normalmente i dati di un programma vengono immagazzinati assieme in una zona prossima alla fine del file che li contiene) e alla verifica sulla reale esecuzione del codice decifrato, sono un esempio delle verifiche aggiuntive necessarie ad impedire che comportamenti normali vengano scambiati per segnali della presenza di un virus. Nel caso presentato, il codice decifrato potrebbe essere semplicemente una stringa di dati in fase di elaborazione, è pero molto difficile che un programmatore implementi una funzione che deve decifrarsi runtime. Una simile porzione di codice, dopo essere stata decifrata verrà analizzata con grandissima attenzione e, in questo caso, ogni minimo indizio di comportamento virale potrà generare una segnalazione di virus sconosciuto. Con questa tecnica, i virus polimorfi vengono facilmente rilevati. Inoltre, non è pensabile simulare il comportamento di tutto il codice dei file esaminati: possono esservi dei file lunghi decine e decine di megabyte, mentre è noto che il tempo di esecuzione è un parametro non trascurabile. Per questo vengono simulate le porzioni di codice che si trovano in aree classiche, come il principio e la fine dei file e l'area vicino all'entry point del programma; in

altre posizioni, le simulazioni vengono eseguite solo in seguito ad altri segni che le rendano consigliabili; per esempio si può avere una scansione mista statica e dinamica, con simulazione del codice presente nelle aree tipiche e nelle aree in cui si rilevino delle signature di particolari comportamenti sospetti, come si vedrà in seguito.

Per rilevare alcuni tipi di virus, come i macro virus, l'euristica dinamica è un bisogno quasi irrinunciabile al fine di ottenere efficienza. I macro virus presentano notevoli difficoltà di rilevazione, insite nel fatto di essere scritti in un linguaggio ad altissimo livello, in cui la maggior parte delle operazioni tipiche di un virus può essere facilmente incluso in operazioni di routine; inoltre, il linguaggio delle macro è interpretato, per cui il codice da analizzarsi è un testo, soggetto a tutte le alterazioni casuali di un testo, come maiuscole o minuscole in posizioni sbagliate, spazi aggiunti, commenti, ecc. La simulazione delle macro sospette verifica specialmente se la macro è in grado di replicarsi, azione assolutamente atipica per un normale utilizzo. In questo caso vi è un'altissima probabilità di trovarsi in presenza di un virus. La soluzione non è certamente banale: vi sono alcune macro che vengono installate nel normal.dot, il contenitore delle macro disponibili a tutti i documenti. Il comportamento atipico, per queste macro, sarebbe il trasferimento dal normal.dot ad altri documenti; vengono allora simulate due generazioni di esecuzione: qualora una macro si trasferisca nel normal.dot, si verifica che poi da li non si trasferisca altrove, simulandone l'esecuzione nel normal.dot. Queste stesse tecniche possono venire utilizzate per la ricerca degli altri tipi di malware: backdoor, cavalli di Troia, worm. Se nel caso dei cavalli di Troia il compito è relativamente semplice – si tratta di individuare comportamenti distruttivi gratuiti – la difficoltà cresce per backdoor e worm. Le backdoor sono programmi molto difficili da rilevare, poiché i comportamenti classici sono l'ascolto o la trasmissione di dati in rete, in formati o su porte non standard, unito alla capacità di rilevare dati non relativi al funzionamento del programma e alla loro eventuale trasmissione; si tratta quindi di operazioni complesse che vengono eseguite in seguito a particolari condizioni. I worm sono programmi che normalmente non si trovano nel file system, o non stabilmente. Possono anche fungere da veicoli per ulteriori malware ma il loro funzionamento consiste nell'infettare un sistema e riprodursi "on-line"; il worm è infatti un malware che si riproduce utilizzando un canale di comunicazione diretto. Il meccanismo è il seguente: un worm, dopo avere preso il controllo di un sistema, verifica se il canale è ancora aperto. Se lo è, cerca dei possibili interlocutori, rilevandoli da qualche file agenda oppure con una scansione del canale (normalmente una rete) oppure ancora in modo casuale. Il worm

conosce qualche errore nella gestione della sicurezza del canale utilizzato: attraverso questo esso convince i sistemi bersaglio ad eseguirlo, dopo essersi installato in essi. Se il canale non è più aperto, il worm può entrare in uno stato di quiescenza. La rilevazione può avvenire con la scansione durante la quiescenza, oppure dopo aver ospitato l'infezione, rilevando le "scorie" del worm o le sue parti installate nel sistema. Invece, un sistema utile è di verificare i dati che entrano nel sistema dall'esterno attraverso i canali diretti, in tempo reale. I possibili comportamenti sono i classici meccanismi virali, uniti alla capacità di rilevare file agenda, replicarsi, prendere controllo di qualche software usato per gestire il traffico sul canale, automatizzare meccanismi normalmente attivati dall'utente. Vi sono molti problemi per la realizzazione di questi sistemi di controllo: prima di tutto, se il canale presenta un traffico intenso, la velocità di esecuzione diventa realmente critica; poi, attualmente il codice eseguibile in modo automatico e trasmissibile è realizzato per macchine virtuali differenti dalla CPU, come la Java Virtual Machine, controlli Activex, Visual Basic Script, linguaggio degli script di client IRC, ecc. Gli scanner, quindi, per effettuarne una simulazione, debbono comprendere le simulazioni di diverse macchine virtuali. Si tratta di programmi complessi e voluminosi, che richiedono di essere estremamente ottimizzati per aumentarne la velocità di esecuzione.

Nonostante i vantaggi della ricerca euristica, vi sono alcuni limiti importanti. Si consideri un virus che attiva dei comportamenti caratteristici solo in determinate condizioni, come un periodo particolare, una certa ora, determinate caratteristiche del file system, ecc. Un tale comportamento può non attivarsi in fase di simulazione e impedire la rilevazione del virus, problema che non si pone nell'euristica statica. Inoltre, vi possono essere dei meccanismi particolari non implementati dal simulatore, come l'utilizzo di istruzioni dell'unità di elaborazione in virgola mobile ovvero l'utilizzo di risultati di chiamate di sistema modificate. Le possibili soluzioni sono una sinergia tra euristica dinamica e statica, per determinare, ad esempio, aree sospette non raggiunte dalla normale esecuzione del codice nel simulatore, in cui attivare la simulazione appositamente. È, quindi, necessaria una simulazione molto fedele del sistema. Viene anche utilizzato un insieme di tecniche finalizzate a rilevare le condizioni in cui un particolare codice virale si attiva, e a renderle artificialmente vere.

Come si è già accennato, il tempo di esecuzione è un fattore critico; invece, la simulazione del codice unita all'analisi del comportamento può risultare molto lunga. Sono necessari algoritmi molto efficienti, ottimizzati espressamente per la rapidità di esecuzione,

uniti a regole intelligenti che evitino ricerche inutili e che determino in modo preciso quando e dove iniziare e quando terminare la simulazione. È noto che una delle tecniche utilizzate dagli scrittori di virus è di implementare lunghi cicli, al fine di spingere il simulatore a sospendere l'analisi perché si è raggiunto il limite di tempo a disposizione.

La tecnica di ricerca euristica può essere soggetta a molti falsi allarmi se non è tarata in modo corretto; infatti, se è possibile rilevare erroneamente delle signature in un codice corretto, sarà ancora più probabile rilevare un insieme di comportamenti semplici, apparentemente scorretti, in un codice integro. Si tenga sempre presente che il problema della rilevazione dei virus è risolvibile a meno di una certa, ineliminabile, percentuale di errore. La segnalazione di un infezione, di un virus sconosciuto o di una nuova variante di qualche virus, porta comunque ad intraprendere una serie di azioni che possono andare dall'attesa del responso da parte del centro ricerche anti-virus, fino alla cancellazione del file, laddove vi sia un backup velocemente reperibile oppure non vi sia la possibilità di trasmettere il file ad un laboratorio per l'analisi. Ripetere spesso queste operazioni per via di falsi allarmi porta a reazioni facilmente prevedibili da parte degli utenti. D'altra parte, la mancata rilevazione di un nuovo virus può portare ad una grande diffusione dello stesso, unita al rischio di attivazione di payload dannosi. I produttori di anti-virus si trovano, quindi, a dover decidere quale deve essere la percentuale di falsi positivi, che determina di conseguenza anche la percentuale di falsi negativi, cioè di virus non rilevati. I loro clienti raramente possono comprendere i meccanismi per cui questo problema non può essere eliminato; facilmente i clienti, se interrogati sulla soglia di falsi positivi e negativi tollerabile, rispondono che tale soglia è nulla. La selezione della sensibilità della ricerca euristica è quindi un problema significativo; alcuni produttori lo hanno demandato all'utente, introducendo un insieme di livelli di sensibilità selezionabili. È comunque palese che una sensibile ricerca euristica, che offre una notevole sicurezza aggiuntiva, si paga in ragione di una crescita nella probabilità di falsi allarmi.

La possibilità di avere falsi allarmi è presente, in ogni modo, anche nella scansione di signature; questa è la ragione per cui la soluzione adottata dai principali anti-virus consiste in una tecnologia mista di tecniche euristiche e di ricerca di signature, in modo da ridurre la probabilità di errori. Un esempio di come si possa realizzare un sistema avanzato di rilevazione di virus conosciuti e sconosciuti è presentato di seguito.

La scansione di signature specifiche di virus non è che una prima analisi grossolana. Nel caso venga rilevata una signature virale, si può confrontare il codice con una mappa delle varianti conosciute e verificare se qualcuna si adatta al codice rilevato. A questo punto può venire eseguita una verifica di integrità, normalmente un checksum, del codice virale per verificare se esso corrisponde esattamente ad una delle varianti conosciute. Se questo ha successo, ci si trova di fronte ad una presenza del virus; questo però, negli anti-virus più moderni non basta: parte a questo punto una simulazione del codice per verificare se il codice virale rilevato è realmente accessibile oppure è inattivo. Se, invece, la verifica di integrità non rileva una variante conosciuta, si può comunque eseguire una verifica euristica particolarmente vigile, specialmente nell'area in cui si è rilevata la signature. Se si rilevano comportamenti sospetti di essere virali si può innescare una simulazione completa per verificare ancora se il codice è raggiungibile. A questo punto ci sono buone possibilità di trovarsi in presenza di una nuova variante di un virus.

Terminata la scansione di signature senza successo, oppure contemporaneamente a questa, si esegue un'altra scansione alla ricerca di signature specifiche di comportamenti sospetti. Nel caso si rilevino alcuni match ravvicinati, si può iniziare una simulazione del codice in quella posizione. Si può anche evitare la scansione per signature comportamentali e passare al passo successivo, che consiste nell'effettuare la simulazione del codice in particolari aree note come possibili veicoli di virus: le prime istruzioni successive all'entry point, le istruzioni poste in coda al file, le istruzioni poste in corrispondenza di aree che dovrebbero invece contenere dati oppure situate ai bordi o al centro di zone contenenti valori nulli o valori di riempimento. L'analisi consiste nel cercare i meccanismi descritti sopra, quali la riproduzione, la potenzialità dannosa o la capacità di modificarsi. La simulazione non può essere eseguita troppo a lungo, quindi si stabiliscono a priori il numero massimo di passi di simulazione eseguibili senza rilevazioni significative; questi possono essere aumentati se si rilevano sezioni sospette ma ancora non vi è una sufficiente certezza, oppure se si sono rilevate precedentemente, nell'area, delle signature virali, o comportamentali, significative. Nel caso di modifica il codice viene simulato fino a terminare la decifrazione e poi si verifica se la routine decifrata viene anche eseguita. In questo caso, allora, è possibile ripetere ricorsivamente tutto il processo sulla porzione di codice appena scoperto; questo è fattibile poiché si tratterà comunque di una breve porzione. Contemporaneamente alla ricerca di meccanismi virali si può effettuare una ricerca di comportamenti opposti, cioè che facciano diminuire la probabilità di presenza di un virus nella porzione di codice simulato. Al termine può essere effettuato un confronto tra il

"punteggio" positivo e negativo ottenuto dall'analisi; se si è raggiunta una certa soglia verrà segnalato il sospetto di un virus sconosciuto. Nel caso di ricerca di variante sconosciuta, attivato dopo avere rilevato una signature, si può diminuire la soglia della valutazione euristica. La soglia può anche essere resa nulla se il codice esaminato, anche se non appartiene a nessuna variante conosciuta, "assomiglia" abbastanza ad una delle mappe delle varianti memorizzate. Questo, dopo aver verificato se il codice viene effettivamente eseguito, sempre oppure sotto certe condizioni. Questo è facilmente rilevabile: in presenza di un salto incondizionato o di un ritorno che escluda la porzione sospetta, non dovrebbero esserci dubbi; invece se i salti sono condizionali si può esaminare sia la possibilità che il flusso di esecuzione devi sia che non devi. Infine, la tecnica presentata, con lievi variazioni, viene effettuata sul settore di avvio e sul master boot sector delle unità collegate, ed in memoria. La tecnica presentata viene riassunta nel diagramma seguente:

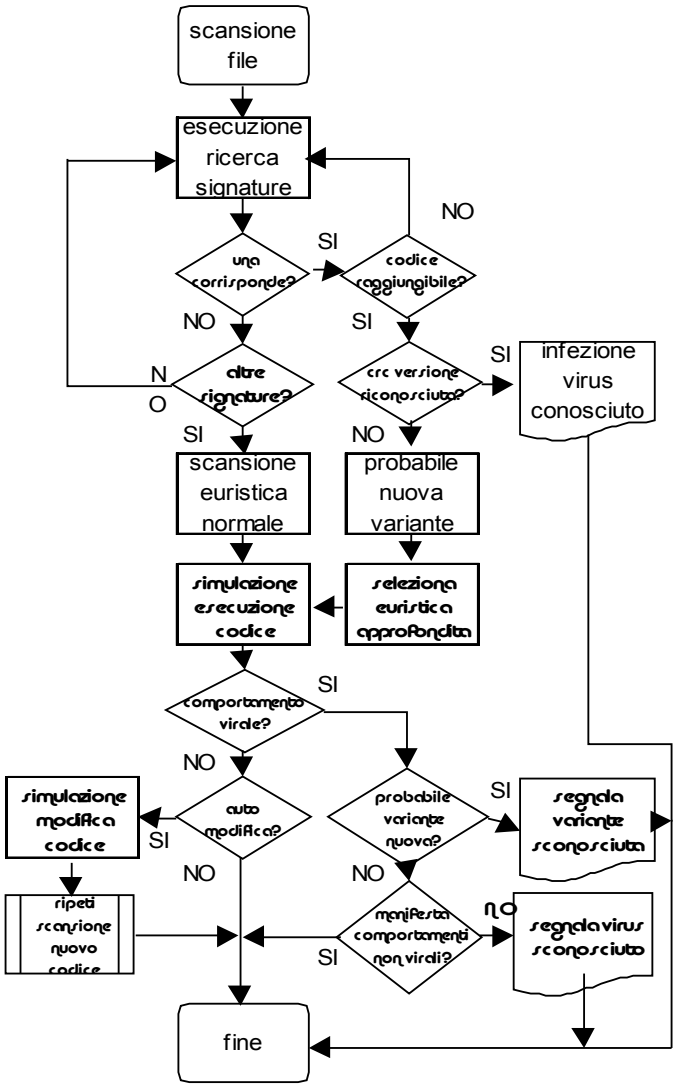

Figura 4

Il database delle azioni sospette e delle regole per l'osservazione non è statico; questi dati sono soggetti ad una continua verifica e relativo aggiornamento per includere le nuove tecniche virali, anch'esse soggette ad una continua evoluzione.

Una delle possibili innovazioni è di utilizzare le reti neurali per il riconoscimento del codice virale. Le reti neurali sono infatti conosciute per la loro efficacia nel campo della classificazione. Una delle implementazioni note è quella sviluppata dal centro ricerche dell'IBM [36], consistente in una rete neurale adatta al riconoscimento dei boot virus. L'insieme di dati utilizzato per istruire la rete consisteva in alcune centinaia di virus e di un centinaio di possibili settori di boot validi. L'idea principale è stata quella di individuare un insieme di caratteristiche tipiche dei boot virus sotto forma di trigrammi, dopo averne selezionati automaticamente un grande numero ed aver conservato solo quelli che non apparivano all'interno dei possibili settori validi. L'insieme delle rilevazioni della presenza o assenza di un particolare trigramma viene fornito in input alla rete per la classificazione. Nel riconoscimento dei settori di boot, viene posta maggiore enfasi sulla riduzione dei falsi positivi; difatti una segnalazione errata di questo tipo verrebbe ripetuta in migliaia di casi, essendo i settori di boot validi relativamente pochi. I risultati segnalati dai ricercatori dell'IBM riferiscono che, al momento della redazione della relativa pubblicazione, il prodotto aveva rilevato il 75% dei nuovi boot virus e aveva generato solo due casi di falso positivo.

4.1.3 – Verifica di integrità

Le tecniche di verifica di integrità consistono nell'esaminare tutti i file presenti nelle memorie di massa del sistema da proteggere e inserire in un apposito database le informazioni di validazione relative ad ogni file esaminato. Con informazioni di validazione si intende un dato calcolato in base al contenuto del file ad esso associato, che deve risultare differente se il file su cui viene calcolato varia. Periodicamente la scansione si ripete, i nuovi file vengono aggiunti al database ed i file già presenti vengono verificati, ricalcolandone le informazioni di validazione e verificando se corrispondono a quanto contenuto nel database. Se tali dati non corrispondono è un segno certo che il file relativo è stato modificato; quindi esso può essere stato infettato da un virus. La tecnica normalmente comprende la verifica

del settore di avvio delle unità a disco fisso, in modo analogo. Si cercherà ora di esaminare nel dettaglio i vari aspetti di questa tecnica.

Le informazioni di validazione debbono essere capaci di rilevare se un insieme di dati o file F, esaminato precedentemente nell'istante T, durante una successiva verifica eseguita nell'istante T', con T'>T, risulti essere stato alterato. Empiricamente, considerando il nostro insieme una stringa di n bit, possiamo definire che è stato alterato se: il numero di bit della stringa è aumentato o diminuito, oppure uno o più bit sono stati modificati. Una maniera di verificare questo è di effettuare una copia del file nel database; questo però richiede di raddoppiare l'occupazione in spazio dei dati. Normalmente si calcola, per ogni file F, un breve valore H(F) di lunghezza fissa (tipicamente da 2 a 32 byte); dove H è una funzione tale che, definita con F' una qualunque modifica di F, le probabilità che H(F) = H(F') siano estremamente basse[52]. Queste funzioni vengono spesso definite checksum, perché inizialmente la funzione utilizzata era proprio la somma di tutti i byte del file, oppure codici di controllo. Le moderne tecniche sono molto più complesse; vi è anche una classe di codici di controllo basati su una chiave segreta. Anche se quest'ultima classe può apparire non significativa, nel campo anti-virus ha un'importanza non indifferente; la stessa importanza ha la scelta di algoritmi di generazione dei codici che non permettano facilmente di essere forzati. Un esempio è in grado di chiarire la ragione di queste precauzioni: si ponga di scegliere un codice di controllo basato sulla verifica della parità; se una tale tecnica venisse adottata da numerosi prodotti anti-virus, gli autori di virus potrebbero con grande facilità aggiungere al codice del virus un byte, calcolato in fase di infezione, capace di mantenere la stessa parità del file originale. Il problema della forzatura del database non viene considerato indifferente dai ricercatori del campo; pubblicazioni specializzate nell'argomento consigliano, in ambienti ove sia richiesta una certa soglia di sicurezza, l'utilizzo di codici di controllo basati su chiave segreta; consigliano inoltre la conservazione del database in sede separata [52]. Attraverso l'uso di una chiave segreta diversa per ogni esemplare del software di controllo (generata in modo random, per esempio), e di un algoritmo di generazione del codice robusto, le possibilità di forzatura dei codici di controllo sono molto basse. La conservazione in separata sede è consigliabile, poiché si sono già rilevati molti casi di virus in grado di cercare e cancellare i database di codici di controllo generati da diversi noti programmi anti-virus; molti software obsoleti, in caso di mancata rilevazione dei database, si consideravano in fase di installazione e rigeneravano i database in modo autonomo e senza informarne l'utente. In questo modo l'infezione era del

tutto inosservata. L'informazione della perdita dei database, seppure meno grave del caso appena presentato, fornisce solo l'informazione della probabile presenza di un virus attivo, senza specificare in quale file esso si possa trovare: un'informazione quasi totalmente inutile e capace di scatenare il panico, specie ove non vi sia un backup completo dei dati, effettuato poco prima della rilevazione. Il programma di generazione deve essere anche protetto in modo da impedire che sia rimpiazzato o modificato da un programma ostile; l'ideale sarebbe la sua conservazione, assieme al database di codici di controllo, in uno o più dischetti a parte e protetti dalla scrittura. Un possibile compromesso, nel caso che quest'ultima precauzione sia impossibile, è di permettere all'utente di scegliere la posizione e il nome del database, onde evitare una facile rilevazione, eventualmente cifrandolo. È consigliabile mantenere assieme i dati relativi ai checksum delle diverse directory, oltre che per le ragioni di sicurezza menzionate, per evitare una crescita dell'occupazione dello spazio. È altresì raccomandabile che il programma di verifica esegua un'auto-verifica di integrità prima dell'esecuzione, se presente sull'hard disk.

Una tecnica anti-virus basata su verifica di integrità è praticamente esente da falsi negativi, se vengono rispettate le misure precauzionali descritte; non si può dire altrettanto dei falsi positivi. In un sistema normale molti file vengono modificati quotidianamente. Sebbene il database comprenda solo i file che possono essere infettati da un virus, come eseguibili e documenti capaci di includere macro, anche questi sono soggetti a modifiche con probabilità maggiore di un'infezione virale. Questo problema può essere risolto includendo informazioni aggiuntive; nel caso dei documenti, il database può includere solo quelli contenenti macro, informando l'utente di ogni nuova rilevazione; in questo modo l'utente può segnalare, al programma di verifica, l'inserimento o la modifica di macro nei documenti. La stessa cosa può essere realizzata per i file eseguibili, che possono essere stati appena aggiornati ad una versione successiva oppure modificati o generati da un compilatore. Per questa ragione, in fase di verifica, il programma di controllo informa l'utente di ogni modifica rilevata ai file inclusi nel database e di ogni file rilevato per la prima volta, come dei file cancellati; è compito dell'utente decidere se le rilevazioni si debbono considerare sospette o normali, oppure può inserire a priori le modifiche da considerarsi normali. Da queste caratteristiche si può intuire che, nonostante le scarse possibilità di falsi negativi e la robustezza della tecnica, qualora vengano seguite le norme presentate, vi sono molti punti a sfavore:

- L’utente deve tenere traccia delle modifiche effettuate ai file del sistema, deve comunque avere l’abilità di distinguere una modifica sospetta da una normale.
- Questa tecnica è in grado di rilevare le modifiche operate sul sistema, ma non può rilevare se un file introdotto ex novo nel sistema sia o meno infetto da un virus.
- La rilevazione della presenza di un virus avviene dopo che si è verificata almeno un'infezione, fatto non auspicabile, poiché il virus può eseguire già da subito payload dannosi.
- In caso di attivazione di virus con tecniche stealth, la tecnica non è più efficace.
- Vi sono virus, denominati slow infectors, che infettano file eseguibili solo quando essi sono aperti in scrittura; in questo modo risulta normale all’utente rilevarne la modifica.
- La perdita dei dati relativi al checksum rende impossibile effettuare ulteriori rilevazioni.
- Modifiche sulla struttura delle directory delle unità monitorate, ed in generale ogni azione che richieda la modifica di molti file, deve essere eseguita solo dopo una verifica completa, poiché dopo è praticamente indispensabile ricostituire una parte del database, evitando così i moltissimi messaggi che verrebbero generati.
- Virus, come Brain, che infettino soltanto i dischetti, sono praticamente non rilevabili.
- Esiste la possibilità, remota ma non nulla, che il programma rilevi una modifica dovuta a malfunzionamento dell’unità.

Per queste ragioni, la verifica di integrità non è mai stata utilizzata da sola e nemmeno come tecnica primaria di controllo. È stata piuttosto utilizzata come tecnica complementare in alcuni prodotti. La tendenza attuale è di basarsi su altre tecniche complementari come la verifica comportamentale (behavior blocker o monitor), affidata a processi residenti.

4.1.4 – Cleaning

Il cleaning o disinfezione, consiste nel cancellare il codice virale da un file infetto, ripristinandone la funzionalità. Questo può avvenire solo se la variante del virus viene rilevata in modo preciso; infatti è possibile rimuovere un virus, senza danneggiare il file infettato, solo se ne si conosce esattamente la morfologia, in modo da cancellare tutte le parti virali ed utilizzare i dati contenuti in esse, come l’indirizzo di inizio del programma

originale, per ripristinare il funzionamento del programma disinfettato. Il programma che esegue la rimozione deve quindi contenere le informazioni sulla morfologia del virus, sulla posizione dei dati relativi all'interno di esso e sul genere e numero di modifiche che esso apporta al file infetto. Può anche verificarsi che la rimozione non abbia buon fine, producendo un eseguibile non più funzionante; per questo viene spesso consigliato di effettuare una copia del file prima della rimozione. Nel caso questa non riesca, si può inoltrare il file ai centri di ricerca della azienda anti-virus produttrice del software. Vi sono poi alcuni casi di virus che non possono essere rimossi con successo: i virus overwrite, per esempio, sovrascrivono il file originale invece di aggiungere il proprio codice in testa o in coda al file.

Esistono delle tecniche, note come rimozione o cleaning euristico, che consistono nel tentare di ripristinare la funzionalità di un file infettato da un virus sconosciuto. Una possibile tecnica consiste nell'affiancare al proprio sistema anti-virus un verificatore di integrità, che includa nel proprio database, oltre al checksum, altri dati utili al ripristino dei file infetti. I dati possono consistere in una copia dei primi byte del file, che contengono l'header e le prime istruzioni, assieme alla dimensione originale del file. Nel caso di infezione sconosciuta, il disinfettore può ripristinare la porzione iniziale del file e troncare questo alla lunghezza originale; poi eseguire una verifica di integrità del risultato per controllare se si è ripristinato il file all'aspetto originale. In caso di fallimento possono essere eseguite regole di disinfezione alternative, come ripristinare l'entry point del file e rilevare se vi sono porzioni di codice non raggiungibili dal flusso normale di esecuzione; in questo modo si dovrebbe poter rilevare virus presenti in aree interne al file, come zone di dati. Anche queste tecniche molto evolute sono soggette a funzionare solo in una parte dei casi reali: vi possono essere stati degli errori nella programmazione del virus o nel modo di effettuare l'infezione che rendano impossibile ritornare alla condizione originale. La tecnica presentata potrebbe essere valida anche contro i virus overwrite, ammesso che la sovrascrittura avvenga all'inizio del file e la porzione di codice memorizzato nel database di ripristino sia sufficientemente lungo. Si tratta di tecniche in fase di studio e ancora raramente implementate in prodotti commerciali.

4.1.5 – Monitor

Si tratta di una categoria di programmi che vengono installati in memoria e vi restano residenti; essi eseguono diversi controlli in background al fine di rilevare infezioni virali. La tecnica utilizzata è quella dell'analisi comportamentale, da cui il nome, spesso usato, di "behavior blocker". Il processo analizza tutte le attività svolte dai programmi che vengono eseguiti e rileva quelle che sono sospette; la reazione classica è di bloccare l'azione e di informare l'utente del nome del programma che ha intrapreso l'azione, richiedendo quale contromisura intraprendere. Le contromisure tipiche sono: interruzione del processo sospetto, blocco della sola azione sospetta, cancellazione/rinomina del programma origine dell'azione sospetta, oppure proseguimento senza intraprendere nessuna contromisura.

Normalmente le azioni sospette sono le seguenti:

- Modifica di file eseguibili
- Accesso diretto al disco
- Accesso al settore di avvio
- Tentativo di formattare un disco rigido
- Ricerca di file eseguibili in un'unità

Esistono altri tipi di controllo che possono essere intrapresi, il più valido e comune è di effettuare una scansione al volo di ogni file che viene aperto od eseguito; in questo caso il monitor può rilevare se il file che sta per essere eseguito o copiato è infetto da un virus, conosciuto o sconosciuto. Uno dei requisiti degli algoritmi relativi a questo genere di controllo è certamente la rapidità, è infatti necessario analizzare moltissimi dati senza rallentare il sistema.

Questo genere di programmi può essere di grande utilità, specialmente quelli comprendenti anche la scansione al volo dei file eseguibili. Moltissimi recenti software anti-virus comprendono questo tipo di programma, affiancato normalmente da uno scanner.

4.1.6 – Digital immune systems

Le analisi della tendenza del fenomeno rivelano principalmente un aumento della crescita dei virus, con un andamento non lineare ma a sbalzi e con una continua evoluzione delle tecnologie virali; è prevedibile che i futuri virus sfrutteranno l'aumentata connessione dei computer, dovuta allo sviluppo di Internet, anche con tecniche di diffusione di massa attraverso l'infezione della posta elettronica e di altri meccanismi automatici client-server. Per questo motivo i worm avranno anche un notevole sviluppo che li potrebbe portare a prevalere numericamente sugli altri tipi di malware; inoltre, si tratta di un metodo di infezione rapida e particolare.

Per contrastare queste tendenze, soprattutto in ambito corporativo, alcune aziende di sviluppo di software hanno studiato ed intrapreso la strada dei sistemi digitali immuni. Si tratta di sistemi anti-virus basati sull'analisi residente, capaci di rilevare la presenza di virus dal differente comportamento di un sistema, in una maniera più sofisticata dei behavior blocker, che sono rivolti prevalentemente alla ricerca di virus di file eseguibili; si tratta di analizzatori dei diversi comportamenti standard degli utenti del sistema, capaci di rilevare una possibile sequenza di azioni rivelatrice di un'infezione virale di qualunque tipo. Per esempio, l'utilizzo di un particolare software di analisi dei file eseguibili, come un debugger, normale se avviene in un orario di ufficio, diventa sospetto se avviene invece ad orari come le cinque o le sei di mattina; oppure, se è normale inviare posta elettronica a diversi indirizzi con diversi oggetti, è invece sospetto l'invio di grandi quantità di posta, ad indirizzi differenti, aventi tutti lo stesso oggetto oppure il medesimo file allegato. La rilevazione di questi comportamenti fa scattare altri sistemi di analisi più approfonditi, come gli scanner. Se viene rilevato un virus o un worm sconosciuto, il sistema deve essere in grado di inviare un esemplare infetto ad una centrale di analisi, dove sia sviluppato in modo automatico un rimedio. Questo può consistere in un'estrazione di una signature, eventualmente assieme alla produzione di dati sul virus che ne permettano la rimozione. Si noti che, per esempio, l'algoritmo analizzato nel paragrafo 4.1.1.1, è stato sviluppato dall'IBM proprio allo scopo di essere utilizzato in un sistema immune. La generazione di dati per la rimozione può essere effettuata usando una sorta di rimozione euristica, che cerchi di rilevare, dall'analisi del comportamento del virus, le sue caratteristiche e la sua morfologia; per esempio rilevando le operazioni eseguite al termine della routine virale, che normalmente hanno lo scopo di trasferire il controllo al programma infettato e, quindi, utilizzano i dati archiviati per l'esecuzione del programma originale.

Considerando i clienti di un fornitore di un sistema immune collegati tra di essi per mezzo di Internet, i dati generati in questo modo possono essere diffusi dallo stesso sistema che ha rilevato il problema, secondo uno schema simile a quello degli anticorpi, a tutti i sistemi adiacenti; analogamente nel caso di rilevazione all'interno di una rete aziendale. In questo modo si può notare che l'infezione viene contrastata da una diffusione del rimedio e dei dati di rilevazione, che si può considerare come una contro-infezione; così il rimedio percorre la stessa strada probabilmente percorsa dall'infezione. Un simile meccanismo ha due vantaggi: la capacità di contrastare la rapida crescita del numero dei virus e dei worm, automatizzandone la rilevazione e l'analisi, richiedendo l'intervento umano solo in qualche sporadico caso; inoltre, l'update dei dati necessari a contrastare l'infezione viene automatizzato ed effettuato in tempo reale, utilizzando gli stessi nodi per la diffusione degli aggiornamenti, secondo una sequenza analoga a quello dell'infezione ma certamente più rapida. Si tratta di un problema ancora in fase di studio; per esempio, per quanto riguarda lo studio di una topologia e del sistema di comunicazione di eventuali server di analisi decentrati, al fine di impedire un sovraccarico, che si può realisticamente verificare nel caso di un'infezione rapida, quale quella generata dai worm [47][37].

4.2 – Fonti di ricerca di nuovi virus

La gestione, la raccolta e lo scambio di codice virale presenta alcuni problemi, alcuni dei quali di natura etica. Coloro che non appartengono alla categoria dei ricercatori anti-virus accreditati dalla comunità scientifica o da un'azienda, spesso non si pongono problemi di sorta e utilizzano come fonte di materiale per esperimenti i siti e le newsgroup di "virus exchange", ovvero di scambio di virus e di collezionisti ed autori di virus. I ricercatori ufficiali non utilizzano queste fonti per il loro normale lavoro di ricerca; le motivazioni di questa scelta sono diverse: ragioni di natura etica, viene infatti considerata estremamente negativa la pubblicazione con qualsivoglia mezzo di codice sorgente virale. Per quanto riguarda poi la pubblicazione o distribuzione di codice eseguibile virale, essa è considerata un reato in molte nazioni, tra cui anche l'Italia, come si è già visto. Poi ragioni di natura pratica: i virus reperibili nelle BBS virali che siano anche rilevanti sono un sottoinsieme; spesso in questi siti si trovano alcune collezioni di virus prodotti da qualche autore ma non rilevati "In The Wild", cioè nella lista ufficiale dei virus effettivamente diffusi (vedere paragrafo 5.4), i famosi "Zoo virus"; altrimenti, si trovano virus con errori di

programmazione che li rendono non funzionanti o virus che non corrispondono al nome dichiarato. La fonte primaria, attualmente, è sicuramente quella degli utenti dei diversi prodotti anti-virus, i quali trasmettono alle case produttrici, per un'analisi approfondita, quei file riconosciuti come “sospetti” di infezione dalle ricerche euristiche dei diversi prodotti. Quest’ultimo canale di ricezione di virus è sicuramente valido per virus di recente concezione; resta però evidente il problema di iniziare una simile attività: non esistono collezioni pubbliche e non è realistico pensare di ricevere un esemplare di ogni virus esistente dai nuovi clienti, primo perché questi opterebbero sicuramente per un'azienda produttrice di un anti-virus già funzionante, secondo perché i virus hanno raggiunto un numero vicino ai 50000. Inoltre, l’attività di un ricercatore è estremamente complessa e la gestione della collezione ed il relativo studio degli esemplari richiedono tempi molto lunghi; per questo un nuovo ricercatore dovrebbe trovare un collega disposto a mettergli a disposizione almeno una parte della sua collezione; questo potrebbe essere anche sufficiente, infatti è noto che i virus non presenti nella lista “in the Wild” sono estremamente rari, quindi un prodotto in grado di rilevare con sicurezza almeno i virus più recenti, i virus “in the Wild” ed un certo numero di virus datati è già teoricamente presentabile sul mercato. Un’altra fonte valida è quella dello scambio tra i ricercatori stessi, che avviene attraverso invio di esemplari protetti da crittografia robusta; si deve però considerare che le collezioni di virus vengono considerate strettamente riservate, oltre che per ragioni di sicurezza, per ragioni di segreto commerciale. La concorrenza tra prodotti anti-virus, in questo caso intesa come la capacità di riconoscere un numero di virus superiore ai prodotti rivali, è fortissima. Vi sono poi le collezioni delle organizzazioni non associate ad un anti-virus, pubbliche e private. Fino a poco tempo fa, il CIAC dei Lawrence Livermore National Laboratories, aveva un database di descrizioni di virus onnicomprensivo, attualmente non più aggiornato. La maggior parte di queste organizzazioni pubbliche hanno rinunciato, negli ultimi tempi, a mantenere un database aggiornato, per le crescenti difficoltà relative al numero di virus esistenti e al tempo necessario per lo studio di questi; spesso consigliano di riferirsi ai database di descrizioni di industrie anti-virus. Vi sono ancora buone collezioni presso le organizzazioni che rilasciano certificazioni; però questi database non necessitano di essere onnicomprensivi, essendo solo una sorta di test della bontà di un anti-virus. Normalmente presso queste organizzazioni si trova un ricercatore che gestisce la collezione, fornita in parte da altri ricercatori, per esempio quelli affiliati alla organizzazione WildList, che dovrebbero essere meno gelosi della propria collezione dei ricercatori delle aziende anti-virus commerciali. È comunque

comprensibile che i ricercatori non distribuiscano esemplari di virus a chiunque ne faccia richiesta; quindi la nascita di nuove organizzazioni o aziende in grado di creare e mantenere una collezione di virus valida è, con il passare del tempo, sempre meno probabile.

4.3 – Valutazione dei prodotti anti-virus

In questo paragrafo si fornisce una descrizione critica delle attuali tecniche di valutazione.

La valutazione dei prodotti per mezzo di test di rilevazione può essere eseguita, a patto di avere una collezione di virus sufficiente. Questo problema non è di facile risoluzione: i virus presenti nei siti di virus exchange vengono considerati dai principali ricercatori come "zoo virus", cioè virus da collezione, privi di significatività; inoltre, l'utilizzo di tali materiali viene considerato eticamente scorretto. Convincere un ricercatore a farsi inviare una collezione di virus, ben anche incompleta, è un'impresa irrealizzabile: sia dal punto di vista etico che da quello commerciale, tali dati vengono considerati riservatissimi; tra l'altro, da un punto di vista commerciale, una collezione può essere usata per generare un database di signature, elemento critico della tecnologia degli scanner. La produzione di nuovi virus allo scopo di testare i prodotti, poi, viene considerata una grave scorrettezza da un punto di vista etico.

La soluzione ufficiale, rivolta al pubblico, è l'utilizzo del file di test standard, prodotto dall'EICAR. Questo istituto, universalmente accreditato e riconosciuto, ha prodotto un file di 70 byte, denominato "Eicar standard anti-virus test file". Si tratta di un file eseguibile in formato COM, che in fase di esecuzione produce come output la stringa "EICAR-STANDARD-ANTI-VIRUS-TEST-FILE!". Per convenzione, anche se il file non contiene assolutamente meccanismi virali, eccetto la capacità di auto-modificarsi, gli scanner lo rilevano come un virus e lo segnalano come un file infetto dal virus "Eicar_Test_File" o un nome analogo. Il contenuto è un segmento di codice il cui scopo è di visualizzare la stringa descritta e poi terminare l'esecuzione in modo normale. Il codice risulta piuttosto complesso, anche perché è stato studiato in modo che i valori numerici delle istruzioni che lo compongono siano tutti riconducibili a caratteri visualizzabili. In questo

modo, è possibile produrre il file con un semplice editor di testo, nel quale si digitino i seguenti caratteri:

```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTI-VIRUS-TEST-FILE!$H+H*
```

È poi sufficiente salvare il file con l'estensione ".COM". [80]

La codifica in assembler è la seguente:

```
0100 58            POP      AX
0101 35 4F 21      XOR      AX,214F
0104 50            PUSH     AX
0105 25 40 41      AND      AX,4140
0108 50            PUSH     AX
0109 5B            POP      BX
010A 34 5C         XOR      AL,5C
010C 50            PUSH     AX
010D 5A            POP      DX
010E 58            POP      AX
010F 35 34 28      XOR      AX,2834
0112 50            PUSH     AX
0113 5E            POP      SI
0114 29 37         SUB      [BX],SI
0116 43            INC      BX
0117 43            INC      BX
0118 29 37         SUB      [BX],SI
011A 7D 24         JGE      0140

011C 45 49 43 41
0120 52 2D 53 54
0124 41 4E 44 41
0128 52 44 2D 41
012C 4E 54 49 56
0130 49 52 55 53
0134 2D 54 45 53
0138 54 2D 46 49
013C 4C 45 21 24   db 'EICAR-STANDARD-ANTI-VIRUS-TEST-FILE!$'

0140 48 2B 48 2A   db 48h, 2Bh, 48h, 2Ah
```

Il programma modifica autonomamente gli ultimi quattro byte, trasformando il codice nel seguente:

```
0140 CD 20        INT 20h
0142 CD 21        INT 21h
```

che esegue la chiamata di sistema per visualizzare la stringa, INT 20h (i registri sono già stati configurati in precedenza), ed infine la chiamata di sistema per terminare il programma e restituire il controllo al sistema operativo, INT 21h. Risulta evidente che il test Eicar è

utile solo per rilevare l'effettivo funzionamento di un prodotto anti-virus, ma intorno alla sua validità non fornisce alcuna indicazione.

Vi sono alcune organizzazioni che producono test comparativi dei prodotti anti-virus, altre invece forniscono una certificazione di conformità ad alcuni standard determinati e pubblicati dalle organizzazioni stesse (Vedere paragrafo 5.4).

Si può affermare che i risultati di una suite di test, per essere significativi, non possono limitarsi alla percentuale di virus rilevati durante la scansione di una vasta collezione di file infetti, come invece spesso avviene. È necessario innanzitutto utilizzare una collezione omogenea, con tutti gli esemplari compresi nella lista "in the Wild" e con una percentuale opportuna di virus dei differenti tipi. È necessario, poi, che la collezione comprenda insiemi di più file infetti per ogni esemplare di virus polimorfico, poiché ad ogni riproduzione questo genere di virus cambia aspetto. Si evidenzia, poi, che un test che si riconduca alla sola percentuale di virus rilevati nella scansione di una collezione, seppure si tratti di una collezione estremamente valida, non prende in esame altri importanti aspetti: la capacità di rilevare virus sconosciuti e la percentuale di falsi positivi. Si noti poi che, come già evidenziato, non è estremamente significativa la rilevazione di virus "zoo" oppure non più rilevati da anni. Le stesse aziende anti-virus effettuano dei test interni sulla percentuale di falsi positivi, utilizzando ampi database di programmi commerciali o freeware. [41] [44] [39]

Si segnala, al riguardo, un'importante ammonizione di diversi ricercatori, intorno all'esecuzione di test fai-da-te. Questi test, oltre che mancare di significatività per via di una collezione incompleta o perché privi dei requisiti evidenziati, sono pericolosi: infatti sono noti diversi casi, in letteratura, di varianti di virus diffusi in seguito a test effettuati senza le dovute precauzioni. In particolare, si ammoniscono le imprese sui rischi di contagio possibili in seguito all'esecuzione di test o esperimenti su macchine aziendali, effettuati da personale non specializzato. [39]

5 - Materiale on-line pro e contro i virus

In questo capitolo vengono descritte le risorse disponibili sulla rete Internet, riguardo ai prodotti anti-virus e ai siti di organizzazioni che si occupano di studiare e fronteggiare le situazioni di emergenza dei computer prodotti da malfunzioni e "attacchi" di diversi tipi. Si esaminano inoltre le risorse parallele con cui i collezionisti e gli autori sono in grado di scambiarsi il codice virale e di reperirlo.

5.1 - Le newsgroup

Con newsgroup intendiamo un particolare sistema di scambio di messaggi via Internet tra persone interessate allo stesso argomento. Quando ci si collega ad un server di newsgroup sono presenti molte directory contenenti i messaggi, suddivise per argomenti. Gli utenti possono consultare i messaggi di una directory, rispondervi ed eventualmente inviarne di nuovi. I vari server sparsi nel mondo, periodicamente si scambiano tra loro i nuovi messaggi, in modo che ognuno abbia in memoria i messaggi inviati agli altri. Vi sono molti newsgroup dedicati al problema dei virus, sia da parte di chi combatte o studia gli effetti dei virus, sia da parte di chi colleziona e studia i virus per una forma di pura curiosità. Quest'ultima categoria non è ben vista dalla comunità scientifica internazionale, perché viene considerata una fonte di problemi. Infatti, i ricercatori nel campo anti-virus evitano in modo scrupoloso di pubblicare le sorgenti di virus in loro possesso, perché si pensa incentivi la comparsa di nuove varianti di virus esistenti, il ché, come si è visto nel paragrafo 4.1.1, costituisce un notevole problema.

I newsgroup principali sull'argomento sono in lingua inglese, si tratta di comp.virus e alt.comp.virus. Questi newsgroup si differenziano tra loro per un motivo essenziale: comp.virus si avvale di un moderatore, situato presso il MIT, che riceve tutti i messaggi che vengono inviati al newsgroup e seleziona quelli che a suo giudizio sono inerenti al tema e pubblicabili. In effetti, il moderatore di comp.virus è rigoroso, tanto che nel newsgroup, spesso, i messaggi presenti non superano i tre-quattro, sempre uguali per mesi, contenenti liste di domande e risposte (FAQ) inerenti il campo anti-virus. Alt.comp.virus, invece, è un newsgroup libero, in cui chiunque può inviare qualsiasi cosa e la moderazione è lasciata alla

netiquette del singolo; gli utenti sono i più diversi e si scambiano attivamente informazioni sulla presenza di nuovi virus e sulle tecniche per evitare o eliminare l'infezione, comprese valutazioni e suggerimenti sui programmi anti-virus. In lingua italiana esistono diversi newsgroup interessanti. Il principale è it.comp.sicurezza.virus (non moderato), paragonabile ad alt.comp.virus per fascia di utenza e argomenti. Esiste poi it.comp.sicurezza.cert-it, newsgroup moderato dal CERT - Computer Emergency Response Team italiano, con sede presso il dipartimento di scienze dell'Informazione dell'università di Milano, il cui livello è prevalentemente tecnico e riguarda non solo il problema dei virus ma ogni problema di sicurezza informatica. Per quanto riguarda i newsgroup dei collezionisti-programmatori di virus in lingua inglese, ne esistono diversi, di differenti livelli: da quelli che si considerano "pro-virus" a quelli che trattano problemi di programmazione di virus, fino ad alcuni che gestiscono scambi di codice virale sorgente tra utenti. Da poco tempo, grazie all'introduzione in alcuni server italiani della classe di newsgroup "free.it.*" (che non vengono creati per votazione, come gli altri, ma per mezzo della richiesta di un singolo), esiste un newsgroup in lingua italiana per lo scambio e la trattazione di sorgenti di virus, peraltro non ammesso da molti server.

5.2 - Siti dei programmatori e collezionisti di virus.

Del problema legato alla distribuzione dei virus, si è già accennato nel capitolo 2. Come si può rilevare da alcune pubblicazioni [45], la reperibilità di codice sorgente virale viene considerata estremamente dannosa dalla comunità scientifica anti-virus. In realtà sono presenti moltissimi siti, per la maggior parte dislocati in nazioni con legislazione non estremamente rigida nel campo informatico oppure dove il diritto alla libertà di espressione supera le restrizioni legali sulla distribuzione di software dannoso. Un esempio è il sito ucraino, dal nome che, tradotto in italiano, suona come "Paradiso dei virus". L'indirizzo non viene citato per ragioni di sicurezza, si dà solo una descrizione del sito e del suo contenuto. In figura 5 si può esaminare la home page del sito: in alto a sinistra vi è la citazione dell'articolo 19 della dichiarazione universale dei diritti dell'uomo, sulla ibertà di espressione, riportata dall'autore del sito per spiegare le ragioni della sua esistenza.

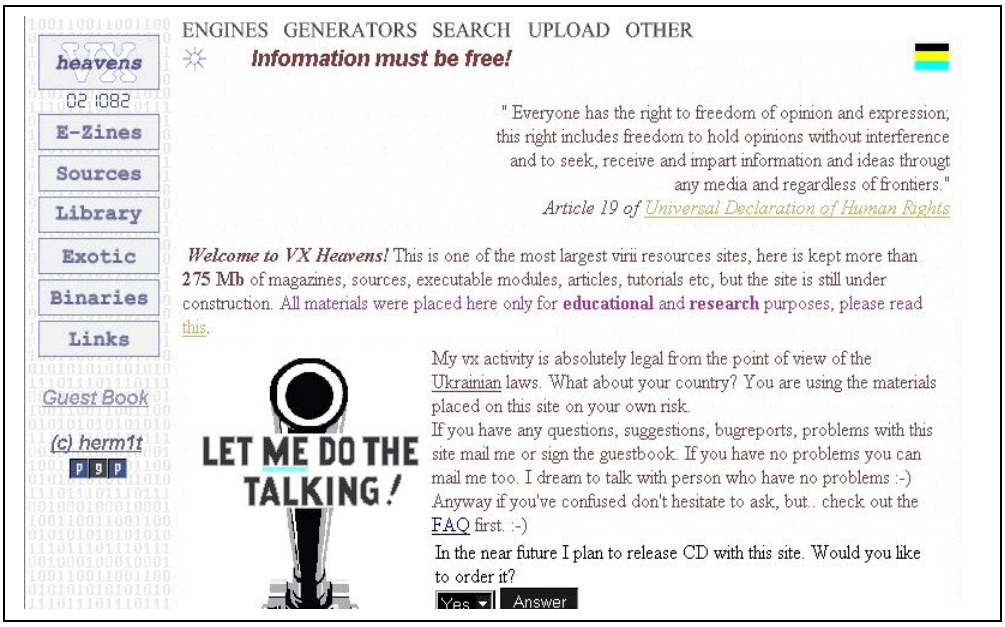

Figura 5

Poco più sotto si può notare una postilla bene evidenziata, spesso presente nei siti di questo tipo, in cui si afferma che il contenuto del sito è riservato solo all'istruzione e alla ricerca; sempre nel tema, si nota una immagine, probabile volantino, sulla libertà di parola. Il sito in questione è ricco di materiale inerente i virus, in particolare in esso vi è una collezione praticamente completa di motori polimorfici e di crittografia. La collezione è aggiornata e presenta alcuni esemplari recentissimi, assieme ai primissimi esempi, come il TpE. L'autore, inoltre, indaga sull'interesse dei visitatori ad acquistare un CD contenente la banca dati del sito, di prossima realizzazione. Sono presenti alcuni link ipertestuali ad una sezione FAQ e ad un documento di scarico di responsabilità. Nei frame attorno alla pagina evidenziata si vedono i link a diverse sezioni: motori polimorfici, generatori di virus, ricerca nel sito, sezione upload, riviste elettroniche o e-zines, file sorgenti, libreria di documenti, file eseguibili, esemplari esotici, raccolta di link. La sezione upload permette di inviare dei file all'autore del sito, il quale, si suppone, selezionerà i più interessanti e aggiornerà così la sua raccolta. Il sito contiene anche una vasta documentazione tecnica, composta di pubblicazioni scientifiche note disponibili on-line, assieme ad alcune pubblicazioni "tecniche" degli autori di virus, riguardo alle tecniche di programmazione, sistemi per eludere gli antivirus, ecc.

Il secondo sito che si esamina è un sito dal nome famoso per la storia degli autori di virus, il cui nome suona in italiano come "scambi di virus". Il sito presenta un insieme di

risorse standard, comprendenti riviste elettroniche, link vari, sorgenti di virus (si afferma alcune migliaia), programmi utility. La particolarità di questo sito è di essere dedicato allo scambio di codice sorgente di virus; infatti, è possibile contattare l'autore ed inviargli file, che verranno poi usati per aggiornare l'archivio, ora diviso per ordine alfabetico e comprendente circa 5 megabyte di materiale.

L'altra interessante particolarità è lo sforzo che si rileva per mantenersi all'interno della legalità, frutto di probabili proteste e comunicazioni da parte di autorità; il messaggio di apertura invita a verificare le leggi del proprio stato e a non connettersi, nel caso queste non lo permettano. Il testo del messaggio di benvenuto recita "Questo sito contiene informazioni e codice sorgente relativo alla programmazione di virus. L'ingresso in questo sito costituisce una implicita dichiarazione che il visitatore rinuncia a considerare questa associazione, o chiunque dei suoi proprietari, fornitori di accesso, volontari e fornitori di contributi, responsabili per qualunque danno che possa risultare dall'utilizzo del materiale presente su questo sito. Il contenuto di questo sito è una pura risorsa informativa e non è qui per essere utilizzato in connessione con qualunque attività illegale. Prima di accedere a questo sito verificate che le vostre leggi locali non lo proibiscano. Se accettate i termini precedentemente esposti, e le leggi non vietano il transito di queste informazioni nella vostra area, sentitevi liberi di entrare, altrimenti scollegatevi ora."

Figura 6

Nella pagina centrale si può ancora leggere un informativa dell'autore che diffida tutti gli autori di contributi ospitati nel sito dal distribuire file eseguibili, segno che evidentemente questo è vietato dalle leggi vigenti nell'area in cui il sito è dislocato.

L'ultimo sito di cui si fa menzione è probabilmente dislocato nell'area delle filippine, e chiamato la "pagina dei link virali di Tally". In questa pagina non vi sono né sorgenti né virus né altro materiale disponibile, vi è invece una tabella contenente 208 link a siti di collezionisti e autori di virus; ogni voce è catalogata secondo una delle seguenti categorie: piccolo (small), medio (medium), pagina d'autore (author), informazioni (info), scambista (trading), gruppi (groups), anti-virus (anti), Unix, riviste elettroniche (zines). Le "pagine d'autore" sono delle home page personali contenenti materiale virale, come i "gruppi", che sono home page di gruppi di autori di virus. Inoltre, ad ogni voce è allegata la descrizione del contenuto del sito e la data di inserimento nella lista. La prima linea della lista contiene una dichiarazione di scarico di responsabilità simile alle altre già esaminate, il cui testo

afferma che le informazioni sono dedicate a coloro che studiano o collezionano virus, oppure sono ricercatori nel campo anti-virus.

5.3 - Siti dei principali produttori di anti-virus

Questo paragrafo è collegato in modo particolare con il paragrafo 4.3, pertanto vi si noteranno delle ripetizioni dovute all'interdipendenza degli argomenti trattati.

I siti delle industrie anti-virus sono una grande e importante fonte di informazione. Si rivolgono in genere all'utente medio, che, se non sempre è in grado di comprendere fino in fondo la terminologia scientifica, può comunque attingervi suggerimenti di natura pratica. Le informazioni presenti su questi siti, infatti, si limitano spesso solo a curiosità e dati essenziali per potersi difendere dai virus, procedure per contrastare particolari tipi di virus e annunci pubblicitari delle aziende, che presentano il proprio anti-virus come il migliore in assoluto.

L'elenco seguente di siti di industrie anti-virus è fornito dalla pagina link del sito dell'EICAR, descritto nel paragrafo seguente.

www.anet.cz/alwil	Avast!
www.avp.com	AntiViral Toolkit Pro
www.cheyenne.com	Cheyenne Software
www.commandcom.com	Command Anti-virus
www.drsolomon.com	Dr Solomon's Anti-Virus Toolkit
www.eliashim.com	Eliashim
www.datafellows.com	F-Secure
www.complex.is	F-Prot
www.antivir.de	Datentechnik Antivirenprogramm AntiVir
www.look.com	Look Software Systems, Inc.
www.mailessentials.com	Mail essentials Email anti-virus gateway
www.mcafee.com	McAfee (VirusScan)
www.norman.com	Norman Data Defense Systems
www.symantec.com	Symantec Norton Anti-virus

www.pandasoftware.com	Panda Anti-virus
www.safetynet.com	SafetyNet, Inc. (VirusNet)
www.segurasolutions.com	Segura Solutions Anti-Virus
www.sophos.com	Sophos
www.anti-virus.com	Trend Micro Devices (PC-Cillin)

5.4 - Siti degli osservatori internazionali sulla sicurezza.

Esistono alcune organizzazioni che si occupano di osservare il fenomeno dei virus; alcune di esse, in realtà, si occupano più genericamente di monitorare i problemi relativi alla sicurezza informatica e fornire materiale ed, in certi casi, assistenza a chiunque ne faccia richiesta. Si tratta di organizzazioni statali, universitarie, a volte legate ad istituti particolari ed altre volte indipendenti. Si dà una descrizione dei siti delle principali organizzazioni di questo tipo, prendendo in esame i servizi ed il materiale messo a disposizione.

I marchi, i loghi ed i nomi delle associazioni appartengono ai rispettivi proprietari.

- WildList

La WildList è sicuramente tra le più conosciute organizzazioni relative al campo degli anti-virus, se non la più conosciuta. Si tratta di un'organizzazione, il cui sito è www.wildlist.org, con lo scopo dichiarato di educare il pubblico sul problema della minaccia dei computer virus e di fornire sia al pubblico sia alle industrie anti-virus aggiornate, accurate e complete informazioni su virus che si trovano “in the Wild”, ovvero inseriti nella lista di quelli diffusi realmente in un certo momento. La dicitura “in the Wild” è diventata famosa nel campo della ricerca anti-virus, al punto da considerarsi parte del gergo informatico relativo a questo campo. La WildList, tra gli altri servizi, gestisce due liste di virus, la principale (detta WildList) e la supplementare; vi sono circa 55 volontari, ricercatori nel campo anti-virus sparsi in tutto il mondo, incaricati di rilevare le infezioni virali nella propria area geografica: quando uno di questi volontari rileva un'infezione di un determinato virus, lo segnala alla WildList. Queste segnalazioni sono archiviate ed elaborate nel seguente modo: la Wild List, o lista principale, comprende tutti i virus rilevati da almeno

due volontari di aree geografiche non corrispondenti; ogni voce viene inserita nella lista, che è aggiornata mensilmente, e vi rimane finché non trascorrono 18 mesi dall'ultima rilevazione. La lista supplementare viene compilata con un meccanismo identico, con la differenza che è sufficiente una sola rilevazione per accedervi. Ogni virus che si trova nella lista principale viene considerato "in the Wild"; questa lista viene utilizzata dalle industrie anti-virus e dalle organizzazioni che rilasciano certificazioni per prodotti anti-virus, per verificare la bontà dell'aggiornamento del database di virus rilevati. In effetti, molti anti-virus rilevano una quantità di virus vicina al 100% dei virus rilevati nel mondo, però si considera di importanza centrale la capacità di rilevare i virus presenti nelle Wild List più recenti. Il sito dell'organizzazione contiene un archivio con tutte le WildList pubblicate, dalla fondazione nel 1993; inoltre gli indirizzi di posta elettronica di tutti i volontari incaricati di rilevare le infezioni. Vi è inoltre un link al sito della F-Secure, che mette a disposizione il suo database di descrizioni tecniche dei virus per la descrizione degli elementi della WildList; la pagina è la seguente: www.f-secure.com/virus-info/wild.html, in questa pagina è presente una copia della WildList con le voci collegate con link alle rispettive pagine descrittive. Sono, inoltre, disponibili nel sito alcune pubblicazioni in formato elettronico e riferimenti a pubblicazioni, assieme ad alcune notizie ed informazioni sulle varie attività della WildList Organization ed argomenti correlati, come i test dei prodotti anti-virus ed il problema della convenzione per i nomi dei virus. L'organizzazione fornisce, ultimamente, un nuovo servizio denominato REVS, Rapid Exchange of Virus Samples, in altre parole scambio rapido di esemplari di virus; consiste in una mailing list cifrata che si occupa di diffondere esemplari di virus agli altri ricercatori in modo sicuro.

- CERT

Il significato di questa sigla è Computer Emergency Response Team. Attualmente esistono CERT locali nelle differenti nazioni; il primo CERT, che è anche considerato il centro di coordinazione degli altri, si trova negli Stati Uniti ed è integrato nella Carnegie Mellon University, istituto di ingegneria del software, a Pittsburg. Il CERT/CC, sigla con cui si indica il CERT Coordination Center, è nato per volontà dell'agenzia DARPA, immediatamente dopo l'incidente verificatosi in seguito alla diffusione dell'Internet Worm realizzato da Robert Morris (vedi paragrafo 1 e 3); il comunicato stampa originale con cui la DARPA comunicava notizia della fondazione del primo CERT specificava: "La DARPA

annuncia oggi che è stato stabilito un Team per rispondere alle emergenze dei computer (C.E.R.T.) per affrontare i problemi relativi alla sicurezza dei ricercatori utenti di Internet. Nel provvedere direttamente servizi alla comunità degli utenti di Internet, il CERT si focalizza sui bisogni specifici della comunità dei ricercatori e si presenta come un prototipo per istituzioni analoghe in altre comunità informatiche. Il CERT ha il compito di dare una risposta alle minacce per la sicurezza dei computer, come il recente problema dei virus che hanno invaso molti computer nell'ambito della Difesa e della ricerca universitaria."[23]

Il sito del CERT/CC, rispondente all'indirizzo www.cert.org, contiene un insieme di pubblicazioni, alert, informazioni e tools per risolvere i problemi relativi alla sicurezza in ogni campo; inoltre un elenco aggiornato di "advisories" o informative sulle vulnerabilità dei sistemi e dei protocolli aggiornato. Quindi non più principalmente i virus, che sono una minaccia di importanza non superiore ai problemi posti dalle recenti tecnologie riguardanti le transazioni elettroniche e le comunicazioni sicure. La pagina del sito riguardante i virus comprende: una sezione FAQ; una sezione raccolte di Hoax e catene di lettere relative; una sezione database di virus; una riguardante organizzazioni e pubblicazioni anti-virus; una riguardante i rivenditori di programmi per la verifica dell'integrità dei sistemi; una riguardante i produttori di software anti-virus; una riguardante pubblicazioni in formato elettronico sull'argomento; infine, una sezione intitolata "altre risorse" che comprende indicazioni su mailing list, siti diversi, newsgroup relativi all'argomento. Si nota, esaminando questa pagina, che il CERT non possiede un archivio proprio riguardante l'argomento; infatti, nelle sezioni elencate si trovano sempre dei link a risorse presenti su atri siti esterni al CERT, inseriti in istituzioni spesso private. Anche la sezione relativa al database di virus comprende diversi link relativi ai database delle diverse industrie anti-virus; vi è solo un link al database del CIAC (paragrafo seguente) ma, come da postilla, non è più aggiornato.

L'esempio ed il suggerimento della DARPA è stato seguito da numerose nazioni, esiste anche un CERT Russo, reperibile all'indirizzo www.cert.ru, in cui si trova anche una descrizione in lingua inglese. In Italia esiste un CERT situato presso il dipartimento di informatica ed applicazioni dell'università di Salerno, con un database di pubblicazioni, note informative di sicurezza, tools di crittografia, anti-virus e per la verifica della sicurezza in rete. Esiste anche un CERT situato nel Dipartimento Scienze dell'Informazione, presso

[23] Comunicato stampa della Carnegie Mellon University, 13 dicembre 1988, disponibile all'indirizzo http://www.cert.org/about/1988press-rel.html.

l'università di Milano, che si occupa anche di moderare il newsgroup italiano it.comp.sicurezza.cert-it.

- ICSA

L'ICSA è l'acronimo di International Computer Security Association. Si tratta di un'organizzazione profit che commercializza un sistema per la gestione della sicurezza delle reti, consistente in un pacchetto integrato di servizi comprendente help desk, fornitura di informative di sicurezza in tempo reale, consulenze immediate su qualunque tipo di problema relativo alla sicurezza possa presentarsi, esecuzione di analisi di robustezza del proprio sistema o rete in relazione alle intrusioni e alle problematiche relative alla sicurezza in generale. Inoltre l'ICSA si occupa di certificare prodotti commerciali inerenti la sicurezza informatica. Il sito contiene moltissime informazioni relative a diversi aspetti della sicurezza. Esso comprende: una rivista consultabile on-line cui è possibile sottoscrivere un'abbonamento, denominata "Information Security Magazine"; una sezione laboratori che fornisce un insieme di "consorzi" dedicati a diversi ambiti della sicurezza, tra i quali, quello dedicato al problema dei virus comprende a sua volta una raccolta di alert relativi ai più diffusi virus, una raccolta di hoax, l'elenco dei prodotti anti-virus certificati dall'ICSA, lo standard di certificazione richiesto, un database di virus del sistema Macintosh, un insieme di link a siti di industrie anti-virus, diverse recensioni di prodotti anti-virus. Vi sono poi sezioni dedicate alla raccolta di informative aggiornate sulla vulnerabilità dei sistemi e molte pubblicazioni inerenti l'argomento generale della sicurezza.

I requisiti dall'ICSA[24] per la certificazione degli anti-virus si possono esaminare nel sito proprietario. La pagina relativa ai certificati riporta differenti campi di certificazione dei prodotti anti-virus, elencati di seguito:

- Anti-Virus Scanner Criteria
- Anti-Virus Cleaning Criteria
- Anti-Virus Products for Internet E-mail Gateways Criteria
- Anti-Virus Products for Microsoft Exchange Server
- Anti-Virus Products for Lotus Notes
- Anti-Virus Scanner Criteria for Online Services

[24] http://www.icsa.net/html/communities/antivirus/certification/

L'obiettivo generale della certificazione, che l'ICSA dichiara di perseguire, è il significativo miglioramento dell'affidabilità commerciale e della sicurezza dei computer. L'obiettivo particolare è, invece, di rendere disponibile alla comunità degli utilizzatori una selezione di prodotti che forniscano i seguenti servizi:

1. Proteggere i sistemi informatici ed i supporti di memorizzazione dall'intrusione di computer virus.
2. Rilevare i virus su un sistema o un supporto di memorizzazione infettato.
3. Ripristinare l'operatività di un sistema soggetto ad un'infezione di virus informatici.

Assieme a queste informazioni, è possibile reperire la lista dei software certificati dall'ICSA e il programma per mantenere la certificazione valida. Si tenga presente che un prodotto anti-virus è soggetto ad un continuo aggiornamento.

- CIAC

Il CIAC, acronimo di Computer Incident Advisory Capability, servizio informativo sugli incidenti informatici, è un servizio interno del Dipartimento dell'Energia statunitense, ovvero DoE. Esso è nato come servizio interno di supporto tecnico per i calcolatori del DoE, comunque fornisce anche alcuni servizi rivolti alla comunità informatica di utilizzatori della rete Internet: note informative ed analisi delle vulnerabilità dei sistemi e delle possibili minacce per la sicurezza, training ed istruzione, analisi delle tecnologie. Il team CIAC fa parte del Computer Security Technology Center, che è un dipartimento del Lawrence Livermore National Laboratory, incaricato di fornire soluzioni per la gestione della sicurezza nelle agenzie governative statunitensi. Il LLNL è un laboratorio situato presso l'università della California e anch'esso al servizio del Dipartimento dell'Energia statunitense. Il CIAC è stato fondato nel 1989 ed è, assieme al CERT/CC, un team pionieristico nel campo del supporto alle emergenze informatiche. I suoi contributi per la comunità di utilizzatori di Internet sono riconosciuti in campo internazionale; è inoltre uno dei fondatori del FIRST, di cui si parla più avanti nel presente capitolo.

Il sito fornisce una quantità di materiale scientifico e divulgativo nel campo dei virus informatici. L'elenco delle risorse contiene:

- Una sezione relativa agli hoaxes, comprendente l'elenco e la storia degli hoaxes più diffusi; diversi elenchi per la ricerca all'interno del sito, per trovare un particolare hoax; la descrizione dei problemi generati dagli hoax; indicazioni su come riconoscere e come comportarsi se si riceve una mail che segnala un hoax.
- Una libreria di bollettini relativi alle vulnerabilità rilevate nei vari sistemi informatici e indicazione di come porvi rimedio.
- Un database di virus comprendente descrizione, informazioni ed effetti di numerosi virus relativi ai P.C., ai sistemi Macintosh e ad altri sistemi.
- Una sezione comprendente link e descrizioni dei principali tool anti-virus e relativi alla sicurezza, distribuiti via Internet.
- La sezione dei documenti serie 2300, ovvero una libreria di documenti in formato elettronico comprendenti un vasto insieme di argomenti relativi alla sicurezza dei computer e dell'informazione.
- L'archivio delle C-Notes, una collezione di articoli ed informazioni riguardanti la sicurezza dei computer distribuiti in base a reali bisogni rilevati dal CIAC.
- L'archivio delle catene di lettere più comuni circolanti attualmente su Internet.
- La pagina dei sistemi operativi, il cui intento è di fornire informazioni specifiche sui principali sistemi operativi e sulle loro specifiche problematiche di sicurezza.
- Una raccolta di link ad altri siti riguardanti il campo della sicurezza.

Gli indirizzi ai quali sono state reperite le informazioni presentate sono:

- www.ciac.org - indirizzo della Home Page del CIAC
- www.ciac.org/CIAC/CIACWelcome.html - indirizzo della pagina informazioni sul CIAC
- doe-is.llnl.gov – indirizzo della pagina relativa ai contenuti del Dipartimento dell'Energia statunitense
- ciac.llnl.gov/cstc/CSTCHome.html – Home page del Computer Security Technology Center, situato presso il laboratorio LLNL
- www.llnl.gov – Home page del Lawrence Livermore National Laboratory

- EICAR

L'EICAR, acronimo di European Institute for Computer Anti-virus Research, è un istituto privato, relativamente avaro di informazioni autobiografiche. L'obiettivo dichiarato è: "unire università, industrie, media assieme ad esperti nel campo della tecnologia, della sicurezza e legale provenienti da organizzazioni civili, militari, governative, per il rispetto delle leggi e della privacy, il cui obiettivo sia uno sforzo unito contro la creazione e la proliferazione di codice con fini malvagi, quali i virus informatici e i cavalli di Troia, e contro i crimini informatici, le frodi e gli abusi di computer e reti, inclusa lo sfruttamento dei dati personali per fini negativi", come si può leggere nella pagina relativa, all'indirizzo http://www.eicar.org/mission.html.

L'istituto comprende diversi gruppi di lavoro, presentati nelle pagine web, in cui si trovano le informazioni per i contatti, le aree di lavoro e gli obiettivi di ogni gruppo, il materiale ed i servizi realizzati, le date ed i temi dei convegni organizzati dall'EICAR, alcune pagine di link ad altri siti correlati ed una pagina contenente alcuni alert riguardanti cavalli di Troia. La maggior parte del materiale prodotto è disponibile ai membri, dietro il pagamento di una quota annua, differenziata a seconda della categoria di appartenenza.

L'EICAR è famoso per la realizzazione di un test riconosciuto in campo internazionale per la verifica del funzionamento di un qualunque scanner anti-virus. Una descrizione approfondita di questo test è contenuta nel paragrafo 4.3. L'indirizzo cui si può reperire il file contenente il test è http://www.eicar.org/anti_virus_test_file.htm.

- Virus Bulletin

Virus Bulletin, il cui sito è www.virusbtn.com, è una rivista scientifica inglese, la cui redazione è situata ad Abington, riguardante gli sviluppi nel campo dei computer virus e dei prodotti anti-virus. La rivista Virus Bulletin ha edizione mensile e ad essa collaborano, con articoli, i principali ricercatori nel campo Anti-virus. La rivista Virus Bulletin organizza, inoltre, una conferenza annuale di carattere tecnico, con interventi sui più recenti aspetti del problema dei computer virus; per l'anno 2000, per esempio, sono in programma alcuni interventi, tra cui la relazione sulla possibilità che i telefoni cellulari comprendenti la tecnologia WAP possano presto diventare oggetto di infezione virale. L'organizzazione

della rivista fornisce una sorta di certificazione del software, consistente in un riconoscimento intitolato "VB 100% Award" che certifica che il prodotto è stato sottoposto dallo staff della rivista ad una serie di test, da cui risulta che esso è in grado di rilevare, sia nella scansione manuale che in quella automatica, se supportata, il 100% dei virus compresi nell'elenco WildList più aggiornato. L'assegnazione di questo riconoscimento ha cadenza bimestrale e riguarda, a rotazione, gli anti-virus relativi ad una delle piattaforme più diffuse: Windows NT, Windows 98, NetWare, DOS, ecc. Nel sito è possibile reperire le seguenti informazioni:

- Il programma e le informazioni per l'iscrizione relative alla successiva conferenza, in fase di organizzazione;
- L'elenco dei vincitori del "VB 100% Award" sia dell'edizione attuale che di quelle passate, assieme ad alcune informazioni su come viene rilasciato questo riconoscimento, cosa rappresenta e come sono eseguiti i test.
- Le tabelle di prevalenza, compilate mensilmente, che contengono una relazione relativa alle segnalazioni di infezione, ricevute dallo staff della rivista, con le percentuali di infezione di ogni virus segnalato, rispetto al numero totale di segnalazioni.
- La WildList aggiornata e l'elenco completo delle WildList precedenti.
- Una pagina comprendente più di cinquanta riferimenti ad industrie anti-virus, per ognuna delle quali è riportato il nome del software anti-virus prodotto, l'indirizzo di posta elettronica del servizio informazioni, se presente, e l'indirizzo del sito.
- Una sezione riguardante il progetto VGrep. È noto che il problema della denominazione dei virus ha una certa consistenza; diverse fonti, come gli stessi software anti-virus, assegnano nomi differenti allo stesso virus. VGrep è un software che permette di creare delle tabelle relative ai differenti nomi con cui diversi anti-virus individuano lo stesso virus. Contiene un database con i nomi assegnati dai diversi anti-virus alla collezione di virus della VB, che viene utilizzato per la creazione delle tabelle; è sufficiente fornire al programma una stringa, contenente il nome o una parte del nome comune di un virus, e il programma visualizzerà una tabella con i diversi nomi assegnati dai principali anti-virus a quello stesso virus.

Sono presenti, nel sito, una piccola raccolta di pubblicazioni, una raccolta di editoriali e alcuni dati tecnici specifici sui test eseguiti per l'assegnazione degli award; queste ultime informazioni, stranamente, non sono accessibili dalla home page. Sono stati trovati soltanto

attraverso una rilevazione, effettuata da un motore di ricerca inglese, riguardo ad argomenti correlati al problema dei virus.

- IBM AV

L'IBM gestisce un sito dedicato alla documentazione sul problema dei virus informatici, il cui scopo è contribuire alla risoluzione del problema dei virus e fornire ai visitatori le informazioni e le notizie più rilevanti, con il fine di trasmettere le nozioni necessarie a divenire capaci di proteggersi. Il sito comprende otto sezioni:

- Stampa: in questa sezione sono contenuti articoli, link ad interviste contenuti nei siti delle agenzie informazioni principali, materiale informativo di base utile ai mass-media. Vi sono elementi di notevole interesse scientifico e divulgativo.
- Alerts: sono contenute le ultime note informative di sicurezza relative ai più recenti e diffusi virus, assieme alle ultime novità sugli hoax e su notizie errate diffuse dagli stessi produttori di anti-virus.
- Pubblicazioni: contiene tutte le pubblicazioni scritte sulla tecnologia Immune System, sull'epidemiologia dei virus informatici, ed altre notizie correlate. Contiene materiale di significativo interesse sia divulgativo che scientifico, utilizzato in maniera rilevante nella stesura di questo lavoro, in particolare sull'analisi delle motivazioni degli autori di virus, nel campo dell'analisi epidemiologica e riguardo alle future innovazioni.
- Virus: in questa sezione sono conservati ed archiviati gli alert meno recenti.
- Hoaxes: archivio delle rilevazioni di hoax ed hypes meno recenti.
- Wildlist: è contenuta una copia della lista WildList aggiornata.
- Education: contiene un insieme di note relative ad argomenti divulgativi principali, come un glossario, una storia dei virus, una serie di istruzioni per chi è stato infettato, una descrizione generale del problema dei virus, un insieme di pubblicazioni sulla ricerca anti-virus.
- Archivio: sono contenuti i numeri arretrati della rivista elettronica anti-virus online, prodotta dalla IBM.

È presente, all'interno del sito, un'altra sezione denominata "Inside The Lab", rivolta ad utenti esperti nel campo scientifico, reperibile all'indirizzo

www.av.ibm.com/InsideTheLab. Essa comprende, oltre che i link ad alcune delle sezioni già esaminate nel sito base, le seguenti sezioni:

- Virus description: contiene un elenco, presentato in ordine alfabetico, dei link a pagine con la descrizione tecnica dei più diffusi e recenti virus, comprendente circa duecento voci.
- Virus cross-reference: contiene una tabella che permette di individuare un virus attraverso i suoi differenti alias.
- Eicar test file: è il link alla pagina del test virus creato dall'EICAR.
- Sezione glossario.
- White papers: in aggiunta alle pubblicazioni, raggiungibili anche dal sito base, sono presenti alcune pubblicazioni avanzate che esaminano il problema virus da un punto di vista corporativo. È compresa anche un'interessante pubblicazione relativa alla disciplina anti-virus consigliata per le grandi aziende.
- Who's who: sono contenute le note descrittive dei principali ricercatori nel campo anti-virus, riconosciuti a livello internazionale. La lista comprende una quarantina di voci.
- Link: una sezione di link a siti di altre organizzazioni ed aziende anti-virus.

- TISSEC

Si segnala, infine, il gruppo di "speciale interesse" dell'ACM, denominato SIGSAC, ovvero Special Interest Group on Security Audit and Control, che sponsorizza la ricerca, attraverso l'organizzazione di workshops e conferenze sul tema, e si occupa del TISSEC, ACM Transaction on Information and System Security, giornale scientifico della famiglia ACM, la cui home page è www.acm.org/tissec.

6 - Riconoscimenti

Un particolare ringraziamento va al dottor Vesselin Bontchev del Virus Test Center, University of Hamburg, Germany, attualmente ricercatore presso la Friskies International, Reykiavik, Islanda, per il suo aiuto, per i suoi suggerimenti e per le nozioni tecnico-scientifiche messe a disposizione.

7 - Bibliografia

[1] H. J. Highland,
Computer virus handbook,
Elsevier Advanced Technology Science Publisher Ltd.
Oxford, 1990.

[2] Claudio Romeo, Alessandro Valli,
Guida ai virus dei computer,
Jackson libri srl, 1996.

[3] Pamela Kane,
V.I.R.U.S. Protection - Vital Information Resource Under Siege,
Bantam Books, 1989.

[4] Peter J. Denning,
Computer under attack - Intruders, worms and viruses,
ACM Press, New York, USA, 1990.

[5] Gabriele Faggioli,
Computer crimes,
Edizioni Esselibri – Simone,
Napoli, 1998.

[6] C. Nachenberg,
Computer Virus-Anti-virus Coevolution,
CACM: Communications of the ACM,
gennaio 1997, vol. 40, n.1 p.46.

[7] Fred Cohen,
Computational Aspects of Computer Viruses,
Computer and Security, Elsevier Science Publisher,
vol. 8, n.4, 1989, p.325-344.

[8] Fred Cohen,
Computer Viruses, Theory and Experiments,
Computers and Security, Elsevier publisher, n. 6, 1987, p. 22-35.

[9] Fred Cohen,
Models of Practical Defenses Against Computer Viruses,
Computers and Security, Elsevier Science Publisher,
vol. 8, n.2, 1989, p.149-160.

[10] H. J. Highland,
A Macro Virus,
Computers and Security, Elsevier Science Publisher, vol.8, 1989, p.178-188.

[11] Stoll, Clifford,
Stalking the Wily Hacker,

Communications of the ACM, vol. 31, n. 5, maggio 1988, pp. 484-497.

[12] Pamela Samuelson,
Can Hackers Be Sued for Damages Caused by Computer Viruses?,
Communications of the ACM, vol. 32, n. 6, giugno 1989, pp. 666, 668-669.

[13] Fred Cohen,
Why 32-Bit Desktops Need Virus Protection,
Datamation, vol. 41, n. 13, pp. 41-43, luglio 15, 1995.

[14] M. Alexander,
ThinkWrap - Hackers and virus writers are winning,
Datamation, vol. 41, n. 19, p. 106, 1995.

[15] B. C. Soh, T. S. Dillon, P. County,
Quantitative risk assessment of computer virus attacks on computer networks,
Computer Networks and ISDN Systems,
vol. 10, n. 27, pp. 1447-1456, settembre 1995.

[16] Philip Fites, Peter Johnston, Martin Kratz,
The computer virus crisis,
Van Nostrand Reinhold, New York, 1989, p. 77-85.

[17] Carolyn P. Meinel,
How Hackers Break In ... and How They are Caught,
Scientific American, vol. 279, n.4, p. 70-81, ottobre 1998.

[18] J. O. Kephart and G. B. Sorkin and D. M. Chess and S. R. White,
Fighting computer viruses,
Scientific American International Edition,
vol. 277, n. 5, pp. 56-61, novembre 1997.

[19] Paul Wallich,
Wire Pirates,
Scientific American, vol. 270, n.3, p. 90-94 98-101,
marzo 1994.

[20] A. K. Dewdney,
Computer Recreations Of Worms, Viruses and Core War,
Scientific American, vol. 260, n.3, p. 90-93, marzo 1989.

[21] J. Iliads, S. Gritzalis, V. Oikonomou,
Towards secure downloadable executable content: The Java paradigm,
Computer safety, reliability and security, 17th International
conference SAFECOMP '98 ,
LNCS vol. 1516, ottobre 1998, p.117-127.

[22] Paul-Michael Agapow,
Computational Brittleness and the Evolution of Computer Viruses,
Parallel Problem Solving From Nature,

IV. Proceedings of the International Conference on Evolutionary Computation,
LNCS, Springer-Verlag, vol. 1141, pp. 2- 11, 22-26 September 1996.

[23] Leonard M. Adleman,
An Abstract Theory of Computer Viruses,
Advances in Cryptology - CRYPTO'88,
LNCS, Springer-Verlag, n. 403, p.354, 1988.

[24] Allen Goldberg,
A specification of Java loading and Bytecode verification,
Kenstrel Institute, USA,
5th ACM Conference on Computer and Communications Security,
sponsored by ACM SIGSAC, p. 49-58, novembre 3-5, 1998.

[25] B. Menkus,
Hackers: know the adversary,
Computers and Security, Elsevier Science Publisher, vol. 10, n.5, pp.405-409, agosto 1991.

[26] Harold Joseph Highland,
Anatomy of Three Computer Virus Attacks,
Computers and Security, Elsevier Science Publisher, vol. 8, n.6, pp. 461-466, 1989.

[27] Vesselin Bontchev,
The problems of Wordmacro virus upconversion,
Computers and Security, Elsevier Science Publisher. vol. 18, n.3, p.241, 1999.

[28] Harold Joseph Highland,
Procedures to reduce the computer virus threat,
Computers and Security, Elsevier Science Publisher. vol. 16, n.5, p.439, 1997.

[29] Jon David,
The new face of the virus threat,
Computers and Security, Elsevier Science Publisher, vol. 15, n. 1, p. 13, 1996.

[30] Vesselin Bontchev,
Possible macro virus attacks and how to prevent them,
Computers and Security, Elsevier Science Publisher, vol. 15, n.7 p. 595, 1996.

[31] Alan Solomon,
The virus authors strike back,
Computers and Security, Elsevier Science Publisher, vol. 11, n. 7, p. 602, 1992.

[32] Silvana Castano, Giancarlo Martella,
Linee di tendenza nei virus del calcolatore,
Rivista di Informatica, edita da AICA (Associazione Italiana per l'Informatica e il Calcolo Automatico), Milano, vol. 21, n. 3, luglio-settembre 1991, p. 275.

[33] Jon David,
The novell virus,
Computers and Security, Elsevier Science Publisher, vol. 9, n. 7, 1990, pag. 593.

[34] Gerald L. Kovacich,
Hackers: Freedom Fighters of the 21st Century,
Computers and Security, Elsevier Science Publisher. vol. 18, n.7, p. 573, 1999.

[35] J. O. Kephart, W. C. Arnold,
Automatic Extraction Of Computer Virus Signatures,
Proceedings of 4th Virus Bulletin International Conference,
R. Ford editor, Virus Bulletin Ltd., Abington, UK, 1994, p. 178-184.

[35] E. H. Spafford,
The Internet Worm Program, An Analysis,
ACM Computer Communication Reviews,
vol. 19, n. 1, January 1989, p. 17-57.

[36] Gerald Tesauro, Jeffrey O. Kephart, Gregory B. Sorking,
Neural Networks for Computer Virus Recognition,
IEEE Experts, vol. 11, n. 4, agosto 1996, p. 5-6.

[37] Steve R. White,
Open Problems in Computer Virus Research,
IBM Thomas J. Watson Research Center,Yorktown Heights, NY, USA,
presentato a: Virus Bulletin Conference di Monaco, Germania, ottobre 1998,
http://www.av.ibm.com/ScientificPapers/White/Problems/Problems.html.

[38] David Chess,
Future of Viruses on the Internet,
presentato a: Virus Bulletin International Conference, San Francisco, California, USA,
ottobre 1997, disponibile presso:
http://www.av.ibm.com/ScientificPapers/Chess/Future.html.

[39] Sarah Gordon, Richard Ford,
Real World Anti-Virus Products Reviews and Evaluations,
Proceedings of Security on the I-Way. NCSA, 1995,
disponibile presso Computer Security Resource Center, National Institute of Standard and Technology, US Commerce Department:
http://csrc.nist.gov/nissc/1996/papers/NISSC96/paper019/final.PDF

[40] Sarah Gordon,
Anti-virus, Rx for AV,
Special report, infosec 2000, n.6,
http://www.infosecuritymag.com/nov99/infosec2000/Gordon.html

[41] Vesselin Bontchev,
comunicazioni personali,
2000.

[42] Vesselin Bontchev,
The Bulgarian and Soviet Virus Factory,
Proceedings of 1st International Virus Bulletin Conference, 1991, p. 11-25.

[43] Vesselin Bontchev,
Possible Virus Attacks Against Integrity Programs and How to Prevent Them,
Proceedings of 2nd International Virus Bulletin Conference, 1992, p. 131-141.

[44] Vesselin Bontchev,
Analysis and Maintenance of a Clean Virus Library,
Proceedings of 3rd International Virus Bulletin Conference, 1993, p. 77-89.

[45] Vesselin Bontchev,
Future trends of Virus Writing,
Proceedings of 4th International Virus Bulletin Conference, 1994, p. 65-82,
revisione del 1999 dalla home page dell'autore,
http://www.complex.is/~bontchev/future.html.

[46] Vesselin Bontchev,
Macro Virus Identification Problem,
Proceedings of 7th International Virus Bulletin Conference, 1997, p. 175-196.

[47] J. O. Kephart,
A Biologically Inspired Immune System for Computers,
pubblicazione on-line del centro anti-virus IBM,
http://www.av.ibm.com/ScientificPapers/Kephart/ALIFE4/alife4.distrib.html

[48] Peter V. Radatti,
The Plausibility of UNIX Virus Attacks,
Cybersoft Inc., febbraio 1993,
http://www.vfind.com/papers/plausibility.html

[49] Sarah Gordon,
The Generic Virus Writer,
Proceedings of 4th International Virus Bulletin Conference, Jersey, UK, settembre 1994,
disponibile presso centro anti-virus IBM,
http://www.av.ibm.com/ScientificPapers/Gordon/GenericVirusWriter.html.

[50] Sarah Gordon,
The Generic Virus Writer II,
Proceedings of 6th International Virus Bulletin Conference, Brighton, UK, settembre 1996,
disponibile presso centro anti-virus IBM,
http://www.av.ibm.com/ScientificPapers/Gordon/GVWII.html.

[51] A. Fedeli,
Corporate anti-virus discipline,
Centro ricerche IBM,
gopher://index.almaden.ibm.com/0VIRUS/PAPERS/FEDELI.TXT,
http://www.av.ibm.com/InsideTheLab/Bookshelf/WhitePapers/Fedeli/AVDISC/avdisc01.html fino a 08.html

[52] Yisrael Radai,
Integrity Checking for Anti-Viral Purposes: Theory and Practice,

Hebrew University, Gerusalemme, Israele, 1994,
http://www.virusbtn.com/OtherPapers/Integrity/integrity_ps.zip;
(revisione successiva di
Checksumming Techniques for Anti-Viral Purposes,
proceedings of 1st international Conference of Virus Bulletin, settembre 1991).

Pubblicazioni interne Symantec,
[53]*Understanding Heuristics,*
http://www-cl-1.symantec.com/avcenter/reference/heuristics.pdf;
[54] *Understanding and managing polymorphic viruses,*
http://www-cl-1.symantec.com/avcenter/reference/striker.pdf.

[55] Sarah Gordon,
The Anti-Virus Strategy System,
© 1995 Virus Bulletin, disponibile elettronicamente in:
http://www.av.ibm.com/ScientificPapers/Gordon/Strategy.html

[56] James R. Dukart,
Ways of Computing Will Evolve to Stay Ahead of Virus Threat,
Office.com, 30 agosto 2000,
http://www.office.com/global/0,2724,168-16102|1,FF.html

[57] Symantec, Banca dati descrizioni virus,
http://www.symantec.com/avcenter/venc/data/

[58] Kasperski Laboratories, Banca dati descrizioni virus,
http://www.avpve.com/viruses/

[59] TrendMicro, Banca dati descrizioni virus,
http://www.trend.com/vinfo/virusencyclo/

[60] David Noack
Computer Viruses Cost $12 Billion in 1999,
APB NEWS, Jan. 20, 2000,
http://www.apbnews.com/newscenter/Internetcrime/2000/01/20/virus0120_01.html

[61] Kathleen Ohlson,
'Love' virus costs approaching $ 7 Billion, research firm says,
Computerworld, 9 maggio 2000,
http.//www.computerworld.com/cwi/story/frame/0,1213,NAV47_STO44810,00.html

[62] Dati sulla sicurezza dell'ACM con link a moltissime pagine interne ed esterne
ACM Transactions on Information and System Security
http://www.acm.org/tissec/Topics.html

[63] CIAC Computer Incident Advisory Capability,
U.S. Department of Energy, Computer Security Technology Center (CSTC),
Lawrence Livermore National Laboratory (LLNL),
Virus Information Update,
http://ciac.llnl.gov/cgi-bin/index/documents/

CIAC-2301_Virus_Information_Update_5-98.pdf;

[64] *Microsoft "Office HTML" and "IE" Script Vulnerabilities,*
http://www.ciac.org/ciac/bulletins/k-061.shtml.

Alan Solomon,
[65] *A guide to evaluate anti-virus software,*
[66] *Guidelines for Anti-virus Policy,*
[67] *Macro Virus Heuristic,*
[68] *Java, Activex and Virus Threat,*
[69] *IRC Worms,*
[70] *The Cookie Monster: The Risks of Internet Cookies and Aggregate Data,*
pubblicazioni tecniche on-line,
McAfee.Com Corporate,
http://www.drsolomon.com/vircen/vanalyse/index.cfm.

[71] Robert M. Slade,
History of Computer viruses,
1992, versione elettronica conservata presso:
http://bioinformatics.bocklabs.wisc.edu/~janda/sladehis.html

[72] Alan Solomon,
S&S International,
A Brief History of PC Viruses,
versione elettronica conservata presso:
http://bioinformatics.bocklabs.wisc.edu/~janda/solomhis.html.

[73] H. Fuhs,
Encryption Generators Used in Computer Viruses part 1 e 2,
Fuhs Security Consultants,
Network Security Center, Computer Virus Research Lab, Wiesbaden, Germany,
http://www.fuhs.de/fachartikel/Encryption_Generators_Used_in_Computer_Viruses_Part_1.html e part2.html.

[74] Nick Fitzgerald, Lehigh University,
VIRUS-L comp.virus Frequently Asked Questions (FAQ) v2.00,
news:comp.virus.

[75] David Harley, ICSA,
Viruses and the Mac FAQ,
news:comp.virus.

[76] Sito del Dipartimento Anti-virus IBM.
http://www.av.ibm.com/
http://www.av.ibm.com/inthelab/

[77] Sito di Virus Bulletin International
http://www.virusbtn.com/

[78] Sito della International WildList Organization,

http://www.wildlist.org/

[79] Sito di ICSA, International Computer Security Association,
http://www.icsa.net/html/communities/anti-virus/index.shtml

[80] Sito della European Institute for Computer Anti -Virus Research.
http://www.eicar.org/

[81] Sito del CERT/CC, Computer Emergency Rescue Team / Coordination Center,
Carnegie Millon University,
http://www.cert.org/nav/aboutcert.html

[82] Sito del CERT Italiano presso Dip. Scienze dell' Informazione, Università di Milano
http://security.dsi.unimi.it

[83] Sito del CERT Italiano presso Dip. Scienze dell' Informazione, Università di Salerno
http://cert.unisa.it

[84] Sito della Trend Micro Inc.,
http://www.trendmicro.com/

[85] Sito della Panda Software Inc.,
http://www.panda.com/

[86] Sito della Cybersoft Inc.,
http://www.cybersoft.com/

[87] Sito della Friskies International,
http://www.complex.is/

[88] Sito dei Kaspersky Laboratories,
http://www.avp.com/

8 - Contatti

<scoeli@tiscali.it>
http://web.tiscali.it/_stefano_ (underscore stefano underscore)