*Vírus Obs: este texto é de 1988, e não foi atualizado.

1 Um cenário possível

Imagine um usuário trabalhando sozinho em casa. Ele tem um PC devidamente equipado com um winchester, e está terminando de digitar um texto, que para variar, deveria ter sido entregue ontem.

Em um dado momento, ao dar um comando DIR sobre o winchester, ele percebe que alguns arquivos desapareceram.

Começa uma busca desesperada, e parece que os arquivos continuam desaparecendo.

No ambiente que era familiar e completamente sob controle, começam a acontecer coisas sobre as quais o usuário não tem nenhuma soberania. Tampouco tem consciência do que está a acontecer.

Tudo termina com a perda total ou parcial dos arquivos, seguida de uma perda pior ainda: a confiança cega que o usuário depositava naquele computador.

Esta é uma possível descrição para um ataque de "vírus"(ou verme) de computador. Tal cenário tem ocorrido freqüentemente no último ano, e prevê-se que nos próximos, este estado de coisas se agrave.

Vírus (do latim: veneno), Diminuto agente infeccioso, invisível, não tem metabolismo independente, tem capacidade de reprodução apenas no interior de células vivas. Reproduz-se com continuidade genética, podendo sofrer mutações. (Mestre Aurélio).

O grande interesse que os vírus de computadores tem despertado, principalmente entre os leigos, reside na semelhança de sua atuação em relação aos vírus biológicos.

Desde já é preciso se ressaltar, que embora tenham origens totalmente diferentes (isto é, não tem nada a ver um com outro), certas semelhanças de comportamento entre os vírus biológico e de computador justificaram toda a celeuma que a imprensa promoveu sobre o assunto.

Nesse ponto de vista, se justificam as recomendações a princípio em tom de gozação, mas que como veremos, são muito pertinentes, tais como:

"Faça do micro seu único parceiro", ou "Use disquetes com camisinha (selo de proteção)".

Estas orientações se justificam a partir do momento em que o uso descontrolado de computadores e disquetes, com trocas múltiplas de discos e máquinas, sem controle de quem usou o que e quando, caracterizam um ambiente propício ao surgimento e disseminação dos vírus. Parafraseando os especialistas em AIDS, podemos dizer que os vírus crescem e causam problemas em ambientes de alta promiscuidade.

2 Definição

A prioridade no uso deste termo, é do americano Fred Cohen na sua tese de doutorado em 1983, que versava sobre segurança de sistemas. Na definição de Cohen, "um vírus é um programa que pode infectar outros programas, modificando-os de maneira a permitir a inclusão de uma cópia sua".

Para nossos propósitos, podemos usar a seguinte definição:

Um verme ou vírus, é um programa de computador (software), com duas características:

1. Se reproduz sem que o usuário perceba.

2. Causa problemas no computador, em geral muito tempo depois que ele se instalou.

Podemos afirmar com certeza, que fazer um programa com estas características não é fácil. Talvez a maior dificuldade foi ter a idéia inicial. Mas, mesmo depois, não é qualquer um que consegue escrever um vírus, que funcione bem.

Se quiserem atender aos dois requisitos acima, os vírus precisam ser programas de alta qualidade, bem feitos e testados. Especialistas estimam que para escrever o ISRAELI (um dos primeiros vírus), foram necessários 4 meses/programador, de excelente qualidade.

Para escapar ao olho do usuário, eles invariavelmente precisam trabalhar em baixo nível, isto é muito próximos do hardware do computador. Vírus (de PCs) invariavelmente são programados em Assembler, e usando os recursos do BIOS (Basic Input Output Services).

Outro fator que limita os vírus é o seu tamanho. Considerando que eles devem se reproduzir em mídias magnéticas compartilhadas, das quais o maior representante é o disquete de 5 1/4", com capacidade máxima de 360Kbytes, verificamos que eles têm que ser pequenos. Aliás esta é característica indeclinável se eles quiserem passar um bom tempo desapercebidos. Em resumo, a maioria dos vírus conhecidos têm um tamanho médio de 2000 bytes. Considerando uma média de 4 bytes por instrução de máquina, chegaremos ao resultado de aproximadamente 500 instruções de máquina por vírus.

3 Programando em camadas: a cebola

O universo do software tem se expandido nos últimos 30 anos, graças a uma idéia tão simples quanto radical: a divisão de um problema complexo (como aliás é implementar qualquer rotina em linguagem de máquina, que relembrando é a única a ser entendida pela máquina) em diversos níveis.

O surgimento das linguagens de alto nível (Fortran, COBOL, APL), nada mais é do que uma extensão do conceito. Se para o homem era (e é) difícil dialogar diretamente com a máquina em linguagem de baixíssimo nível, a solução foi colocar uma série de intermediários.

Cada nível recebe ordens do nível anterior e ordena ao próximo nível, desconhecendo os níveis que não são seus vizinhos. Agindo assim, cada um dos níveis pode se concentrar um tópicos específicos, um de cada vez, agilizando e simplificando a tarefa de criar software.

Uma materialização deste conceito seria a representação de uma cebola, onde cada camada, conhece a anterior e a posterior, ignorando a presença e a função das demais camadas.

No caso específico do PC (que é o que nos interessa aqui), a primeira camada (a mais interna) seria representada pelo hardware. Os equipamentos que lêem e gravam (discos), ou que recebem dados (teclado) ou que os cedem (alto-falante, vídeo, impressora etc).

Esta camada recebe ordens simples, tais como: imprimir um caractere, indicar qual a tecla foi apertada, gerar um som etc.

Sobre esta camada, encontramos os controladores do hardware, que recebendo ordens específicas, atuam sobre o hardware. Exemplos desta camada, são os controladores de disco, de vídeo etc.

A terceira camada é representada pelo BIOS. Trata-se de um conjunto de programas, gravados em ROM (isto é preparados em fábrica) destinados a conversar intimamente com os controladores do hardware. Apenas as funções básicas estão programadas, como por exemplo: ler um setor, imprimir um caractere etc.

A quarta camada, é representada pelo DOS (disk operating system). Esta camada, conversa com o BIOS reduzindo ordens complexas (como por exemplo DIR *.*) ao conjunto de ordens simples necessárias ao BIOS executar o pedido.

A quinta camada, pode ser representada por um aplicativo, seja pacote, seja linguagem. Para simplificar poderíamos considerar o dbase, como sendo esta camada. Assim, ela ordenaria ao DOS quais arquivos manusear e de que forma.

A sexta camada, no nosso exemplo, poderia ser representada pelos programas escritos pelo usuário. Eles ordenariam ao dbase o que e quando fazer.

Finalmente a última camada é o usuário.

Atente-se que:

- A rigor, as únicas camadas indispensáveis são a primeira (hardware) e a última (usuário). Por exemplo, ao mandar a RAIS para o banco em disquete, o chefe de pessoal (sétima camada) só precisa gravar o disco (primeira camada).
- Todas as demais são artificiais, e visam simplificar e padronizar os procedimentos para obter resultados produtivos usando o computador.
- Cada camada, só conhece seus dois vizinhos: O gerenciador dbase, só reconhece o DOS de um lado, e o programa do usuário de outro. Ele não sabe o que é BIOS, controlador, etc.
- Cada camada filtra um pouco da complexidade anterior, permitindo um nível de abstração cada vez mais alto, e conseqüentemente afastando o usuário da máquina.

3.1 Exemplos da cebola

O BIOS reconhece a existência de 3 dimensões nos setores de disco: face, trilha e setor. J o DOS imagina o disco como um conjunto linear de setores (uma única dimensão).

O DOS tem mais de 50 funções de acesso a disco, enquanto o BIOS só tem 5 (reset, leitura, gravação, formatação e verificação). Toda a organização de diretórios, sub-diretórios, arquivos, FATs, etc fica retida a nível de DOS. O BIOS desconhece tudo isto.

4 Histórico

Certamente, ninguém é capaz de descobrir qual a real origem dos vírus de computador. Começa pela dificuldade de estabelecer onde cada vírus surgiu, (afinal, os vírus não são documentados), e prossegue

pelo fato quase óbvio, de que os vírus não têm origem única.

Mas, apesar das dificuldades, alguns marcos podem ser citados. Em 49, Von Neumann, o criador dos computadores no seu aspecto teórico, escreveu um "paper"de título *"Theory and Organization of Complication automata"*. Neste artigo ele defende a possibilidade dos programas se reproduzirem e competirem por recursos dentro dos computadores. Os que leram o artigo acharam que Von Neumann fora longe demais: - Os computadores estavam engatinhando e nem a mais deslumbrada mente otimista poderia imaginar o que eles fariam à humanidade. A idéia, é que os computadores são vulneráveis a um tipo de auto-destruição. Por compartilharem a mesma memória, os dados e os programas estão sujeitos ao mesmo tratamento. Assim, é perfeitamente possível um programa se alimentar de outros programas, em vez de se alimentar de dados (que é o usual).

10 anos depois, três pesquisadores (todos na faixa dos 20 anos: Douglas McIllroy, Victor Visottsky e Robert Morris), trabalhando nos laboratórios da Bell, AT&T, resolveram inventar um jogo (que ali só era jogado à noite, quando a carga da máquina era pequena). Neste jogo, durante um certo intervalo de tempo, digladiavam-se na memória do computador, diversos programas cujo nome era organismos, que tinham dois objetivos:

- se reproduzir rapidamente
- eliminar os programas adversários, comendo suas instruções e conseqüentemente aumentando de tamanho.

Ao final, ganhava o jogador cujos programas fossem mais volumosos, e estivessem em maior número na memória.

O nome do jogo era CORE WAR, (guerra na memória), e ele começou a despertar muita curiosidade no meio acadêmico e de pesquisa. Inúmeros laboratórios (IA Lab do MIT, Palo Alto Research da XEROX etc) se interessaram e começaram a entrar no jogo. Os autores nunca tiveram receio, pois as máquinas eram stand-alone (isoladas) e se a situação fugisse ao controle, bastava desligar a máquina em questão. A gerência da AT&T Bell, era conivente com isso, pois encarava a iniciativa como um alargamento do conhecimento na área de PD.

Alguma ocorrência deve ter feito com que os criadores originais voltassem atrás em sua decisão, e eles resolveram destruir todos os fontes, documentação etc do CORE WAR. Fizeram também um pacto entre eles de jamais revelar a maneira de funcionamento e a implementação do jogo.

Muitos anos depois, (em 83), Ken Thompson, o criador do UNIX, ao receber o prêmio Turing (concedido pela ACM, e considerado o prêmio Nobel da informática), em seu discurso citou o CORE WAR, indicando seu funcionamento e as maneiras de implementá-lo.

4.1 Produtoras de software

As empresas produtoras de software têm sido acusadas de terem originado o fenômeno "vírus". Esta acusação é procedente, pois pelo bem ou pelo mal, o vírus combate a pirataria. Aliás tem sido muito mais eficaz neste combate, do que leis, campanhas publicitárias e similares.

O combate se dá pela razão de que a melhor defesa contra os vírus é evitar a promiscuidade. Isto pressupõe o uso de softwares de origem conhecida, com garantia do produtor e do qual não são feitas cópias, exceto para back up. Os fabricantes não querem nada mais do que isto.

Um dos poucos vírus do qual se conhece a origem (o Pakistani Brain) comprovadamente nasceu como uma forma de punição consciente dos autores dele contra os piratas.

4.2 Vendedoras de vacinas

Empresas interessadas em vender vacinas. Quando uma virose aparece, na sua forma virulenta, ocasionando perda de dados e parada (total ou parcial) da organização, não há dinheiro que pague uma solução rápida e indolor para o problema. Assim, não são poucos os que imaginam que uma possível explicação para os vírus passa pela seqüência:

1. Alguém inventa um vírus
2. Ato contínuo, escreve uma vacina 100
3. A dupla é customizada, com detalhes profundos casados
4. O vírus é disseminado

5. Depois que o problema assuma proporções grandes, aparece a salvação: ”uma vacina”
6. A vacina é vendida
7. O vendedor é o ”salvador da pátria”.

4.3 SOFTWAR

Em 1978, dois autores franceses (T. Breton e D. Beneich) escreveram o livro SOFTWAR, que trata do vírus como arma na guerra fria entre as potências (EUA e URSS). Embora fosse ficção, o livro chegava bem perto do conceito de vírus. A seguir, um resumo do livro:

Os franceses desenvolvem o software para um sistema de previsão meteorológica, e que roda em um supercomputador Gray (Cray). Os soviéticos manifestam o desejo de adquirir o conjunto (hardware+software). Os franceses gostam da idéia, pois ganharão um bom dinheiro, e migrarão para um CRAY II. Entretanto, uma cláusula no contrato original, obriga os franceses a consultar os americanos antes de efetuar a venda.

Os americanos não se opõem, desde que eles tenham a possibilidade de ”revisar”o software francês, e se certificar de que não há nenhum segredo militar no programa.

Na ”verificação”, o que eles fazem realmente é colocar um vírus que quando ativado, ordenará ao computador fazer loucuras.

O programador do vírus é um professor universitário americano, que teve um caso com uma estudante de informática soviética, há 10 anos. Coincidentemente (?) a estudante soviética agora é a diretora do programa russo de informatização, e é quem vai receber o computador Cray. Coincidentemente também, a soviética tem uma filha de pai ”desconhecido”e que tem 10 anos.

O sistema meteorológico se liga a uma rede mundial de recepção, análise e distribuição de dados sobre clima. O gatilho do vírus é a informação de que a pressão barométrica sobre a ilha de Santo Tomás (no caribe) é de 1029 milibares. O posto de meteorologia da ilha, que é americano, recebe ordens severas de jamais alimentar a rede mundial de computadores com esta pressão. Quando ela ocorresse, deveria ser informado 1030 milibares.

O gatilho é desarmado quando a pressão de 1028 milibares sobre a ilha é informada.

O único defeito deste vírus é que ele só pode ser detonado quando as condições climáticas reais se aproximam dos valores de gatilho, a fim de que ninguém desconfie.

Um dia o vírus é experimentado, e os soviéticos se vêem em palpos de aranha. Um brilhante programador russo fica intrigado - não com o erro - mas com o súbito desaparecimento do bug, e resolve começar a investigar o sistema meteorológico. Ele liga um TRACE e depois de algum tempo analisa quais os caminhos do programa que nunca ou quase nunca são percorridos. Na teoria dele, é aí que está o vírus.

Acertou na mosca, mas logo após comunicar à sua chefe sua descoberta, ele desaparece sequestrado pela KGB.

O sucesso da experiência anima os americanos a lançar a segunda fase da softwar: Um novo vírus, revendido através de um software de controle de redes de comunicação originalmente desenvolvido da Índia, e que os soviéticos estão secretamente adquirindo.

Enquanto isso, a mocinha da história, começa a pensar porque a KGB teria dado um sumiço em seu assistente. Desconfiada, ela começa a investigar todos os computadores da rede soviética e verifica que em todos houve uma ”manutenção”da KGB que colocou em cada um, uma ROM que será ativada quando uma palavra chave for fornecida à rede de computadores. Quando o firmware começar a funcionar, ele deixará inativo o computador.

O fato dos russos terem infectado sua própria rede, se explica pela luta de facções no Kremlin: é a época do fim de Yuri Andropov (e do seu eficiente assistente: Mikail Gorbachev) e do seu sucessor: Konstantin Tchernenko. Relembrando, enquanto Andropov lutava pela modernização da sociedade, Tchernenko era contrário a essa modernização, e não queria saber nada de computadores.

A história prossegue, e logo após a heroína descobrir a sabotagem soviética, ela vai a uma conferência de FDT (fluxo de dados transfronteiras), levando a filha, e onde é claro se encontra o professor americano.

Mil rolos, e a filha é raptada. Ameaças de guerra nuclear, confusões, todo mundo procurando a menina, e ela havia ido se refugiar no quarto de hotel do pai dela, sem que ninguém se apercebesse disso.

No acerto final entre os dois (o mocinho e a mocinha), este lhe implora a palavra chave da KGB que detonaria o vírus soviético, e ela como boa russa, desconfiada de qualquer americano (ainda que seja o mocinho) recusa.

Na última página, ele vai se despedir das duas (mãe e filha) no aeroporto, e enquanto a mãe passa soberana, sem sequer lhe dirigir a vista, a menina corre para abraçá-lo e dizer-lhe ao ouvido, um segredo da mãe: a palavra chave.

5 Bomba relógio e cavalo de tróia

Embora o assunto vírus seja novo, não o são os programas do tipo bomba relógio e cavalo de tróia. Eles podem ser considerados os ancestrais dos vírus, e são estudados há mais de 20 anos.

O conceito mais simples é o de bomba relógio. É um trecho de programa clandestino, que é colocado deliberadamente pelo autor em meio a um programa normal, cuja única característica é pertencer a algum ciclo periódico do sistema.

O trecho clandestino, busca a ocorrência de um evento (gatilho), e enquanto ele não ocorrer, o programa bomba permanece adormecido. No dia em que as condições se realizam (e provavelmente o autor já está longe), o programa ataca.

A diferença para o vírus, é que a bomba relógio não se reproduz, (a menos que cópias do sistema sejam distribuídas), mas de qualquer maneira não é uma auto-reprodução, como ocorre nos vírus autênticos.

A literatura registra inúmeros casos de bomba retardada, alguns já fazendo parte do folclore, como aquele programador que colocou um teste no sistema de folha de pagamento, para ver se ele (o programador) ainda constava dos arquivos. No dia em que isto deixou de acontecer (isto é o programador foi demitido, ou pediu demissão), o sistema desandou a fazer loucuras.

Já o conceito de cavalo de tróia, é um pouco mais sofisticado, e se destina a infiltrar em sistemas que tenham proteção contra usuários comuns. Nas máquinas IBM, por exemplo, existem dois modos de processamento: o normal e o privilegiado. Certos comandos só podem ser emitidos no modo privilegiado. Para fazer um cavalo de tróia, nosso hacker (que normalmente não tem acesso ao modo privilegiado), escreve um programa utilitário (que contém o cavalo de tróia) e deixa-o residente na biblioteca compartilhada de utilitários. O programa ao ser chamado pergunta: "Estou no modo privilegiado?"Se não estiver, ele aguarda quietinho. Um belo dia, um analista de suporte ou qualquer profissional de manutenção (que tem status de privilegiado) resolve usar o programa utilitário. Neste momento, as condições propícias aparecem: o programa é chamado e ganha o controle da CPU em modo privilegiado, e o trecho terrorista é executado em modo privilegiado.

Resta-nos a constatação de que certamente não houve uma causa única: com certeza as condições para o surgimento do fenômeno foram aparecendo aos poucos e em mais de um lugar ao mesmo tempo. O resultado está aí: uma epidemia.

6 Começo para valer

As condições ideais para as epidemias dos nossos dias se formaram há muito poucos anos: de 88 para cá. Tais condições são:

- A verdadeira disseminação das máquinas tipo PC (Só no Brasil estima-se existirem mais de 500.000 PC compatíveis). Estas máquinas não têm nenhuma proteção, e qualquer programa que se instale na máquina tem acesso a todos os recursos dela.
- Redes públicas tipo BBS (Boulletin Board Systems). Tais redes são mania nacional nos EUA, lá existem mais de 3.000 conhecidas. Nestas redes, qualquer proprietário de um computador, disca para um certo número e efetua a conexão da sua máquina a outra, onde o sistema reside (via modems). Estabelecida a comunicação, as máquinas trocam praticamente tudo: textos, programas fonte e objeto, trechos de programas, subrotinas, macetes etc (e vírus também são trocados, é claro).
- Redes particulares de comunicação de dados, sejam de micros, sejam de mainframes. Aqui, programas são distribuídos freqüentemente, o que facilita a disseminação dos vírus.

A primeira grande virose, que inclusive chamou a atenção dos meios de comunicação (não PD) para o assunto, ocorreu em novembro de 1988. Nessa época, houve uma infestação em larga escala na rede ETHERNET, e o autor foi Robert Morris Tappan Jr, um estudante de 24 anos, do curso de pós graduação em informática da Universidade de Cornell na Califórnia. Como curiosidade, este rapaz é filho de Robert Morris especialista em segurança de computadores e funcionário da NASA (que não escapou à contaminação).

Os prejuízos foram estimados em US$ 100.000.000 (cem milhões de dólares) em perda de horas de máquina e em mão de obra especializada para reparos.

Há algumas semanas, o autor da infecção foi indiciado em tribunais americanos, e pode ser condenado a pagar uma multa de até 250.000 US$, acompanhado de uma pena de prisão de até 5 anos.

No Brasil, as infecções têm ficado praticamente restritas ao universo dos PCs, uma vez que temos poucas redes públicas (menos de 10), e mesmo nossas redes privadas são em pequena quantidade e diversidade. Assim, este trabalho, daqui para a frente privilegiar as viroses em PC, que são a nossa maior ameaça.

6.1 Importante

Como não estamos falando de entes sobrenaturais, certas coisas (óbvias, mas importantes) precisam ser ditas:

Um programa precisa estar na memória e receber o controle da CPU para poder atuar. Enquanto ele estiver em disco, por mais virulento que seja, nenhum mal causar . Para poder ser considerado um vírus, esta carga na memória precisa ser feita sem que ninguém perceba.

Para que isto aconteça, o vírus precisa se "grudar"em outro trecho executável e este sim, inofensivo. São candidatos:
a) O programa de boot
b) O processador command.com
c) qualquer programa executável.

O vírus precisa se duplicar. No caso do PC isto é feito, quando o vírus se auto-grava em mídia magnética, preferencialmente compartilhada (disquete). Também isto deve ser feito sem que o usuário perceba, e a maneira é interceptar as chamadas de DOS (interrupção 21h) e decidir quando é hora de se gravar no disco.

Obviamente, o vírus ocupa algum espaço no disco, e portanto é necessário conhecer o mapa de alocação interno do mesmo.

Fazendo uma comparação com uma doença humana, por exemplo, a dengue. Ela só atua no homem, aqui causando estragos. Entretanto, é disseminada pela ação de um mosquito, que ao picar uma pessoa doente, se contamina e mais tarde ao picar uma pessoa sã, contamina-a também. O mosquito nada sofre com a doença. No nosso exemplo, o mosquito equivale ao disquete e o homem ao computador com sua memória.

7 Anatomia

Assim como os médicos começam sua formação profissional estudando anatomia e vendo o funcionamento de cada uma das partes do corpo humano, também nós para estudar a manifestação dos vírus, precisamos conhecer como funciona internamente um computador do tipo PC.

Tendo isto em mente, vamos fazer uma rápida parada no estudo do vírus e vamos passar a olhar internamente o PC.

As partes do PC que nos interessam neste momento são: a memória e os discos magnéticos.

7.1 Memória

A memória do PC está organizada como uma série de bytes. Cada um deles é individualizado pelo seu endereço, tal como se tivéssemos uma seqüência de caixas postais, todas numeradas. O primeiro byte da memória recebe o endereço 0 (zero), e o último byte recebe um número que varia de acordo com a capacidade da memória do micro. Tipicamente, o valor máximo para PC é de 640 Kb. O PC tem a habilidade de trabalhar com números armazenados em palavras (2 bytes), cuja valor máximo, considerando números apenas positivos é de 65.536. Isto sugere que os endereços dentro do PC (que são palavras) somente podem atingir este valor. Isto é parcialmente verdadeiro. A questão é: Como o PC pode atingir endereços maiores do que 64K ? A resposta a isto, é o que se chama de endereços segmentados: Trata-se da combinação de 2 palavras, que dão como resultado um número de 20 bits (portanto podendo atingir $16 \times 65.536 = 1.048.576$). Os endereços no PC são, representados por dois valores de 16 bits, por exemplo: 8D88:1004. Cada um destes valores recebe um nome próprio: O primeiro, escrito antes dos dois pontos, chama-se "segmento", e o segundo chama-se "off-set"ou deslocamento. O endereço absoluto final resultante desta combinação é assim obtido: (considerando-se números hexadecimais)

1. Multiplica-se o valor do segmento por 16 (o que significa colocar um zero à sua direita).
2. Soma-se com o valor de off-set
3. Este resultado é o endereço absoluto desejado.

Veja-se no exemplo dado acima: (8D88:1004)
$8D880 + 1004 = 8E884$
Suponhamos o endereço: F230:0385, visto de modo real:

```
+---------------------------------+
| 1 1 1 1 0 0 1 0 0 0 1 1 0 0 0 0 | 0 0 0 0  => equiv. F230 x 16
+---------------------------------+
          0 0 0 0 0 0 1 1 1 0 0 0   1 0 0 1  => equiv. a 0385
-------------------------------------------
  1 1 1 1 0 0 1 0 0 1 1 0 1 0 0 0   1 0 0 1  => equiv. a F2685
```

7.2 Espaço de endereçamento

O número 1.048.576 é considerado o espaço de endereço do PC. Isto significa que potencialmente o PC pode endereçar até este valor de bytes. Mas não significa, como veremos a seguir, que o PC tenha esta memória toda. Para efeitos didáticos, vamos dividir este espaço de 1Mbytes em 16 blocos de 64Kb cada. Na lista a seguir, o que significam e contém os blocos dentro de um PC padrão:

Blocos de memória do PC	
Blocos de memória dentro do PC	Endereço inicial
Bloco 0 - Memória do usuário até 64K	00000
Bloco 1 - Memória do usuário até 128K	10000
Bloco 2 - Memória do usuário até 192K	20000
Bloco 3 - Memória do usuário até 256K	30000
Bloco 4 - Memória do usuário até 320K	40000
Bloco 5 - Memória do usuário até 384K	50000
Bloco 6 - Memória do usuário até 448K	60000
Bloco 7 - Memória do usuário até 512K	70000
Bloco 8 - Memória do usuário até 576K	80000
Bloco 9 - Memória do usuário até 640K	90000
Bloco A - Memória estendida de vídeo	A0000
Bloco B - Memória padrão de vídeo	B0000
Bloco C - Expansão da ROM (XT, EGA etc)	C0000
Bloco D - Outros usos	D0000
Bloco E - Outros usos	E0000
Bloco F - BASIC residente e ROM-BIOS sistema	F0000

O Bloco F, contém as rotinas de ROM. Podemos considerá-las compostas de quatro partes a saber: o programa de início do computador, o BIOS (Basic Input/Output System), o BASIC residente e as extensões de ROM.

7.3 O programa de início

O programa de início (start-up) executa diversas tarefas: Inicialmente ele roda um teste em todos os componentes da máquina para saber se estão em ordem e respondem corretamente. Estabelece a tabela de vetores de interrupção que contém os endereços dos programas manuseadores das interrupções dentro do BIOS. Verifica se existem equipamentos opcionais ligados ao PC, (Se existirem, passa o controle para eles, a fim de que inicializem), verifica se o disco está OK e prepara-se para carregar o DOS do disco.

A parte final do processo de start-up, é conhecida como "boot-strap loader". Trata-se de carregar o programa que se encontra no setor 0 do disco padrão, e se feito com sucesso, passar o controle para ele.

Como veremos depois, este programa que reside no setor 0, é quem se encarrega de carregar o resto do DOS do disco.

Observação: Nem sempre é o DOS que vai ser carregado. Por exemplo, no video-game "Night mission", e o próprio programa que é carregado. Ele roda sem a assistência do DOS. Pode ocorrer também de ser carregado o vírus, e depois deste rodar é que o DOS é carregado.

Para PCs originais da IBM, se houver erro na carga do registro de boot, automaticamente o sistema passa a carregar o BASIC residente. Para os PCs compatíveis (e que não tem o BASIC), o resultado é uma mensagem de erro, que varia de fabricante para fabricante. No MICROTEC, por exemplo, a mensagem é: "Coloque disco no DRIVE A, Aperte uma tecla".

Depois que o controle é passado para o registro de boot, a primeira coisa que este faz é verificar se o DOS está gravado neste disco. Isto significa procurar dois arquivos escondidos (de nomes IBMBIO.COM e IBMDOS.COM). Sendo encontrados, eles são carregados para a memória junto com o arquivo COMMAND.COM. O arquivo IBMBIO.COM contém extensões do ROM-BIOS. Essas extensões podem ser adições ou substituições nas operações básicas de E/S e podem incluir correções ao BIOS existente, novas rotinas para novos equipamentos etc. Desde que este programa reside em disco, ele é uma excelente maneira de alterar o BIOS. Tudo o que é necessário para alterar uma rotina do BIOS é alterar o vetor de interrupções colocando nele o novo endereço da nova rotina (que substitue a antiga).

O arquivo IBMDOS.COM contém as rotinas de serviços do DOS. Os serviços DOS, assim como os serviços do BIOS podem ser chamadas por nossos programas. As funções DOS oferecem mais sofisticação e um controle mais eficiente sobre as operações de E/S do que as rotinas do BIOS, principalmente para os disquetes. Todos os processos padrão para os discos (formatação, leitura e gravação de dados, abertura e fechamento, eliminação, pesquisa de diretórios) estão incluídas nas funções DOS e são chamadas por muitos programas de aplicação (tais como FORMAT, DIR etc). O arquivo COMMAND.COM contém as rotinas que vão interpretar as digitações fornecidas como respostas a "A>"ou "B>". Pela análise de uma tabela de nomes de comandos, elas vão poder analisar se o comando pedido é interno (tal como DIR, RENAME ou ERASE), ou se é externo (tal como FORMAT, DEBUG) ou se é um programa do usuário tal como RIMA181, ou SUPIMPA.

7.4 ROM-BIOS

As rotinas do BIOS providenciam os serviços fundamentais que são necessários para a operação do computador. Boa parte deles são os programas de controle dos periféricos (teclado, vídeo, discos etc). Estes programas traduzem um comando (como por exemplo, leia um determinado setor do disco), em todas as etapas necessárias para executar esta ordem, incluindo a exaustiva detecção de erros e sua correção. Conceitualmente, o BIOS está entre os programas (incluindo o DOS) e o hardware. Inicialmente os programas invocam o BIOS através de uma combinação entre o código de interrupção e o código de serviço desejado. O outro lado do BIOS conversa com o hardware solicitado, inclusive manuseando eventuais interrupções de hardware que os dispositivos gerem.

7.4.1 REGISTRADORES

O processador do micro foi planejado para executar instruções e realizar operações aritméticas e lógicas ao mesmo tempo em que recebe instruções e passa dados de e para a memória. Para fazer isto ele usa os registradores. Existem 14 registradores de 16 bits cada um no PC. Quatro são chamados "de dados"(AX, BX, CX e DX), Quatro são registradores de segmento (CS, DS, SS e ES), Cinco são de deslocamento (IP, SP, BP, SI e DI) e o último é chamado registrador de flag.

REGISTRADORES DE DADOS Quando o computador está processando dados, grande parte do tempo é gasto trazendo e levando dados de e para a memória. Este tempo de acesso pode ser muito reduzido se deixarmos os resultados mais freqüentes dentro do próprio 8088. Esta é a finalidade dos quatro registradores de dados. Cada um deles pode ser subdividido e separadamente acessado em dois meio registradores de 8 bits cada. Os meio-registradores de alta (high) ordem são conhecidos como AH, BH, CH e DH. Os de baixa (low) ordem são chamados de AL, BL, CL e DL. Por exemplo adições e subtrações podem ser feitas diretamente da memória, mas serão muito mais rápidas se usarmos os registradores. Embora estes quatro registradores estejam disponíveis para qualquer uso, os principais são:
AX: acumulações e registrador principal em cálculos aritméticos.
BX: geralmente usado para apontar para o início de uma tabela na memória. Também pode conter o deslocamento de um endereço segmentado.
CX: Contador repetitivo em loops. Por exemplo, a instrução LOOP do BASIC usa este registrador para

controlar o fim do loop.
DX: Uso geral

REGISTRADORES DE SEGMENTO CS: (code segment) aponta para o segmento que contem o programa em execução.
DS: (data segment) localiza o segmento de dados corrente.
SS: (stack segment) aponta para a pilha que contém endereços e parâmetros em uso pelo programa
ES: (extra segment) usado para suplementar o segmento de dados. Usado também para transferência de dados entre segmentos.

REGISTRADORES DE DESLOCAMENTO IP: (instruction pointer) localiza a instrução corrente dentro do segmento de código (CS). Também chamado de program counter (PC).
Os programas não acessam diretamente o IP, mas algumas instruções tal como JMP e CALL alteram o conteúdo de IP ou salvam-no e restauram-no a partir da pilha. SP (stack pointer) e BP (base pointer) apontam para segmento de pilha e para seu endereço dentro do segmento.
SI: (source index) e DI (designation index) freqüentemente usados com os registradores AX, BX, CX e DX para providenciar deslocamentos dentro da rea de dados, por exemplo, para transferir um string de um lado para outro da memória.

REGISTRADOR DE FLAG Conjunto de uma série de indicadores de um bit cada, chamados flags. Normalmente cada um dos indicadores é examinado sozinho e não em conjunto. Existem 9 indicadores em uso e 7 sem uso. São eles:
CF: carry flag - vai um
OF: overflow - estouro de campo
ZF: zero - resultado zero ou comparação igual
SF: sign - resultado negativo
PF: parity - paridade ímpar (número ímpar de uns)
AF: auxiliary carry - usado em operações decimais
DF: direction - (E->D ou D->E) em operações repetidas
IF: interrupt - determina quando uma interrupção pode ser feita
TF: trap - controla operações passo-a-passo (DEBUG)

7.5 Interrupções

Nos endereços 0 a 400 (hexa), está a tabela de vetores de interrupção. São 1024 bytes para conter até 256 interrupções (4 bytes cada, um endereço completo: 1 palavra para o segmento e outra para o off-set). O acesso é simples: Para a interrupção 0, o endereço de desvio está na posição zero. De um modo genérico o endereço para a interrupção "n"está na posição "n x 4".

Quando uma interrupção ocorre, o controle do computador é passado para uma rotina de manuseio de interrupções (programa) que está gravada em ROM. O manuseador de interrupções é chamado carregando seu endereço de segmento Quando uma interrupção ocorre, o controle do computador é passado para uma rotina de manuseio de interrupções que está em ROM. O manuseador de interrupções é chamado carregando seu endereço de segmento nos registradores que controlam o fluxo de processamento CS:IP (code segment:instruction pointer). Os endereços segmentados usados para localizar manuseadores de interrupção são chamados vetor de interrupções. O vetor de interrupções é pré-estabelecido durante o processo de start-up para apontar para as rotinas em ROM. Eles são estabelecidos em forma de uma tabela em RAM como um par de palavras representando os endereços de offset e de segmento. O vetor pode ser alterado para apontar para outras rotinas de manuseio, bastando:

1. Carregar a nova rotina em alguma posição de memória
2. Localizar o elemento (código) que a chama em V.I.
3. Alterar o conteúdo deste elemento apontando para 1).

Como regra geral, as interrupções da família PC podem ser divididas em 7 categorias: 1) de microprocessador (0,1,2,3 e 4)
2) de hardware (2, 8, 9 11 a 15)
3) de software (5, 16 a 28 e 72)
4) de DOS: (32 a 255)
5) de BASIC: enquanto o BASIC está em uso
6) de endereço: apontam endereços de tabelas (29 a 31, 68 e 73)
7) de uso geral: de 96 a 103

As interrupções de endereço, são diferentes na medida em que não h rotinas associadas a elas. Na realidade elas apontam para tabelas muito importantes na operação, por exemplo: vídeo initalization table, disk table base e graphics caracteres table.

PRINCIPAIS ROTINAS DE INTERRUPÇÃO DO PC		
Inter	Ender	Uso
0	0000	Gerado pela CPU quando h divisão por zero
1	0004	Etapa-por-etapa (DEBUG)
2	0008	Interrupção não mascar vel
3	000C	Estabelece break-points em programas (DEBUG)
4	0010	Overflow aritmético
5	0014	Invoca rotina de "print screen"
8	0020	Invocada por hardware: tique de relógio
9	0024	Ação de teclado
14	0038	Sinal de atenção do disquete
15	003C	Usado no controle de impressora
		----- serviços do BIOS ------
16	0040	de vídeo
17	0044	de lista de equipamentos
18	0048	de tamanho-memória
19	004C	de disquete
20	0050	de comunicação
22	0058	standard de teclado
23	005C	de impressão
24	0060	ativação de ROM-BASIC
25	0064	ativação de boot-strap start-up
26	0068	de data e hora
27	006C	Pressão de break (se rotina criada por nós)
28	0070	tique de relógio (idem)
29	0074	Aponta para a tabela de parâmetros de con-
		trole de vídeo
30	0078	Aponta para a "disk base table"
31	007C	Aponta para os caracteres expandidos do ASCII
		(em vídeo gr fico)
32	0080	Invoca serviço DOS de fim-programa
33	0084	Invoca todos demais serviços DOS
34	0088	Se criada, é invocada ao fim programa sob DOS
35	008C	Se criada, é invocada ao pressionar Break
36	0090	Se criada é invocada em erro crítico sob DOS
37	0094	Invoca serviço leitura absoluta de disquete
38	0098	invoca ... gravação
39	009C	Termina o programa, mas mantem-no na memória

7.6 RASCUNHO DO BIOS

A segunda parte desta área inicial da memória, é usada como rascunho pelo BIOS. Trata-se dos 256 próximos bytes (endereços 400 a 500 em hexa). A seguir algumas dessas posições e suas significações: 0410 - Palavra de lista dos equipamentos:

FEDCBA98 76543210	Descrição do significado
xx.....	Número de impressoras instaladas
..x.....	1 se impressora serial está instalada
...x....	1 se adaptador de joy-stick instalado
....xxx.	Número de portas seriais RS232 instalada
.......x	0 se o chip DMA está instalado
..... xx.....	+1: número de drives instalados00=1drive)
....xx....	Modo vídeo inicial (11=80 cols monocrom)
....xx..	RAM na system board (00=16K,01=32K,11=64K
....x.	Não usado (sempre 0)
....x	1 se existe algum drive de disco.

0413 - Memória us vel em Kbytes. O serviço 18 do BIOS devolve este valor. (2 bytes)

0417 - Controle de estado do teclado. Estado das teclas de controle do teclado do PC (2 bytes)

041A - Início buffer de teclado. Aponta para o início do buffer de teclado (que está em X"041E", mas é circular. - 2 bytes.

041C - Fim buffer teclado. Aponta para o fim do buffer de teclado. - 2 bytes.

041E - Buffer de teclado. Possui lugar para até 16 teclas (2 bytes cada), digitadas, mas ainda não lidas pelo BIOS (serviço 22).

043E - Precisa recalibrar. Indicador de necessidade de recalibração dos drives antes de operação de pesquisa. Os bits 0 a 3, correspondem aos drives A a D. 1 indica que não h necessidade de recalibração, e espera-se encontrar zero quando houve algum problema na última utilização do drive. Por exemplo: Quando se pede um dir e não h disquete, o DOS responde: NOT READY READING ERROR B: ABORT, RETRY IGNORE. Ao responder "A", o bit fica setado em zero. (1 byte)

043F - Motores dos drives. Indicador de estado dos motores dos drives. Bits 0 a 3 correspondem aos drives A a D. Se o bit é 1, o motor está ligado. (1 byte)

0440 - Contador de desligamento dos motores. O contador é setado em 37, ao início de cada operação em disco. A cada batida do relógio uma unidade é retirada. Ao chegar em zero, o motor é desligado. (1 byte)

0441 - Erros em disquete. Indicador de erros em disquete. 0 = OK, 1 = ERRO

Indicador de erros em disquetes	
76543210	Descrição do significado
x.......	Time out: Não respondeu no tempo
.x.....	Falha de pesquisa da trilha
..x....	Falha no chip controlador de disquete
...x...	erro de CRC (erro de dados)
....x..	Erro de DMA
.....x.	Sector not found: disquete defeit. ou não form.
......x.	Não encontrou marca de endereço
.......x	Requisição de comando de disquete inválida

0449 - Modo de vídeo. Modo corrente de vídeo: 1 byte

Byte 0449: Modo corrente de vídeo	
00	monocromático - 40 colunas - modo texto

```
| 01 | 16 cores - 40 colunas - modo texto                      |
| 02 | monocromático - 80 colunas - modo texto                 |
| 03 | 16 cores - 80 colunas - modo texto                      |
| 04 | Média resolução - 4 cores - modo gráfico                |
| 05 | Média resolução - monocromática 4 tonalidades-gráfico   |
| 06 | Alta resolução - 2 cores - modo gráfico                 |
| 07 | modo de adaptador monocromático                         |
| 08 | Gráfico de baixa resolução 16 cores não adaptador pad.  |
| 09 | Gráfico média resolução - 16 cores - idem               |
| 10 | Gráfico alta resolução - 4 cores - idem                 |
| 13 | Gráfico média resolução - 16 cores - idem               |
| 14 | Gráfico alta resolução - 16 cores - idem                |
| 15 | Gráfico especial alta resolução - 4 cores - idem        |
+----+---------------------------------------------------------+
```

044C - Tamanho da tela. Varia em função do modo. - 2 bytes.

044E - Deslocamento da tela. Endereço da página presentemente mostrada, dentro da memória de vídeo. 2 bytes

0450 - Localização do cursor. Para 8 páginas, começando na página zero. O primeiro byte de cada palavra indica a coluna e o segundo indica a linha (8 palavras)

046C - Contador de relógio mestre. A cada batida do relógio este contador é incrementado 1 unidade. Quando o contador atinge o equivalente a 24 horas, ele é reinicializado em zero, e o byte X"0470"é setado em 1. (2 palavras tratadas com inteiro de 4 bytes).

0470 - Indicador de "meia noite ultrapassada". É 1 quando isto acontece. Imagina-se que o programa que receber 1 aqui saber incrementar a data. (1 byte)

0471 - Pressão de break. Se o bit 7 é 1, indica a pressão de "break"(1 byte)

0472 - Reboot. Sinaliza um processo de "reboot"pelo teclado (CTRL + ALT + DEL). Quando isto acontece seu valor é X"1234".

0504 - Drive duplo. Usado pelo DOS quando um único drive é usado para simular dois drives. Quando é 00 indica que o drive é A, e quando é 01 indica que o drive é o drive B. (1 byte)

7.7 DISCOS MAGNÉTICOS

Discos são meios ideais para computadores armazenarem dados. Eles combinam baixo custo, a grandes capacidades, e velocidades significativas. A gravação se dá sobre uma superfície recoberta por óxido de ferro, e este material garante a permanência das gravações. A cobertura é colocada sobre um meio de suporte: plástico no caso dos disquetes e alumínio no caso dos discos rígidos. Na superfície, os bytes são armazenados como uma série de pontos magnéticos. Cada ponto corresponde a um bit, que como sabemos pode ter dois valores: zero e um. Não existe uma localização perfeitamente pré-determinada para estes pontos, razão porque é necessário colocar certas marcas de sincronismo. Isto é feito pela formatação. Por isso um disco não formatado não pode ser usado.

Quando colocado no drive (a unidade que lê e grava o disco), este começa a girar, expondo na janela de leitura e gravação toda a superfície de gravação. O disco gira rapidamente: um disquete a 300 rpm e um disco rígido a 3600 rpm. O disquete leva 1/5 segundo para dar uma volta completa e o disco rígido 1/60 de segundo para tanto. Na janela, existe um cabeçote que se desloca em sentido radial em relação ao disco, de maneira a poder atingir qualquer uma das partes de gravação. Ele funciona como se fosse um braço de toca discos. O movimento realizado pelo braço (se aproximando e se afastando do centro) é conhecido como movimento de "seek". Os tempos de seek também são pequenos: 1/6 segundo para o disquete e 1/25 para o winchester.

Se comparamos o disquete a um disco de música (um LP), poderemos ver algumas diferenças: 1. A reprodução no LP se dá por atrito, e nos discos de computador por magnetismo.

2. No LP a gravação se dá em uma única trilha, que se aproxima aos poucos do centro. No disco de computador, existem inúmeras trilhas paralelas e concêntricas.

As trilhas de um disco magnético estão numeradas desde 0 (a mais externa) até nn, dependendo do número de trilhas do modelo. Nos disquetes convencionais de 5 1/4", existem 40 trilhas. Nos disquetes de densidade quádrupla (do AT) existem 80 trilhas. Nos discos rígidos existem de 300 a 600 trilhas. Cada trilha (e todas elas) são divididas em partes chamadas setores. (imagine como fatias de uma torta redonda). O número de setores por trilha também varia. Originalmente no PC, eram 8 setores. Rapidamente este valor cresceu para 9. Nos disquetes de alta capacidade é, em geral, 15, e nos discos

rígidos da família PC, é 17. Os setores são numerados, e o micro reconhece onde a numeração começa pelo furo de referência de início de trilha. O setor contém um número finito e fixo de bytes.

Existem modelos cujo setor era de 128, 256 e até 1024 bytes, entretanto o setor de 512 bytes acabou se tornando padrão, principalmente em nosso País. O setor atua também como unidade de transferência de dados entre a UCP e o drive. Isto é, sempre que nosso programa quiser ler alguma coisa, um número inteiro de setores ser lido. Mesmo que o programa necessite apenas de poucos bytes, sempre ser lido pelo menos um setor. Identicamente para a gravação. Outra dimensão importante para os discos magnéticos é o número de lados: Nos disquetes este valor pode ser um ou dois. No PC original, surgiram alguns drives que só tinham uma cabeça de gravação, e portanto só usavam um dos lados do disco. Fazia-se uma pequena economia de uma cabeça de gravação ao custo de dividir por dois a capacidade de armazenamento do disco: não foi uma boa idéia. Rapidamente, estes drives deixaram de existir, e passou-se a usar os dois lados dos disquetes, embora por compatibilidade, os drives ainda conseguem ler e gravar em face simples.

A quantidade de faces e de trilhas, é uma característica que j vem predeterminada nos discos e nos drives. Entretanto, a quantidade e localização de setores é uma característica determinada por software (no caso pelo programa FORMAT). Esta é a razão pela qual estes disquetes são conhecidos como "setorizados por software". Os discos rígidos têm um número maior de lados. Este fato se explica pelo fato do que chamamos "disco rígido"ser na realidade um conjunto de v rios discos (estes sim, chamados "platters"). Tanto é que aqui surge um novo conceito, o de cilindro. Um cilindro é o conjunto de todas as trilhas (uma em cada platter) que tem mesmo número, ou seja que ocupam lugares correspondentes. Em resumo, após nossa discussão sobre disquetes, chegamos ao seguinte quadro:

Características de discos na família PC					
Disco	Lados	Trilhas	Setores	bytes/setor	Tam.
S8 (dos1.0)	1	40	8	512	160K
D8 (dos1.1)	2	40	8	512	320K
S9 (dos2.0)	1	40	9	512	180K
D9 (dos2.0)	2	40	9	512	360K
QD9(dos3.0)	2	80	9	512	720K
QD15(AT)	2	80	15	512	1200K
XT	4	306	17	512	10M
AT	4	615	17	512	20M

7.7.1 ORGANIZAÇÃO DO DISCO

Como já vimos, todos os elementos de um disco são identificados. Por exemplo, num disquete face dupla com 9 setores nós temos: as faces (numeradas como 0 e 1), as trilhas (numeradas de 00 a 39), e os setores (numerados de 1 a 9). Quando BIOS vai ler um setor, ele é localizado pelas suas coordenadas: número da trilha (ou do cilindro), número da face (ou da cabeça), e número do setor. J o DOS reconhece cada setor pelo seu número seqüencial dentro do disco. Assim, num disquete temos:

Visões do BIOS e do DOS sobre discos	
Visão do BIOS	Visão do DOS
Lado=0, trilha=0, setor=1	Setor=0
Lado=0, trilha=0, setor=2	Setor=1
...	...
Lado=0, trilha=0, setor=9	Setor=8
Lado=1, trilha=0, setor=1	Setor=9
...	...
Lado=0, trilha=1, setor=1	Setor=18
...	...
Lado=1, trilha=39, setor=9	Setor=719

Atente-se que esta visão seqüencial do disco simplifica bastante o trabalho do sistema operacional, mas tem seus inconvenientes: O primeiro é que o DOS não sabe tomar partido do fato de que processando setores em um mesmo cilindro ganha-se muito mais tempo, do que processando-os em cilindros diferentes. Neste caso o DOS desconhece o que sejam cilindros. Outro problema é que o DOS usa inteiros de 16 bits para identificar os setores. Como sabemos o maior número armazenável em uma palavra é 65.536. Como cada setor tem 512 bytes, ficamos limitados a endereçar 512 x 65536 = cerca de 32 milhões de bytes). Este portanto, é o limite atual de discos rígidos suportáveis pelo DOS. Na época de criação do PC tal limite parecia inatingível por muitos e muitos anos. Nem dez anos se passaram, e ele j foi esgotado. Isto parece dar razão a Peter Norton: "Não importa quanto você tenha, não é o bastante".

FORMATAÇÃO FÍSICA E LÓGICA O processo de formatação de um disco se dá sempre em duas etapas, cujos nomes são formatação física (anterior) e lógica (depois daquela). Para um disquete, ambos os processos se confundem e o comando FORMAT executa as duas. As diferenças começam a surgir no caso dos discos rígidos.

A formatação física envolve a criação dos setores, a colocação das marcas de endereço, e a gravação (e posterior leitura) de dados fictícios. J a formatação lógica é a adaptação do disco aos padrões do sistema operacional no qual ele ser usado. No nosso caso, geralmente é o DOS.

Este conceito (formatação física e lógica separadas), existe para permitir o compartilhamento de um disco rígido por dois ou mais sistemas operacionais diferentes. Outra razão, é o limite de 32Mb discutido acima. Se tivermos um disco maior, precisaremos criar duas ou mais regiões nele. (Dividir o disco em regiões é como quebrar o disco em v rios discos menores). No disquete, tal dificuldade não existe, pois o disquete sempre pertence integralmente ao sistema operacional que o formatou, isto é, a formatação lógica e física são uma só coisa.

No disco rígido, nós podemos ter a criação de diversas partições, que vão funcionar como regiões específicas para sistemas operacionais diversos. Por exemplo, podemos ter em um disco rígido uma partição para uso do DOS e outra para uso de, digamos, SOX. Quem cria as partições é o utilitário chamado FDISK. Ele deve ser chamado mesmo que desejemos alocar o disco inteiro para o DOS (caso mais comum). Neste caso, precisamos criar uma única partição e oferecê-la ao DOS. Depois de criadas as partições, cada uma delas deve sofrer a ação do formatador (FORMAT no caso do DOS) específico para cada sistema.

Podemos trocar a qualquer momento o número e o tamanho de nossas partições, entretanto tal procedimento destrói os dados anteriormente existentes. Discos particionados têm no seu primeiro setor, (junto com o programa master boot), uma tabela de partições contendo: o tamanho e localização de cada uma, o tipo, qual está ativa, etc. J o programa master boot identifica qual a partição ativa, e passa o controle para os procedimentos do boot desta partição.

MAIS SOBRE FORMATAÇÃO A formatação é feita uma trilha (de um lado) a cada vez. Desta maneira, não podemos formatar um setor individualmente, embora possamos criar setores com diferentes tamanhos e ordens. Cada setor dentro de uma trilha, tem quatro bytes descritivos. Eles estão logo após a marca de endereço, que é usada mais tarde para identificar setores individuais em operações de leitura, gravação e verificação. Os quatro bytes de endereço, são conhecidos como C (cilindro, ou trilha), H (head, cabeça ou lado do disco), R (registro, ou número do setor) e N (código de tamanho do setor).

Tamanhos padrão para o código de tamanho de setores		
N	Tamanho setor (bytes)	Tamanho setor (K)
0	128	1/8
1	256	1/4
2	512	1/2
3	1.024	1

Quando um setor está sendo lido ou gravado, o BIOS procura os endereços de setores dentro da trilha, nos quais a parte essencial é a informação R (record, ou número do setor). Os parâmetros C e H não são mais necessários nesta marca de endereço, pois o braço já foi posicionado mecanicamente sobre a trilha, e a face também já foi selecionada eletronicamente, mas estes dados estão gravados e são verificados para segurança. Os setores são gravados no disquete na ordem especificada pelos bytes de endereço, e esta

não é necessariamente seqüencial. Os setores podem ser intercalados (interleaved), com finalidades de melhoria de performance e com propósitos de proteção contra cópia. Os disquetes têm os setores gravados em rigorosa ordem ascendente. Em um disquete convencional de nove setores do DOS, os endereços de formatação, para a trilha zero, face 1 seriam:

```
C H R N     C H R N     C H R N     ...     C H R N
0 1 1 2     0 1 2 2     0 1 3 2             0 1 9 2
```

Quando a trilha está sendo formatada, o drive presta atenção ... passagem do buraco índice no disquete. A partir dele são colocadas as marcas de formatação da trilha. O buraco é ignorado em todas as demais operações (leitura, gravação e verificação), pois nestas, as trilhas são localizadas por suas marcas de endereço.

FORMATAÇÃO USADA COMO PROTEÇÃO CONTRA CÓPIA Trilhas podem ser formatadas de diversas maneiras, mas a maioria dos sistemas operacionais podem ler apenas certos formatos. Conseqüentemente, muitos esquemas de proteção são baseados em formatações não convencionais, e tentam impedir a cópia e gravação usando sistemas operacionais. Eis alguns métodos:

1. A ordem dos setores pode ser rearranjada. Este fato determinar mudança no tempo de acesso, e este fato pode ser detectado pelo programa.
2. Pode-se ”apertar”mais setores em uma única trilha (10 é o limite aproximado de setores de 512 bytes).
3. Pode-se omitir o número de um setor.
4. Pode-se gravar um setor com número disparatado (por exemplo 43).
5. Pode-se especificar um ou mais setores com tamanho não convencional.
6. Pode-se gravar a marca de endereço com valores errados para C e H.

ESQUEMA DE PROTEÇÃO DO WORD

O programa MS-WORD na sua versão 2.0 utiliza um esquema de proteção baseado nas idéias aqui vistas. Inicialmente, existem 3 arquivos dentro do disco do WORD que têm byte de atributo igual a X’06’, que significa ”hidden file”e ”system file”. Uma segunda proteção existe na trilha 39, face 1 (segunda). Esta trilha encontra-se formatada de maneira não standard. Em vez de 9 setores, ela tem 12, 11 dos quais com tamanho de 256 bytes e um deles (o quinto) com tamanho de 1024 bytes. Nestes setores, existem além de algumas constantes, um discurso contra os piratas. Na FAT, os 5 clusters referentes a esta trilha/face são apresentados como defeituosos.

7.7.2 DISK BASE TABLE

Todas as operações do drive de disquetes são controladas por um conjunto de parâmetros chamados de ”disk base table”. Embora exista uma tabela default gravada em ROM (na localização F000:EFC7), pode-se criar uma nova versão dela. Para fazer isto basta colocá-la em qualquer parte da memória e alterar o vetor de interrupções (vetor número 30, hexa 1E) para apontar para ela. Todas as versões de DOS, depois da 1.00 criaram suas próprias tabelas diferentes da existente em ROM. A tabela é composta por 11 bytes, a seguir mostrados. Muitas das informações aqui mostradas têm pouco uso, exceto talvez se se planeja criar uma nova tabela em substituição àquela usada pelo DOS.

Conteúdo da DISK BASE TABLE	
Desloc	Uso
0	Byte específico 1: Step rate time/head unload time
1	Byte específico 2: head load time/DMA mode
2	Espera até desligar o motor do drive
3	Bytes por setor (0=128, 1=256, 2=512, 3=1024)
4	Último número de setor
5	Comprimento do "gap" entre setores (read/write)

6	Comprimento dos dados, quando setor não tiver comp
7	Comprimento do "gap" entre setores (formatação)
8	Valor gravado nos setores formatados
9	Tempo de ajuste da cabeça
A	Tempo de start-up do motor

Bytes 0 e 1: Estes bytes são conhecidos como bytes específicos. Eles fazem parte dos comandos mandados para o controlador do disquete (floppy disk controller). O SRT é o tempo que o ROM-BIOS espera que o drive mova o cabeçote de uma trilha para outra. Valor default: 8 milissegundos. O DOS 2.10 reduziu este tempo para 6 milisseg.

Byte 2: Especifica quanto o motor deve ficar ligado após o uso. A unidade é tiques de relógio (aproximadamente 18 por segundo). Em geral existe 37 (hexa 25) que corresponde a 2 segundos.

Byte 3: O mesmo código N usado na descrição dos setores nas trilhas.

Byte 4: Valor default do último setor nas trilhas (8 na ROM e 9 na tabela do DOS 2.10).

Byte 5: Intervalo entre setores para operações de leitura / gravação. Trata-se de uma dica para o BIOS saber quanto deve esperar até procurar a próxima marca de endereço de setor. Usualmente é 42 (hexa 2A).

Byte 6: Data transfer length (DTL). Indica quantos dados serão transferidos quando o setor não tiver tamanho especificado (setado para 255 (hexa FF).

Byte 7: Intervalo entre os setores para operação de formatação. Naturalmente deve ser um numero maior do que aquele descrito no byte 5. Valor normal: 80 (hexa 50).

Byte 8: Estabelece o byte a ser gravado em todos os setores. Originalmente é F6, o símbolo da divisão.

Byte 9: Especifica quanto tempo o sistema deve esperar até que a cabeça se estabilize depois de ler uma trilha. O tempo default é 25 (hexa 19) milissegundos, mas o DOS 2.10 baixou para 15 (hexa 0F) milisseg.

Byte 10: Tempo que o DOS espera a fim de que o disco acelere e possa ser operado. A unidade é 1/8 segundos. Default é 4 (1/2 segundo), mas o DOS 2.10 baixou para 2 ou 1/4 de segundo.

7.8 ÁREAS NOS DISCOS

Quando o DOS formata um disco, ele cria quatro áreas distintas, a saber:

A área de boot Sempre é um único setor, localizado no lado 0, trilha 00, setor 1. É um pequeno programa, escrito em linguagem de máquina, capaz de iniciar o processo de carga do DOS na máquina. Todos os disquetes contém este registro, MESMO que não tenham o DOS gravado neles. Antes de fazer qualquer coisa, este programa verifica a existência dos arquivos IBMBIO.COM e IBMDOS.COM .

Além do programa de boot, encontraremos alguns parâmetros nesta área, a saber:

Parâmetros encontrados no registro de boot

Desl	Tamanho	Descrição
0	2	Jump para programa BOOT (JMP 002E)
3	8	Identificação do sistema (ex.: IBM 2.1)
11	2	Número bytes por setor (ex.: 512, ou H"0020"
13	1	Número setores por cluster (ex.:01 ou 02)
14	2	Número setores reservados no início: 1 p/isq
16	1	Número de cópias da FAT: 2 para disquetes
17	2	Número entradas no diretório raiz (64 ou 112
19	2	Número total setores no disco (720 p/ D9)
21	1	Identificação de formatação (FF, FE, FD, FC
22	2	Número de setores por FAT (1 ou 2)
24	2	Setores por trilha (ex.:8 ou 9)
26	2	Número de lados (ex.: 1 ou 2)
28	2	Número de setores especiais reservados

A identificação de formatação, pode ser: FF (D-8), FE (S-8), FD (D-9), FC (S-9), F9 (QD-9 ou QD-15).

7.8.1 FAT (File alocation table)

Esta área segue o registro de boot, usualmente no setor 2 da trilha 0, face 0. Trata-se de um mapa de uso das áreas do disco, e suas alocações para arquivos específicos. Seguindo o conceito de mapa, a FAT reproduz em pequena escala, um desenho de todos os espaços do disco. A unidade aqui é o cluster - ou grânulo, em português - (unidade de alocação de arquivos = 2 ou mais setores). Cada entrada da FAT aponta para um cluster, e dependendo do código colocado na FAT para um cluster específico podemos saber o estado daquele cluster: desocupado, ocupado por algum arquivo, defeituoso, etc.Devido ... importância da FAT (na realidade é a rea mais importante do disco, no que diz respeito ... sua integridade), a FAT sempre é mantida em duas cópias iguais, e teoricamente podemos usar a segunda para recuperar a primeira, em caso de perda. Entretanto, não saberemos quando elas estão diferentes. CHKDSK, por exemplo, não informa isso. As FATS usam as seguintes áreas:

```
S-8 ou D-8: 2 setores (ambas as cópias)
S-9 ou D-9: 4 setores (ambas)
QD-9      : 10 setores (")
QD-15     : 14 setores (")
```

As FATS para discos rígidos dependem dos tamanhos das partições.

Como vimos acima, cada cluster no disco, possui uma entrada na FAT que estabelece seu "status". O cluster é identificado por um número seqüencial, que começa em dois (0 e 1 reservados).

Distribuição de setores e clusters em disquetes

Formato	Setores	Setores p/ cluster	Cluster	Numeração
S-8	313	1	313	2 a 314
D-8	630	2	315	2 a 316
S-9	351	1	351	2 a 352
D-9	708	2	354	2 a 355
QD-9	1.422	2	711	2 a 712
QD-15	2.371	1	2.371	2 a 2372

A FAT é composta por números binários, como segue: para discos menores cada entrada na FAT ocupa 12 bits, e para discos maiores, como o winchesters de 20Mb, ocupa 16. Embora tais diferenças impliquem diferentes formas de tratamento por parte do DOS, conceitualmente ambas FATs funcionam igual. Devido a sua característica, tudo que um elemento da FAT pode armazenar é um número. Lembramos que cada elemento da FAT corresponde um cluster (de mesmo número) localizado na rea de dados do disco. A interpretação deste número é:

ZERO: Quando um elemento da FAT contém zero, significa que o cluster correspondente está livre, e pode ser alocado a qualquer necessidade.

NÚMERO QUALQUER: Significa que o arquivo que está gravado no cluster equivalente a esta entrada na FAT, CONTINUA no cluster indicado pelo número qualquer aqui colocado.

FFF (ou FFFF): Indica fim de cadeia, ou seja, este cluster é o último ocupado pelo arquivo.

FF7 (ou FFF7): Indica cluster defeituoso. Ele foi detectado pelo programa FORMAT (leu-se algo diferente do que se gravou), e o próprio FORMAT estabeleceu esta entrada na FAT, a fim de que este cluster jamais seja alocado.

FFx (ou FFFx): Usos reservados para o futuro.

O conceito que está por trás da FAT é simples e engenhoso. Para um determinado arquivo, sua entrada no diretório (veremos a seguir) indica qual o cluster inicial que o compõe: digamos por hipótese, que seja o número 70. Os dados começam a ser gravados no cluster 70. Já na entrada 70 da FAT, o DOS coloca o código de continuação: FFFF (se este é o primeiro e ÚLTIMO cluster do arquivo), ou um outro número qualquer diferente de zero, digamos 49. Isto significa que o arquivo continua no cluster 49, e a entrada 49 contém a continuação: FFFF para FIM, ou outro número qualquer, e assim por diante. O tamanho de cada elemento da FAT, e o comprimento da FAT impõe limites na quantidade de clusters a endereçar. Assim, uma FAT de elementos de 12 bits, teoricamente, pode endereçar até 4095 clusters. (de X"000"a X"FFF").

Exemplo de uma FAT Antes de analisarmos a FAT, para exemplificar, vamos imaginar a existência de dois arquivos no diretório do disco, chamados PRIMEIRO.ARQ, e SEGUNDO.ARQ. Eles iniciam nos cluster 7 e 4 respectivamente. Visto isto, vamos à FAT:

Exemplo de uma FAT			
Entrada na FAT	Conteúdo Decimal	Conteúdo Hexad	Significado
0	253	FD	Disco tipo S-9
1	4094	FFE	Entrada não disponível
2	0	0	Cluster livre
3	4087	FF7	Cluster defeituoso (bad track)
4	4095	FFF	O arquivo SEGUNDO.ARQ termina aqui
5	0	0	Cluster livre
6	0	0	Cluster livre
7	8	8	Arquivo PRIMEIRO.ARQ continua em 8
8	10	10	Continua em 10
9	0	0	Cluster livre
10	4095	FFF	Arquivo PRIMEIRO.ARQ termina aqui
11	0	0	Cluster livre.
...	...	...	...

Para encerrar o assunto referente a FAT, devemos considerar a possibilidade de ocorrência de erros dentro dela. Podemos ter alocações circulares, por exemplo o primeiro cluster de um arquivo é o 5, o quinto elemento da FAT aponta para 7, e o sétimo elemento da FAT aponta para 5. Outro erro possível, é a convergência de clusters: Dois arquivos distintos dizem que determinado cluster lhes pertence, ou o contrário, um cluster órfão, de quem ninguém assume a paternidade. Quando isto ocorrer, podem ser utilizados os programas CHKDSK e RECOVER que costumam recuperar a maioria destas falhas.

Área de diretório raiz É a próximo área do disco. Trata-se de uma tabela de conteúdo do disco, com nomes e demais informações de cada um dos arquivos que existem no disco. Seu conteúdo é próximo daquele visto num comando DIR.

É uma tabela, em que cada elemento possui um comprimento de 32 bytes. Desta maneira podemos ter 16 entradas por setor de diretório. Discos D-9 têm 7 setores reservados para diretório raiz, razão pela qual estão limitados a 112 entradas (7 x 16).

Em geral, estas entradas equivalem cada uma a um arquivo. Entretanto, como veremos, podemos ter entradas para identificação de volume, e entradas para sub-diretórios, que neste ponto são tratados como arquivos quaisquer. A entrada de 32 bytes está assim organizada:

Entrada de um arquivo no diretório			
Campo	Desloc.	Tamanho	Descrição
1	0	8	Nome
2	8	3	Extensão
3	11	1	Byte de atributos (veja abaixo)
4	12	10	Reservado para futuro
5	22	2	Hora codificada
6	24	2	Data codificada
7	26	2	Cluster inicial do arquivo
8	28	4	Tamanho do arquivo

Campo 1: Nome Trata-se de um campo armazenado em formato ASCII. Se o campo tem menos de 8 caracteres, ele é preenchido com espaços (X"20"). As letras devem aparecem sempre em maiúsculas, uma vez que as minúsculas nem sempre são corretamente manuseadas. Normalmente, não deveremos ter

brancos incluídos dentro do nome (tal como ABC DEF). A maioria dos programas DOS (tal como DEL ou COPY) não aceitam tal formato. Entretanto, o BASIC consegue criar tais arquivos e as chamadas DOS também operam sem problema. Esta característica, nos sugere a facilidade de criarmos arquivos não facilmente deletáveis ou copiáveis. Três códigos especiais podem aparecer no início deste nome. Se existir um zero binário (X"00"), isto significa que esta entrada jamais foi utilizada. Significa também que não existe mais nenhuma entrada usada abaixo desta. Assim o DOS pode encerrar uma pesquisa de diretório neste ponto. Se existir um X"E5", isto significa que o arquivo foi "deletado". Quando isto acontece, a única alteração que ocorre, além do nome, é a liberação de toda a cadeia da FAT anteriormente alocada. Todas as outras informações referentes ao arquivo são preservadas. Este fato permite a possível recuperação de arquivos "deletados", desde que sua entrada no diretório não tenha sido reusada por outro arquivo. O DOS sempre procura (ao alocar um arquivo) usar as primeiras entradas do diretório que encontra livres. O terceiro código que pode aparecer é o ponto (X"2E"), que especifica um subdiretório. Se o segundo byte é também um ponto, nós estamos olhando para o "pai"do subdiretório corrente. Neste caso, o campo 7, referente a este arquivo, é o cluster que contém o diretório pai. Em outras palavras, quando um subdiretório é criado, automaticamente são criadas nele duas entradas: A primeira ".", tem o seu cluster inicial apontando para a rea de dados onde este subdiretório realmente começa. A entrada ".."refere-se a localização do diretório "pai"deste. Quando o cluster inicial, neste caso, é zero, significa que o diretório "pai"é o raiz.

Observação: No diretório, todos os nomes de arquivo estão em MAIÚSCULO. Se você entrar direto (via norton) no diretório e escrever arquivos em minúsculo, estes arquivos não poderão ser manuseados. (eliminados, copiados etc).

Campo 2: A extensão É a continuação do nome. Pode estar totalmente em branco. A única observação é que quando uma entrada se referir a "volume-label", este campo ser conjugado com o anterior, formando um campo total de 11 bytes. Neste caso, brancos podem ser incluídos no meio ... vontade.

Campo 3: Atributos de arquivo Este campo é composto de 8 bits, cada um deles sinalizando uma condição específica.

Byte de atributos de arquivos no diretório			
7 6 5 4 3 2 1 0	Dec	Hex	Descrição
. 1	1	1	Arquivo "read-only"
. . . . . 1 .	2	2	Arquivo escondido "hidden"
. . . . . 1 . .	4	4	Arquivo "system"
. . . . 1 . . .	8	8	Identificação de volume
. . . 1	16	10	Subdiretório
. . 1	32	20	Arquivo alterado
. 1	64	40	Reservado para futuro
1	128	80	Reservado para futuro

Bit 0 - Arquivo apenas de leitura. Trata-se de uma proteção. Desde que um arquivo tenha este bit setado, ele está protegido contra operações de alteração ou deleção do DOS.

Bit 1 - Arquivo escondido. Quando um arquivo tem este atributo, ele se torna invisível para comandos DOS tais como COPY, DIR, DEL etc (mas estranhamente não para o comando TYPE).

Bit 2 - Arquivo "system". Este atributo é idêntico ao anterior, e só permaneceu no DOS por compatibilidade com o antigo CP/M.

Bit 3 - Identificação de volume. Este atributo indica que a presente entrada não se refere a um arquivo e sim a uma identificação de volume. Neste caso, o nome e a extensão são tratadas como uma coisa só. Este atributo só pode aparecer no diretório raiz. Quando este bit está ligado, as informações de tamanho e cluster inicial não são consideradas, mas as de data e hora sim.

Bit 4 - Atributo de subdiretório. Um subdiretório, neste caso é considerado como arquivo. Todos os campos são preenchidos com exceção do campo de tamanho, que é zero. Assim o tamanho do subdiretório é obtido percorrendo-se até o fim a ligação da FAT.

Bit 5 - Atributo de alteração. Este critério foi incluído para ajudar nas tarefas de backup de discos rígidos. O bit está desligado para todos os arquivos que não tiveram modificações desde o último back up. Este bit normalmente é um para todos os arquivos em disquetes.

Campo 4: Reservado. Esta área de 10 bytes está reservada para possíveis usos futuros e normalmente contém hexadecimais zero (X"00").

Campo 5: Hora. Contém o horário em que o arquivo foi criado ou alterado pela última vez. Este campo atua com o próximo, e os dois juntos podem ser considerados como um campo inteiro de 4 bytes. É assim obtida: Horário = Hora x 2048 + Minutos x 32 + Segundos / 2

Assim, o horário 20:05:10, seria assim representado:

```
20 x 2048 + 5 x 32 + 10 / 2
40960     + 160    + 5     =     41125
```

Campo 6: Data Contém a data em que o arquivo foi criado ou alterado pela última vez. O maior ano suportado pelo DOS ser 2099, e o menor 1980. A fórmula: Data = (Ano - 1980) x 512 + Mês x 64 + Dia

Assim, a data 9 de maio de 1988, ficaria:

```
(1988 - 1980)  x  512  +  5  x  64  +  9
           8   x  512  +  320       +  9
           4096       +  320       +  9     =    4425
```

Campo 7: O número do cluster inicial. Indica qual o primeiro bloco de dados para este arquivo. Indica também em que posição da FAT deve-se procurar a sua continuação. Para identificações de volume e arquivos que não têm espaço alocado, este valor é zero.

Campo 8: Tamanho do arquivo. Até o ponto em que o DOS pode afirmá-lo, este é o tamanho real do arquivo. É um campo de 4 bytes, inteiro e sem sinal, sendo portanto capaz de guardar tamanhos muito maiores do que aqueles proporcionados pelo hardware atual.

7.8.2 A área de dados

Todas os arquivos e subdiretórios residem na rea de dados (que é a maior do disco). O espaço em disco vai sendo alocado aos arquivos à medida em que vai sendo necessário, sempre à razão de um cluster a cada vez. Seria de se esperar que os arquivos ocupassem lugares contíguos, entretanto nem sempre é o que acontece: Quando um arquivo já existe, e é estendido, os locais após ele, podem ter sido já ocupados. Neste caso, ele vai para outras posições não contíguas. Quando um arquivo, de digamos 20K, ocupa, ao ser gravado, a posição que era de um arquivo anterior (que tinha 10Kb) e que foi eliminado. Neste caso os 10K iniciais entram aí, e o resto vai para as próximas reas disponíveis.

A quantidade de fragmentação de um arquivo pode degradar seu tempo de acesso. Também pode ser muito difícil recuperar um arquivo anteriormente apagado, se este estava muito fragmentado. Em geral nestes casos, só se consegue recuperar seu início. Fora estes dois casos, não h maiores problemas na segmentação de arquivos. Em geral os programas não sabem (e não precisam saber) em que lugares estão seus dados, nem qual sua fragmentação.

Uma maneira de ver a fragmentação, poderia ser através de uma função do programa Norton Utilities. A forma de corrigir fragmentações seria através da cópia dos arquivos para discos vazios. Nesta operação os arquivos entram de maneira seqüencial; sem fragmentação.

Tabela de formatos de discos e seus conteúdos

Formatos de discos e seus conteúdos (Boot,FAT,dir,dados)						
		------ setores controle ------				
Formato	Tot setores	Boot	FAT	Diretório	Total	Dados
S-8	320	1	2	4	7	323
D-8	640	1	2	7	10	630
S-9	360	1	4	4	9	351
D-9	720	1	4	7	12	708
QD-9	1.440	1	10	7	18	1422
QD-15	2.400	1	14	14	29	2371

8 CICLO DE VIDA DO VÍRUS

O vírus tem um ciclo de vida bastante característico, que o diferencia dos softwares aplicativos que nós usamos no dia a dia.

1. PROJETO & PROGRAMAÇÃO & IMPLEMENTAÇÃO Como qualquer software, o vírus deve ser planejado. Uma lista de requisitos e maneiras de operação deve ser elaborada. Se atentarmos para o fato de que vírus são softwares de altíssima qualidade, veremos que o seu projeto e programação devem ser também de alto nível. O software deve ser testado, depurado (sic) e seus erros devem ser retirados. No dia em que o produto estiver OK, o vírus pode ser considerado pronto.

2. ATIVAÇÃO Esta fase não é obrigatória, mas tem sido relatada para alguns vírus. Trata-se de uma etapa que só ocorre se o vírus detectar que se encontra grudado a uma cópia pirata de um software qualquer. Este teste (se positivo) funciona como uma chave de liberação do vírus. Isto significa que enquanto ele permanecer "grudado"ao software original, ele permanece inativo.

3. REPLICAGEM OU CONTÁGIO Quando o vírus está pronto, começa a ser espalhado, no princípio por atitudes diretas de seus autores, mas logo depois por ações inocentes de usuários que nada desconfiam. Os mecanismos de contágio são os seguintes:

a) Disquete bom em micro doente Neste caso, um micro previamente contaminado (e com o vírus residente e ativo na memória) infectar todos os disquetes que sejam colocados no micro, desde que cuidados não sejam tomados.

b) Disquete doente em micro bom O disco que saiu da etapa (a) acima, quando colocado em um micro são, fará com que o vírus reviva, se aloje na memória, e passe a infectar todos os demais discos.

As etapas (a) e (b) acima, explicam o surgimento de verdadeiras epidemias em questão de dias. De acordo com os costumes "higiênicos"das organizações, a rapidez de infestação pode ser muito alta.

c) Cópias de disquetes Ao copiar um disquete doente, o micro em si não se contamina, mas o resultado da cópia, também pode estar contaminado, isto é, o vírus vai junto na cópia. Se for um vírus do tipo "boot sector infector"(vide adiante), ele só ir se for feita uma cópia física (tipo COPYWRIT ou COPYIIPC). Entretanto se for um "program infector"ele ir mesmo através de um simples COPY *.*.

d) Bulletin Board Systems Estas redes, caracterizam-se pelo acesso público e livre, praticamente sem nenhuma restrição. Estes serviços permitem extensa troca de programas, sem problemas de ordem legal, moral ou ético. Isto facilita a disseminação de vírus. Nos EUA existem mais de 3000 redes deste tipo, muitas delas de alcance nacional, e lá, este tem sido um dos mecanismos mais atuantes na disseminação de "vírus". Já no Brasil, existem poucas redes. A primeira foi o CIRANDÃO, que foi muito usado enquanto as tarifas de telefonia envolvidas eram subsidiadas. Hoje, quando os preços são reais, o uso tem diminuído bastante. Além do cirandão, que alias até mudou de nome, existem poucas redes, (certamente menos de 10), e de uso restrito e abrangência local. Por este fato, este mecanismo de infecção tem sido pouco importante em nosso País.

Um aspecto parecido ao dos BBS é o das redes particulares. Embora o potencial de disseminação esteja restrito a um único usuário, (geralmente o dono da rede), o mal causado pode ser maior, por não se imaginar que uma rede deste tipo esteja contaminada.

e) Transmissões através de ligações remotas. A transmissão de um arquivo contaminado leva a contaminação, de maneira similar a um COPY local.

4. ATAQUE Depois de um certo tempo, o vírus repentinamente "acorda"passando a causar todo o potencial de malefício que ele carrega. Esta fase se justifica, pois se o vírus atacasse no momento em que fosse copiado, ele imediatamente alertaria os usuários, fazendo com que sua disseminação fosse muito difícil, se não impossível. (Se todos os doentes de AIDS, tivessem os cabelos milagrosa e instantaneamente pintados de cor azul, no momento do contágio, a epidemia diminuiria até acabar...)

Teoricamente, quanto mais tempo se passar entre as fases 3 e 4 mais perigoso é o vírus, pois mais indefeso estar o ambiente no qual ele se insere.

Esta fase é conhecida como o GATILHO do vírus. Devemos atentar que das 4 fases do programa, ele só é percebido na quarta e última.

Usualmente o gatilho está associado ao relógio, tanto no aspecto de data quanto no de hora. Outros vírus, testam partes da memória até encontrar uma configuração adequada, e este é o seu gatilho. Este fato tem levado usuários de PC mais radicais, a eliminarem os relógios de seus PCs (o autor conhece uma pessoa que fez isto), abrindo mão da facilidade de datar tudo que ocorre no PC, e passando a imaginar que tudo é feito no dia 01/01/80.

A propósito, o vírus mais espalhado em nosso País é o temível ISRAELI, que ataca os programas executáveis nos dias em que é sexta feira, 13. Uma lista dos próximos dias nesta situação: 13/10/89, sexta 13/04/90, sexta 13/07/90, sexta

Alguns vírus têm mais de um efeito, que se acumulam na operação normal do programa. Por exemplo, o ISRAELI, além de matar os arquivos executados em dias 13 sextas, após um período de aproximadamente 30 min torna a máquina lenta.

Outros exemplos famosos, foram a mensagem de natal que apareceu em todos os mainframes de uma rede americana, no dia 24 de dezembro, e também o "parabéns para você"que o McIntosh II cantou para sí próprio no dia de seu aniversário.

5. MÚLTIPLAS VERSÕES Para encerrar o ciclo de vida, devemos lembrar que embora seja difícil escrever um vírus de sucesso, é simples alterá-lo para que seus efeitos, ou seu gatilho sejam ligeiramente diferentes. Assim, nem sempre os vírus fazem o que aparentam fazer. Um vírus modificado (podemos chamar isto de mutação?) pode entrar em um novo ciclo de vida, passando novamente por todas as fases.

9 TAXONOMIA (classificação)

Em uma primeira abordagem, os vírus se dividem em duas categorias: benignos e malignos.

a) VÍRUS BENIGNO: É aquele que embora atrapalhe e prejudique as operações normais do usuário, não lhe causa danos irreparáveis. Isto é, não elimina programas, altera arquivos, modifica diretórios etc. Esta é uma classificação elástica, uma vez que, de um ponto de vista rigoroso, nenhum vírus é benigno, pois todos eles atuam sem o conhecimento do usuário.

B) MALIGNO: Atrapalha, estraga, elimina, ... enfim, causa prejuízos dificilmente reparáveis ao usuário. Esta categoria por sua vez se subdivide (para efeitos didáticos apenas) em diversas outras, a saber:

B.1) APARENTEMENTE INOFENSIVO Avisa que não vai fazer nada, tranqüilizando o usuário, enquanto traiçoeiramente, elimina arquivos, estraga discos etc. Trata-se de um vírus maligno. O cúmulo desta categoria é representada por uma vacina, cuja propaganda é o efeito benéfico e protetivo que teria sobre o ambiente do usuário quando na verdade é um vírus. Este caso é verídico e ocorreu em um BBS nos Estados Unidos.

B.2) MERCANTIL Feito pelo fabricante de sua vacina. Neste caso o objetivo do fabricante do vírus é comercial, isto é ele quer ganhar dinheiro. O vírus e a vacina são feitos praticamente juntos, inclusive colocando-se marcas mútuas de reconhecimento, para dificultar o trabalho de outras vacinas.

B.3) TERRORISTA Estraga informações sem aviso prévio. Age onde é menos esperado, e no momento que só ele conhece. Neste caso o vírus tende a se confundir bastante com erros de hardware

B.4) TERRORISTA DE HARDWARE Este tipo de vírus (do qual o exemplar mais conhecido é um vírus que ataca a configuração de AT's que é guardada em uma memória RAM protegida por baterias) caracteriza-se por atacar uma parte do hardware da máquina. Uma possibilidade teórica seria a de um vírus que forçasse milhares de movimentações nos braços dos drives, visando desgastá-los e provocar defeitos no computador.

Por exemplo, antigos monitores Hércules podiam queimar por sobreaquecimento se houvesse intensivo chaveamento entre vídeo normal e inverso. B.5) CONCORRENTE DESLEAL Este vírus tem sido sugerido na literatura como uma possibilidade. O autor desconhece se j apareceu um vírus deste tipo, em termos reais. A idéia é que um vírus ataque o código de um programa execut vel de um concorrente do autor do vírus. Desta maneira, o autor do vírus poderia vender seu programa aplicativo que o mesmo estaria livre de problemas, enquanto que o seu concorrente estaria coalhado de "bugs".

Um exemplo imaginário: Suponhamos que eu queira vender um programa XYZ que é uma planilha. No mercado onde eu atuo, por hipótese, o grande "best-seller"é (também por hipótese) o LOTUS. Neste caso, se eu for desonesto e quiser vender meu produto, posso fazer um vírus que ataque apenas o código execut vel do LOTUS, e a seguir espalharei discretamente que o LOTUS tem alguns defeitos que o meu programa não tem.

B.6) VÍRUS CRIPTOGRÁFICO Usa algoritmos de compressão, para apertar o código original e arrumar espaço para ele poder ser colocado no arquivo. O tamanho original do programa não se altera. Quando chamado, o vírus passa a ler e decriptografar o código original, colocando-o na memória, como ele seria se não estivesse contaminado. O potencial de destruição deste vírus é grande, e é difícil sua descoberta.

B.7) VÍRUS DE BIBLIOTECA Colocado em uma biblioteca de compilador, passar a ser incluído em todos os programas compilados. Difícil é a sua propagação, mas uma vez incluído em um programa, torna sua detecção quase impossível, pois afinal, ele faz parte do programa execut vel. B.8) VÍRUS FECHADURA Sua rotina de ativação está a espera de um código, possivelmente a ser fornecido por um segundo vírus.

VÍRUS NÃO SÃO DOCUMENTADOS Esta é uma observação óbvia, mas nem por isso menos importante. Além de não documentados os vírus fazem o possível e o impossível para impedir que o código seja seguido e que as tarefas do vírus sejam levantadas. Para verificar a verdade desta afirmação, basta desmontar (unassemble) um vírus qualquer e acompanhar o seu caminho. Normalmente salta aos olhos a intimidade do programador com a máquina e a dificuldade dos programadores normais de seguir o raciocínio. Por exemplo, na análise do PING-PONG, não se pode lançar mão do DEBUG que permitiria executar passo-a-passo o programa, pois ele se instala antes do DOS, junto ao BIOS, e com isso impossibilita-se de rodar sob o DEBUG (que roda sob o DOS).

QUANTO AO LOCAL DE INFECÇÃO Considerando que os vírus precisam se carregar na memória sem que o usuário perceba, podemos classificar os vírus em:

BOOT INFECTOR Estes vírus se aninham no boot sector (setor 1, trilha 0, face 0) que é carregado e executado normalmente pelo DOS quando um PC é ligado. Estes vírus não passam de um disco a outro com cópia simples, apenas com cópia física. Para infectar uma máquina é necessário dar BOOT com este disco, e ele se auto-grava em todos os discos que forem colocados em um computador infectado, mesmo que tais discos não sejam discos do DOS. É exemplo deste tipo o vírus PING-PONG. PROGRAM INFECTOR Estes vírus "grudam"em programas executáveis, podendo fazê-lo em todos (que são executados) ou em apenas alguns. Neste caso o vírus passa de disco a disco com um simples COPY. Para infectar uma máquina basta rodar um programa que esteja com o vírus grudado. É exemplo deste tipo, o vírus ISRAELI. COMMAND.COM INFECTOR Este vírus é semelhante ao program infector, mas apenas ataca o programa COMMAND.COM. É exemplo deste tipo o vírus LEHIGH.

VÍRUS DE SOBREPOSIÇÃO É o tipo mais simples de vírus. Ele simplesmente insere-se no início do programa hospedeiro, apagando as informações existentes. Quando se chama o programa, é o vírus que vai ser executado. Ao final, o controle é devolvido ao programa (ou ao que resta dele), e o mau funcionamento do ambiente a seguir, costuma delatar a presença do vírus.

9.1 GRUPOS DE ALTO RISCO

1. MICROS COM LIVRE ACESSO Este tipo de equipamento constitui o maior mecanismo de propagação de viroses de computador. Organizações que usam o conceito de pool de equipamentos costumam se infectar em questão de horas graças a este tipo de procedimento. Basta um usuário desavisado deixar um vírus carregado em uma das máquinas para que todos os discos não protegidos ali colocados se infectem e o ciclo prossiga ao infinito.

2. ESTAGIÁRIOS, TÉCNICOS HARDWARE, PESSOAL ESTRANHO Estas pessoas normalmente andam com uma caixinha de disquetes embaixo do braço. Tais disquetes podem ser fontes importantes de contaminação. Os estagiários, por exemplo, costumam trazer das faculdades inúmeros programas que são trocados livremente naquele ambiente. Às vezes, por falta de maturidade ou conhecimento, eles não têm consciência do perigo que esse procedimento representa.

3. MICROS QUE VÃO PARA O CONSERTO Principalmente quando contém discos Winchester. Neste caso, a probabilidade das máquinas voltarem contaminadas pode ser alta. Deve-se desenvolver um trabalho eficiente junto aos consertadores de máquinas mostrando-lhes o perigo de rodar programas de origem desconhecida nas máquinas. De qualquer maneira, sempre que a manutenção for preventiva o winchester deve ser adrede preparado para evitar não só este problema, como principalmente a pirataria. Também não se deve deixar a manutenção cair no extremo oposto e formatar sempre todos os discos que lhes caiam nas mãos.

4. SOFTWARES DE DEMONSTRAÇÃO Tais programas passam de cliente em cliente, raramente parando mais de uma semana em cada um. Basta que um deles contamine o programa, para que ele passe a ser agente de disseminação. O CITYBANK, segundo seu Vice Presidente Henrique Costábile, "teve dez de seus computadores contaminados inadvertidamente por disquetes de demonstração, trazidos por empresas que querem vender programas ao Banco".

5. JOGOS São o berço ideal para os vírus. Copiados "ad infinitum"por usuários são ...s vezes usados por pessoas sem nenhuma experiência e preocupação com o assunto. Muitas empresas têm aproveitado este fato, para proibirem seus funcionários de rodarem este tipo de programa, MESMO FORA DO HORÁRIO COMERCIAL. É uma boa e salutar pr tica. 6. QUALQUER PROGRAMA DE ORIGEM DESCONHECIDA Deve ser encarado como suspeito, até que haja prova em contrário. Caem nesta categoria muitos produtos oferecidos como DOMÍNIO PÚBLICO, ou mesmo SHAREWARE. Precisam ser usados com critério, e certamente devem passar por uma criteriosa avaliação antes de serem liberados para uso normal dentro da organização e/ou distribuídos nela.

10 ITALIAN PING-PONG ou italianinho saltitante

Este vírus é em princípio benigno. É o primeiro a ter surgido em nosso País. Alguns artigos publicados na imprensa internacional sugerem ter sido fabricado em uma empresa italiana. Após a infecção de um micro, e desde que o gatilho ocorra, uma pequena bolinha (o caracter ascii 07), começa a passear aleatoriamente pela tela, ricocheteando nas paredes do vídeo e em algumas letras e caracteres que porventura estejam sendo mostrados.

O vírus desenha e apaga a bolinha sucessivas vezes em posições colaterais distintas. Este procedimento é quem sugere o movimento da bolinha. É fácil ver este esquema funcionando, pois quando se faz um "scroll"na tela, o vírus não consegue mais apagar a bolinha, e ela passa a ficar permanente na tela. Para o vírus conseguir isto, ele reclama (e obtém) do micro o controle a cada operação de vídeo.

Antes de ocorrer o gatilho, mesmo que a máquina esteja infectada, não h nenhum sintoma no micro. Não se percebe degradação significativa, nem nenhum outro sintoma parecido.

O mecanismo de infecção é o de "boot sector infector", isto é em um disco infectado, no setor 1, lado 0, trilha 0, não se encontra o boot record normal, e sim o vírus, que está l . Para não ser descoberto, o vírus desloca o antigo boot record para outro local do disco. A escolha deste local pressupõe que o mesmo esteja desocupado (Em experiências que fizemos, o setor escolhido pelo vírus foi o setor 192 - face 1, trilha 10, setor 4 – Ignoramos se o local é sempre este, ou foi aleatoriamente escolhido. De qualquer maneira este setor estava vago).

Para impedir que o boot record normal seja mais tarde destruído (já que era uma área livre) o vírus marca o setor na FAT como BAD SECTOR, colocando a constante FF7 na tabela de alocação de arquivos. Assim, uma possível pista da existência deste vírus em algum disquete, é a presença de um setor danificado, ou o que dá no mesmo, um disquete com rea útil menor do que o tradicional, que é de 362.496 bytes).

Entretanto, cautela: esta condição é necessária, mas não suficiente: o disco pode estar defeituoso, sem que exista vírus nenhum.

O contágio do vírus se dá quando um disco infectado é usado para dar "boot"em uma máquina qualquer. Como o vírus ocupa o espaço do boot sector, ele é carregado em primeiro lugar, faz todas as trapaças e arreglos que quiser, e depois, carrega o boot record normal, que trabalha normalmente, terminando por mostrar o símbolo tradicional "A>".

A partir deste momento, a máquina está aparentemente perfeita, mas na sua memória está residindo o vírus. Ele interceptou 3 interrupções vitais para sua sobrevivência:

a) Acesso a disco (interrupção 21h) - para determinar o momento de se auto-descarregar em um novo disco

b) Relógio (interrupção 08h) - para, em conjunto com a anterior, determinar o gatilho

c) Vídeo

Qualquer disco que seja colocado no micro, sem proteção, ser vítima do vírus, ainda que sobre o disco se realize uma operação simples, como por exemplo "DIR".

Outra maneira que nos permite identificar um disco que contenha o Italian Ping Pong, é forçando este disco a formatar um terceiro disco. A formatação ser "bem sucedida", mas o disco resultado não conseguira ligar um PC. Isto porque o vírus não tem competência para se reproduzir via FORMAT.

O gatilho do vírus, até onde pudemos observar (isto é não sabemos se o gatilho é único, ou existem outros) é acessar qualquer disco em hora cheia (isto é 00 minutos). Esta é outra maneira de saber se um disco está infectado: Basta dar boot com ele, estabelecer uma hora qualquer, com 59 segundos e 45 segundos, e dar consecutivos DIR's ou outro comando que force acesso a disco, até passar a hora cheia.

Entretanto, a melhor maneira de identificar o vírus, (e a menos prejudicial) é pesquisar o setor 0 (face 0, trilha 0, setor 1) do disco, e verificar se l está o boot record normal. Ele é facilmente reconhecível pela presença, a partir da posição 375 do setor, das mensagens identificativas de problemas no boot. São elas: "Disco sem sistema ou erro no disco", "Troque e pressione qualquer tecla", "Falha na carga do disco"etc.

Se as mensagens estiverem l , o disco não está infectado. Se não estiverem, o disco está com algum problema (pode ser o vírus). Quanto a Winchesters, se o acesso for feito via BIOS, um pequeno cuidado deve ser tomado, pois o boot record não reside neste local. Quem está l é o partition record e o master boot do disco. O boot record do DOS está em algum outro local. Entretanto, se fizermos o acesso através do DOS (usando o DEBUG, por exemplo) a dificuldade desaparece, j que o DOS só conhece a sua partição.

A cura para um disco doente é:

a) Regravar o boot record normal no setor 0. A melhor maneira de fazer isto é usando o comando SYS do DOS.

b) Liberar a rea marcada como BAD SECTOR, colocando-se 000 nas suas localizações da FAT.

O problema pode ser resolvido copiando-se o conteúdo do disco com COPY *.* (o vírus não vai) e reformatando-se o original a seguir.

Existem no mercado alguns programas que se propõe a restaurar a situação original: DeVírus (Olivetti), Leucócito (Inst. Matem tica da USP), Monócito (Itautec). Tais programas refazem as etapas acima, permitindo o retorno a normalidade.

Para encerrar, a informação, não confirmada, de que uma versão do Italian Ping Pong teria sido alterada para maligna, isto é enquanto a bolinha corre a tela, o disco em questão seria destruido.

11 ISRAELI ou SUMSDOS

Este é o outro vírus que junto com o anterior fez grande estrago em instalações brasileiras. Enquanto os demais vírus surgiram apenas esporadicamente, estes dois (italianinho e Israeli) causaram enormes infestações.

O vírus é maligno (destrói arquivos executáveis: programas) e deixa a máquina muito lenta, após um certo tempo de infecção.

A origem do nome é a especulação de que teria surgido na Universidade de Jerusalém, em dezembro de 1987. Originalmente, o vírus deveria ter como gatilho o dia 13 de maio de 1988, data em que o Estado de Israel comemorou 40 anos de história. Coincidentemente, este dia foi uma sexta feira. Provavelmente, alguém fez uma alteração no código do vírus, mudando o teste. Depois que esta versão (2?) se espalhou, o vírus passou a atacar em qualquer sexta feira 13.

Ele atua atacando programas executáveis que sejam rodados a partir de discos sem proteção. Programas do tipo COM, têm o vírus agregado no começo do código original do programa. O vírus ocupa 1800 bytes aproximadamente. J programas do tipo EXE, têm o vírus "grudado"ao final do código, e ocupando os mesmos 1800 bytes. Por um erro de programação (bug?), o vírus não detecta se j infectou ou não um determinado programa tipo EXE. Com isto, ocorrem múltiplas infecções, e o tamanho dos programas EXE que sejam muito executados, cresce sem parar, a razão de 1.8 Kbytes por execução. Existem comentários de que este erro teria sido corrigido, e não haveria mais múltiplas infecções (versão 3?).

Uma proteção simples é a de tornar todos os programas tipo COM e tipo EXE em "read only files". O DOS prevê esta possibilidade reservando um bit no byte de atributos dos arquivos. Isto pode ser feito diretamente (via NORTON ou similar), ou através do comando DOS ATTRIB.

Novamente, existem duas versões do vírus circulando: A mais antiga trava a máquina quando um programa EXE ou COM protegido via read only é executado. (O vírus está tentando gravar no disco, e o DOS não deixa: ocorre o conflito e o resultado é um "lock"). Novamente, existem coment rios de que o vírus teria se modificado, para antes de qualquer coisa, modificar o atributo para "read-write"independente do estado que ele tinha antes.

Quando um programa infectado é rodado, o vírus se instala na memória. A partir deste momento, ele: a) Infecta todos os programas que forem rodados depois dele. b) Começa a contar 30 minutos aproximadamente. Quando isto acontecer, a máquina fica lenta. c) Se por acaso, for uma sexta feira 13, o vírus eliminar quaisquer programas que sejam chamados.

Um detalhe importante, que para o vírus atuar, deve ter sido informado que é dia 13, sexta em tempo de Boot, e não via comando DATE, que para efeito do vírus é inoperante.

A melhor maneira de manter atenção sobre este vírus é acompanhar os tamanhos dos programas executáveis. Uma arma simples e eficiente é criar um programa pequeno, com 1 ou 2 bytes, rodá-lo periodicamente, e acompanhar o seu tamanho. Enquanto ele for de 1 ou 2 bytes, não existe vírus atuando. Quando este tamanho rapidamente evoluir para valores maiores, é sinal de vírus atuando (neste caso, a condição é necessária e SUFICIENTE, pois não há outra explicação para inchação de programas).

É importante salientar, que só colocar um disco que tenha programas infectados no computador, não é suficiente para contaminá-lo, ainda que seja dado boot com este disco. O Israeli só ganha residência na memória, quando um PROGRAMA infectado é executado.

Quanto ... lentidão do micro, nota-se que o vírus altera a interrupção 08h-clock de relógio (que passa a apontar para dentro do vírus, e não mais para as rotinas originais de BIOS). Nesta rotina o vírus testa uma word para o valor 0002h. Se o valor não foi atingido, ele diminui 1h da word. (Não pudemos descobrir qual o valor original da word, se é que existe algum predeterminado).

Quando este gatilho é atingido, para cada interrupção de relógio (que ocorre a cada 55 milissegundos aproximadamente (ou 18,2 vezes por segundo) o vírus faz 4000h movimentações inúteis de 1 byte na memória. Isto significa 16.384 movimentos inúteis a cada 55 milissegundos. Não é a toa que a máquina fica devagar.

Em um micro de 4.77 Mhz, o pedir-se um DIR, praticamente é possível ver cada letra sendo escrita. Em micros de 8 MHz, a coisa é um pouco (mas não muito) mais rápida.

A cura para infecções do Israeli é a recriação dos programas originais. Este fato sinaliza a importância de se usarem programas não protegidos contra cópia.

Quando usamos programas livremente copiáveis, o próprio fabricante sugere que seja feita uma cópia de trabalho em disquetes comuns, e se guarde o original do software como cópia back up. Nestes casos, a infecção se ocorrer, atacar a cópia de trabalho, que pode ser perdida e refeita tantas vezes quantas se quiser, sem nenhum ”nus. Quando usamos programas protegidos (isto é, que não podem ser copiados) se eles forem vítimas de infecção, teremos problemas, pois a recriação da cópia ter que ser feita pelo fabricante, com ”nus financeiros e de tempo grandes e desnecessários.

Uma boa proteção contra este vírus é a colocação de discos protegidos na máquina. São exceções a esta consideração programas residentes em Winchester (que não pode ser protegido) e programas que sabida e reconhecidamente gravam em seus discos. P. ex.: o MS-WORD guarda um arquivo chamado MW.INI que contém a configuração inicial, e que é sempre regravado ao final de cada sessão de trabalho.

Vários programas têm sido escritos para identificar a presença deste vírus em discos e na memória. O utilitário de domínio público MAPMEM quando executado em uma máquina que esteja contaminada, apresenta uma linha referente ao vírus, permitindo portanto sua visualização.

Para pesquisa em discos, os programas de pesquisa costumam procurar a assinatura do vírus (SumS-Dos). Estes programas funcionarão enquanto a assinatura do software não mudar. Exemplos deste tipo de programa são o Ktavírus e ChkVirus, ambos desenvolvidos pela Embratel.

Finalmente, o jornal Computerworld, edição de 13 de fevereiro de 89, comenta sobre a possibilidade de alguma versão do Israeli apagar TODOS os dados dos Winchesters onde atua.

UMA HEURÍSTICA Uma ferramenta para controlar este tipo de infecção é um programeta de 2 bytes, que deve ser sempre rodado e ter o seu tamanho monitorado para detectar quando um vírus ”gruda”nele.

Este programa pode ser assim criado:

```
a) Chame o DEBUG                  A>DEBUG <enter>
b) Indique entrada de comandos
   assembler a partir da posi-
   ção 100h                       -A 100 <enter>
c) Entre a instrução "fim de
   programa"                       int 20 <enter>
d) Indique fim da entrada          <enter>
e) Solicite conteúdo de CX        -R CX <enter>
f) Mude para 2 bytes               2 <enter>
g) Dê nome ao arquivo (name)      -N saidiabo.com
h) Comande a gravação (write)     -W
i) Termine a sessão (quit)        -Q
```

Este trecho cria o programa SAIDIABO.COM no disco, com tamanho de 2 bytes. O programa pode (e deve) ser executado tantas vezes quantas quisermos, pois ele não faz nada, só entra e sai. Entretanto, se a máquina estiver infectada, após sua execução, ele ter o tamanho alterado, para acomodar o vírus.

Assim, se após rodar o programa, seu tamanho permanecer em 2, é porque a memória da máquina não está infectada: isto é: não foram rodados programas ”doentes”desde que a máquina foi ligada.

12 PAKISTANI BRAIN

Este vírus é o mais conhecido, com relação a suas características básicas, como, quando e por quem foi feito. A razão disso, é que desassemblando o vírus, encontra-se a mensagem ”Bem-vindo ... armadilha-seguido do nome e endereço dos seus autores. Muita gente foi atrás deles, e isto permitiu reconstituir a história.

O vírus pakistani brain é maligno, causando destruição geral dos arquivos em disco. O vírus foi criado em Lahore no Pakistão por dois irmãos, um de 25 e outro de 19 anos, ambos programadores. O mais velho fez faculdade de física em Punjab. De volta a Lahore, aprendeu tudo o que podia sobre computação. Rapidamente começou a programar, e a vender pacotes. Segundo ele, a idéia original de escrever um vírus veio quando ele viu seus programas reproduzidos na cidade (e sem o pagamento de direitos).

Mais tarde, abriu uma butique de computadores e programas e começou a vender cópias de Lotus123 pelo preço de US$ 1.50. Tal programa, na época (1987) custava nos EUA mais de 300 dólares. Como o Paquistão não tinha uma legislação de proteção ao direito autoral (como os EUA, e até o Brasil tem), não era ilegal vender estas cópias. Assim, muitos turistas universitários americanos acabaram levando para seu país cópias de 123 devidamente contaminadas pelo vírus. Estima-se que no auge, este vírus tenha infectado mais de 100.000 PCs. Como curiosidade, as cópias que os irmãos vendiam para paquistaneses eram isentas do vírus. Só as cópias de turistas estavam infectadas.

Este vírus também infecta o boot-sector. Além disso, ele ocupa mais 7 setores do disco, onde é colocado o resto do vírus, bem como o boot sector original. Estes sete setores são marcados como BAD SECTOR, a fim de não serem destruídos.

A infecção se dá , tal como no ping-pong, quando se coloca um disco bom em uma máquina doente, ou vice versa. Como curiosidade, todo disco infectado tem seu nome (volume name) modificado para arroba BRAIN. Outra maneira de identificar o vírus é pela ausência das mensagens do DOS originais no setor 0 do disco.

A cura, para CPUS infectadas, é a recarga de um sistema "bom". A cura para discos é formada por: a) Regravação do boot sector original (comando SYS) b) Mudança do nome do volume c) Limpeza dos sete setores ocupados

Finalmente, este vírus embora não tenha chegado em grande escala ao nosso País, é reputado pelos especialistas americanos como o mais bem feito dos vírus que rodam por aí.

13 ALAMEDA

Este vírus é um boot infector, similar portanto ao pakistani e ao ping-pong. Sua origem está reportada ao Merrit College em Oakland Califórnia, EUA, e a data prov vel de sua conclusão é a da primavera de 1988.

É maligno, e causa perda de dados. Como curiosidade, que o diferencia dos demais, está o fato de que o boot-sector original é transferido para o primeiro setor livre, que NÃO é marcado como bad sector. Com isto, qualquer crescimento de arquivos sobrepõe o setor de carga original e o processo de boot com este disco deixa de funcionar.

O vírus se aperta em 512 bytes, não precisando ter continuação. Aparentemente este vírus só ataca PCs originais feitos pela IBM. Se isto for verdade, desta praga estamos livres no Brasil, pois aqui teoricamente não existem estas máquinas.

14 LEHIGH

Este vírus é maligno (causa a perda total dos arquivos em discos) e teve este nome por ter sido desenvolvido na Universidade de Lehigh, no outono de 1987.

O local de infecção é o programa COMMAND.COM, que sofre as seguintes alterações quando é infectado: a) o tamanho do módulo aumenta 20 bytes b) a data e a hora do arquivo são modificados

O gatilho que dispara o vírus e causa a perda total de dados é o número de infecções: Assim, depois da quarta infecção o vírus libera toda sua virulência e destrói os arquivos.

Há o comentário de que este número teria sido alterado para 10 infecções.

A maneira de corrigir a infecção é recopiar o arquivo COMMAND.COM de fonte original e segura.

15 dBASE VÍRUS

O vírus dbase ainda não chegou ao nosso país. As informações listadas a seguir são de bibliografia americana. De todos os espécimes aqui listados é o mais letal, e o que tem maior potencial de estrago.

Ele não ataca o dbase em sí, mas sim os arquivos cuja extensão é DBF, que como sabemos são os arquivos de dados deste software. Entretanto, ele gruda no programa dbase, para atuar sempre que este for chamado.

Quando ocorre a infecção, o vírus intercepta as chamadas 21h de DOS, principalmente as funções de ”open”de arquivos. Cada vez que um arquivo é aberto, o vírus determina se ele é do tipo DBF. Se não for, a operação é completada sem interferência. Se for, o vírus começa a atuar.

Cada bloco de dados que é trazido para a memória tem os 2 primeiros bytes invertidos. A alteração é registrada em um arquivo chamado BUGS.DAT (”hidden-escondido), para poder ser desfeita pelo próprio vírus.

A situação fica assim: no disco existem inúmeras inversões de bytes por todo(s) o(s) arquivo(s). Só que analisando estes arquivos através do dbase (e conseqüentemente do vírus) nada aparece, pois sempre h a desinversão dos dados alterados, antes deles serem apresentados. O defeito só apareceria se o arquivo fosse lido por um user-program, em pascal, por exemplo, ou ainda através de um utilitário tipo TYPE.

Enquanto o incauto usuário acha que tudo vai ...s mil maravilhas, o vírus amplia a sua nefanda ação.

90 dias após a primeira infecção, o vírus tem o seu gatilho e daí:

a) Ele elimina o arquivo BUGS.DAT b) se auto-elimina do disco c) destrói a FAT do disco.

Com isto os arquivos estão praticamente perdidos. Quaisquer back ups de 90 dias para c também são inúteis.

Este vírus ataca PCs compatíveis.

16 Outros

16.1 nVIR

Desenvolvido na Alemanha Ocidental, na cidade de Hamburgo, em 1987. Só ataca micros do tipo McIntosh. Atua infectando as aplicações (tal como o Israeli). Este vírus teve seu código publicado (e popularizado). Este fato fez com que aparecessem inúmeras versões (cada uma ataca uma coisa diferente), mas todas com mecanismos de funcionamento e infecção parecidas. Como curiosidade, quando existe o dispositivo MacinTalk, ouve-se a mensagem ”Don’t panic”... Este programa é muito virulento e pode infectar todos os programas existentes em um sistema contaminado em questão de minutos. A razão do nome, é que o vírus contém em seu código esta assinatura. Pode ser considerado benéfico, pois somente faz a tela piscar quando qualquer programa é rodado.

16.2 ScoreS

A origem do nome, é o fato do vírus criar um arquivo não visível com este nome. Teve origem na Electronic Data Systems, na cidade de Dallas, em 1987. Ataca McIntosh, e nestes infecta aplicações (tal como o nVIR) que tenham os identificadores ERIC e VULT. Cada aplicação infectada cresce 7Kbytes. A cada 3,5 minutos, pesquisa novas vítimas. Além de destruir arquivos, causa lentidão nos sistemas, problemas na impressão, modifica os ícones do Mac, e causa mal funcionamento do sistema como um todo. Como os identificadores pertencem ... firma EDS, suspeita-se de sabotagem interna, pois o vírus não destrói nenhum arquivo.

16.3 Mac Mag Peace

Bolado pelo editor de uma revista especializada em Macs, ele teria como gatilho, a data 2 de março de 88, primeiro aniversário do MacIntosh II. Neste dia, ele emitiu uma mensagem pedindo a paz mundial e depois se auto-eliminou. Programas contaminados que rodassem depois deste dia, também perderiam o vírus que se retiraria sozinho. A curiosidade neste caso, é que por acidente, o vírus contaminou a cópia mestre do software FreeHand da Aldus, antes do disco ser enviado para duplicação. Calcula-se que 5000 cópias do vírus tenham sido disseminadas por esta forma.

16.4 Sunny Valley

Desafia o usuário a encontrá-lo: ”Welcome - Can you find me?”

16.5 Christmas Card

Vírus em rede de mainframes IBM. Feito dentro da própria empresa. Foi benigno, limitando-se a desejar feliz natal a todos na véspera do Natal. Pelas informações disponíveis, tratou-se de um procedimento feito sob CMS, que quando enviado a alguém (através do comando EXEC) entrava na lista de destinatários

dessa pessoa e re-enviava o mesmo EXEC para todos eles. E em cada um o ciclo se repetiu. Assim, não se tratou de um vírus na acepção completa da palavra, (de vez que não interferiu diretamente na máquina - através de linguagem de máquina), mas mesmo desta maneira, vale como exemplo.

16.6 Cookie virus

Vírus colocado em rede de mainframes de universidades americanas. Baseado no personagem de Vila Sésamo, chamado Beto, e que a todo momento interrompia pedindo: "Me dá um biscoito". Este vírus quando acionado exige que a vítima digite "c-o-o-k-i-e"(biscoito), para contentar o vírus.

17 DESCONFIE SE

1. Coisas estranhas começarem a acontecer em seu micro.

2. Aparecerem arquivos com zero bytes - Deve-se notar entretanto que alguns softwares montam seu esquema de proteção baseados em arquivos com esta característica. Isto pode ocorrer também devido a mal funcionamento de software, ou mais raramente de hardware. Por exemplo, o software LETTRIX (da Hammerlab Co) tem como resultado do processo de INSTALL um arquivo oculto, de 0 bytes de tamanho, de nome FIXED, com os atributos H (hidden) e R (read only)

3. Programas com tamanho exagerado - Característica principal de programas multi-infectados. Além disto devemos ver os programas que "crescem"de tamanho, ainda que em pequena escala. Um programa pronto e acabado não pode crescer de tamanho.

4. Datas anteriores a 01/01/80 - Tradicionalmente a data inicial para todo o universo PC. Na verdade, o DOS não tem como guardar uma data deste tipo. Quando aparecer um valor menor, é porque alguém andou alterando a própria interpretação deste campo. Da mesma forma deve-se olhar arquivos com data maior do que a data atual. (Isto também é estranho!).

5. Idênticas considerações para horas maiores do que 23:59:59. Isto também é inexplicável.

6. Micro muito lento. Esta característica pode sugerir a presença de viroses no micro. Entretanto não é determinante, j que falhas de hardware podem forçar a repetição de operações, e isto é feito SEM avisar o usuário, e é NORMAL. Outra possibilidade em micros chaveáveis (por exemplo: 4.77 ou 8 MHz) é que alguém tenha alterado a velocidade do mesmo para o menor valor.

7. Acessos indevidos a disco. Sempre que durante um processo em memória ou em vídeo ou em teclado, o micro acessar os discos, devemos ficar alertas. Entretanto cuidado deve ser tomado para a presença dos utilitários residentes tipo MAPMEM, PRN2FILE, DCACHE, DOSEDIT e outros, que as vezes comandam operações em disco. Pode acontecer de esquecermos que eles estão na memória, mas se os colocarmos eles l ficarão (e de l lerão discos).

8. Ganhar de presente um programa de "origem desconhecida". Todos eles são, salvo melhor juízo, sérios candidatos a "vírus". Nada de sair distribuindo este programa. Sugiro uma boa quarentena para ele.

Para encerrar este capítulo, uma observação:

Não adianta criar uma "psicose"de vírus. Na nossa instalação, e depois da divulgação maciça do assunto "vírus", qualquer engasgada fora de hora do equipamento, imediatamente é diagnosticada como... VÍRUS.

Esta atitude é tão prejudicial quanto desconhecer o problema. Na verdade, os softwares vão continuar fazendo loucuras, e cada uma delas tem explicações racionais (e banais). Da mesma forma, o hardware vai continuar falhando, apresentando problemas, e criando confusões. Para isto, a solução é ... técnicos de hardware.

SWAT Para organizações grandes, e fortemente dependentes da informática, creio ser boa idéia criar uma "SWAT"para atacar os vírus. Este grupo seria formado por técnicos de bom nível, com formação multi-disciplinar, trabalhando de maneira descentralizada, em diversos setores, mas com reuniões periódicas para troca de informações.

Uma vez levantado um foco, este grupo seria chamado e imediatamente entraria em ação, com prioridade zero sobre todas as demais atividades.

Este enfoque se justifica, pela pressa em isolar o problema, e posteriormente saná-lo. A literatura tem sido pródiga em mostrar empresas que foram apanhadas de surpresa e ficaram 2,3 ou mais dias inativas por causa de infestações. Isto pode ser muito tempo para uma empresa.

18 PARA SE PROTEGER

1. INFORMAÇÃO É A MELHOR ARMA Programas formais de instrução e disseminação de informações sobre o assunto devem ser preparados. A estrutura de Centro de Informações parece ser a mais indicada para dar conta deste recado.

Conhecer o problema é o primeiro passo para dominá-lo. Identicamente, o "overhead"causado pelas medidas profiláticas só vai ser facilmente aceito se os usuários tiverem consciência do perigo e dos problemas que podem advir.

Sugiro palestras, cartazes, artigos em jornais de divulgação interna, demonstrações pr ticas e similares podem e devem ser usados nesta tarefa. Ninguém poder alegar ignorância no assunto.

Como sugestão final, recomendo uma abordagem sóbria do assunto. Não podemos permitir que estes vírus joguem pela janela anos de conscientização e "venda"da informática como ferramenta de aumento de produtividade.

Ou seja, o problema é importante, pode ficar pior, mas não é o fim do mundo, e pode ser atacado com ações simples e baratas.

Citando Raul Fernando Weber e Taysi Silva Weber: "... Na realidade, apesar do grande potencial destrutivo dos vírus, eles são responsáveis por uma percentagem muito pequena do mal funcionamento dos computadores - Especialistas americanos estimam que somente um em cada vinte problemas atribuídos a vírus é realmente causado por um."2. ACABE COM A PIRATARIA Vírus e pirataria andam juntos. Isto parece dar razão aqueles que acusaram a indústria de software pelo surgimento dos vírus. Pelo sim ou pelo não, e aproveitando para cumprir a lei 7646, (lei de software), determine e faça cumprir a ordem: "só se usa software oficial".

A principal vantagem do programa oficial sobre o pirata neste enfoque reside na garantia de origem. Isto é, qualquer mal funcionamento, dar ensejo a cobranças e pedidos de suporte para o fabricante.

3. PROTEJA SEU MICRO Cada micro deve ter uma lista de pessoas e/ou departamentos que podem utilizá-lo. Isto permite o estabelecimento de um "responsável"pelo micro. Alguém que olhe e cuide do equipamento. Esta questão é muito importante no caso de haver winchester, e não tão fundamental para micros com 2 drives.

Esta abordagem desaconselha o uso de micros em "pool", que até pouco tempo atrás era uma tendência em organizações nacionais. O sucessivo barateamento do hardware (ainda que não do software) parece estar indicando que o caminho é a personalização dos equipamentos.

4. PROÍBA JOGOS Organizações sérias e preocupadas com o correto uso dos recursos da informática têm atuado no sentido de proibir o trânsito e uso de jogos no universo PC. Esta proibição tem importância no caso de vírus (pois jogos são agentes contaminadores por excelência), mas eu acho que além disso, ela tem uma vertente psicológica importante na organização, por elevar a informática a um pedestal estratégico na empresa.

A proibição de jogos pode estar sinalizando que a computação é importante demais para se arriscar com este tipo de aplicativo.

5. SUBMETA PROGRAMAS DE ORIGEM DESCONHECIDA À QUARENTENA Nem 8 nem 80. Também não devemos nos proibir de usar programas que sem serem best-sellers, podem ser importantes em um determinado momento na organização. Principalmente agora, que no Brasil começam a surgir os programas do tipo SHAREWARE e de domínio público. Tais programas podem e devem ser usados quando forem necessários. Entretanto, antes de usá-los em larga escala, eles devem sofrer um processo de quarentena, que os teste e os libere para uso generalizado.

Até prova em contrário, estes programas devem ser considerados como potenciais "vírus".

Por quarentena, entendemos um processo sério, calmo e rigoroso de pesquisa de detecção de vírus. Lembrando sempre, que a característica de vírus é a sua reprodução, esta pesquisa visa descobrir se um problema (teoricamente causado por vírus) pode passar de um disco a outro, ou de uma máquina a outra, sem que o operador ordene diretamente isto. 6. COM WINCHESTER NÃO DÊ BOOT EM DISQUETES Em máquinas com winchester, nunca deve ser dado boot com outros discos. Este cuidado é elementar e visa evitar o contágio de boot sector infectors. Se este procedimento for necessário, deve se fazer um teste antes (em outra máquina, sem winchester) para saber se o disco de boot está ou não atacado por algum problema.

7. RETIRE OS DRIVES Esta pode ser uma solução radical, mas eficiente. Todas as máquinas com winchesters (as mais sujeitas a contaminação e onde esta pode trazer maiores problemas) têm seus drivers retirados. Neste caso, algumas ações devem ser tomadas:

a) Uma ligação em rede e/ou micro-mainframe em geral, precisa ser estabelecida.

b) Um drive de disquete, facilmente instal vel deve estar disponível na organização. Este drive ser colocado na máquina sempre que houver necessidade de manutenção, geração de softwares, recuperação de arquivos perdidos etc.

Vantagens: a) No Winchester não reside o FORMAT. b) Back ups são feitos no mainframe, sem maior preocupação do usuário c) Não h risco de contaminação via mídia compartilhada d) Não h risco de uso de programas piratas e/ou jogos e) Não se cria "lixo"no winchester

8. USE PROTETORES Este tipo de programa, (os quais serão vistos adiante) pode ser uma solução para o problema. Trata-se de um programa que se instala na memória e passa a monitorar o trabalho do micro, alertando o usuário sempre que algo "suspeito"acontecer.

Embora tragam uma contribuição importante, principalmente em ambientes sensíveis (onde não pode haver contaminação sob risco de catástrofe), eles enfrentam em minha opinião 2 problemas:

a) Trazem um overhead significativo,tanto maior quanto mais ativa for a "bisbilhotice"dos protetores.

b) Podem causar uma sensação de falsa segurança. Como vimos os vírus são programas magnificamente bem feitos. E que estão mudando (e se aperfeiçoando) todo dia. O uso de um protetor pode dar a sensação de proteção total, quando na verdade os vírus estão atuando. Esta restrição é diminuída, se os protetores não forem usados sozinhos, e sim em conjunto com outras medidas.

9. LEMBRE-SE QUE TUDO PODE MUDAR Talvez a principal ameaça dos vírus seja o fato de que nunca chegaremos a conhecer todas as ameaças em suas totais extensões. A esperança é de que seja apenas um modismo, e que rapidamente as pessoas desistam destas iniciativas. Entretanto, se isto não for verdade, poderemos dizer, quanto aos vírus: "Dias piores virão..."

10. CHEQUE PERIODICAMENTE TAMANHOS DE PROGRAMAS Esta é outra maneira de se controlar a infestação de vírus do tipo "program infector". Eu imagino que possa haver, próximo ao micro, uma listagem com os principais programas da instalação e seus correspondentes tamanhos. Algo do tipo: DEBUG: 15717, FORMAT: 11245, COMMAND: 24316, CHKDSK:10096, TREE: 9558 etc. Não custa olhar para a lista e periodicamente verificar um ou mais programas, para ver se não tiveram seu tamanho modificado.

11. PONHA DISQUETES COM SELO Sempre que for possível, coloque os disquetes com o selo protetor contra gravação. Trata-se de uma proteção simples, e até onde podemos saber, 100

12. CUIDADO COM A MANUTENÇÃO Os técnicos de manutenção representam perigo para a disseminação de vírus. Primeiro, pelo fato de em geral não terem conhecimentos de software e de suas sutilezas. Segundo por trabalharem com múltiplas máquinas, vindo de ambientes os mais diversos possíveis. Terceiro por usarem mídia compartilhada ... exaustão.

Por estas razões, qualquer máquina que volte do conserto (com winchester) deve ser considerada suspeita e entrar em quarentena. máquinas sem winchester, devem ser zeradas (desligadas) após a manutenção.

13. EM MICROS PÚBLICOS: DESLIGUE ANTES DE USAR Só o simples CTRL+ALT+DEL pode (em tese) não ser suficiente. Por via das dúvidas, antes de usar um micro público, deve-se desligá-lo e religá-lo a seguir. De qualquer maneira, o uso direto (sem desligar e sem dar CTRL+ALT+DEL) equivale a um risco inaceitável. Aqui, ponto para as máquinas que têm o botão facilmente acessável.

14. COLOQUE CHKDSK NO AUTOEXEC Esta recomendação j era feita antes dos vírus como maneira de conhecer o estado dos discos. Agora ela ganha importância por informar:

a) Eventuais arquivos ocultos b) Setores ruins c) Outros problemas de alocação (grânulos órfãos, ou alocação circular) etc

Tais problemas podem ser ocasionados por vírus, mas mesmo que não sejam, devem ser investigados, e sua causa removida.

15. FAÇA BACK UPS INTELIGENTES Principalmente para poder enfrentar vírus com prazos de gatilhos muito estendidos. Em vez de fazer uma única cópia todas as sextas feiras (por exemplo), devemos nos preparar para poder recuperar arquivos que foram estragados h muito tempo. P. ex.:

Ciclo semanal: um ou mais jogos de discos para o ciclo semanal (todas as sexta feiras)

Ciclo mensal: um ou mais jogos de discos para o ciclo mensal (todas os últimos dias do mês)

Ciclo semestral ou anual: idem idem.

Além destes back-ups, os sistemas sensíveis devem prever backups dos arquivos de movimentação, a fim de poder recriar arquivos mestres, mesmo com backups velhos e desatualizados.

16. ALTERE ATRIBUTOS DE PROGRAMAS Todos os arquivos COM e EXE devem ser "read only". Isto impedir (na medida do possível) que eles sejam apagados, alterados, tenham nome modificado etc. Isto pode ser feito pelo comando ATTRIB do DOS. Até a versão 3.20 ele precisa ser emitido sub-diretório a sub-diretório, e a partir da versão 3.30 existe a atribuição global que vale para todo o winchester.

Na versão 3.20 o comando é: ATTRIB +R *.COM, e ATTRIB +R *.EXE

Na versão 3.30 e posteriores o comando é: ATTRIB +R *.COM/S O parâmetro /S significa global (vale para todos os subdiretórios)

Se um programa precisar ser alterado, ou regravado, ele deve ter seu atributo re-modificado para read-write. O comando que faz isto é ATTRIB -R <nome-arquivo>.

17. ESCOLHA MICRO PARA HOMOLOGAÇÃO E TESTE Estas atividades devem ser sempre executadas em um só micro, e este fato deve ser conhecido em toda a organização. Aqui o termo homologação, ganha significado inclusive em termos de validar um programa que não traz risco ... organização.

18. MUDE O COMMAND.COM DE LUGAR Para prevenir vírus que atacam o COMMAND.COM, deve-se mudá-lo de lugar, pois quem o chamar via DOS o encontrar facilmente, mas quem o chamar via BIOS (é o caso do vírus) não vai encontrar nada.

Criar um subdiretório (de preferência escondido), talvez de nome HIDDEN e copiar-se para l o COMMAND.COM original. Feito isto, devemos: a) Em CONFIG.SYS, colocar SHELL=C:/HIDDEN/COMMAND.COM/P b) Em AUTOEXEC.BAT, colocar SET COMSPEC=C:/HIDDEN/COMMAND.COM

19. EM REDES EVITE O USO DO FILE-SERVER Esta recomendação sugere que se evite este uso, pelo potencial de contaminação que este uso poderia vir a ter. Como a maioria dos vírus não consegue atuar inter-estações, qualquer infestação teria car ter local.

20. EVITE DEL OU ERASE Preferindo programas que efetivamente limpam a rea anteriormente alocada aos arquivos. Para limpeza total de discos, o comando preferido deve ser o FORMAT.

21. EM MICROS DE 2D: MANTENHA DOS GRUDADO Em instalações onde se costumam usar micros com 2 drives, cada micro deve ter um disco especialmente desenhado para dar BOOT na máquina. Este disco deve de alguma maneira estar localizado próximo ... maquina, e ele garante que sempre é o mesmo S.O. a ser utilizado, e é claro, ser protegido.

22. SEGURANÇA NÃO É DE GRAÇA Todas estas recomendações vão causar incômodo, ou pelo menos modificação na maneira de atuar da instalação. É preciso assumir coletivamente, que este preço está sendo pago para se evitarem problemas de contaminação. Para quem j sofreu o problema na carne, não h o que convencer, os fatos falam por sí. Mas considerando-se o car ter preventivo das atitudes, uma boa parte do esforço deve ser gasto em convencer da necessidade das reservas.

19 VACINAS & PROFILAXIA (prevenção)

VACINAS Vacinas são programas que protegem o ambiente do micro quanto a infestação de vírus. O termo vacina, não deve ser levado ao pé da letra, pois funcionamento deste programas não é idêntico ao das vacinas usuais.

Melhor seria empregarmos o termo protetores, pois é isto o que eles fazem.

1. As vacinas na acepção da palavra, não são muito recomendáveis, pois: a) podem trazer uma sensação de falsa segurança b) Cada vírus exige sua vacina particular. Não é possível ser genérico. Daí a dificuldade. Mesmo uma vacina provada e aprovada para um certo vírus pode deixar de funcionar para uma mutação daquele mesmo vírus.

PACOTES ANTI VÍRUS: Buscam vírus (ou pelo menos programas estranhos) na memória, no vetor de interrupções, nos disquetes e nos winchesters. Estes produtos em geral procuram pela assinatura (ou por uma constante conhecida) dentro dos programas. Por exemplo, qualquer programa contaminado pelo ISRAELI, ter a assinatura "SumSDos". Existem inúmeros no mercado americano, e j começam a aparecer no Brasil. Estes programas são baratos (muitas vezes gratuitos) e depois de verificados quanto a serem "vírus aparentemente inofensivos"podem ser usados praticamente sem contra-indicações. Entretanto, devemos fazer duas ressalvas:

1. O vírus e o pacote anti-vírus são programas. Cada um tem um "quantum"de inteligência do seu autor. Só um dos dois pode ganhar. É errado achar que o pacote anti-vírus sempre vai ganhar. Isto pode ser verdade para os vírus velhos, j conhecidos. Entretanto novos vírus, ou mesmo mutações de vírus antigos podem levar a melhor sobre estes pacotes.

2. Quando isto ocorrer, teremos uma falsa proteção, que segundo muitos autores é pior do que nenhuma proteção. Fazendo uma comparação, imagine uma casa no meio do campo sem para raio. Numa tempestade forte, seus habitantes podem resolver sair a céu aberto para não terem riscos. Agora imagine a mesma casa, na mesma tempestade com um pára-raio insuficiente, ou mal instalado, ou com qualquer problema de funcionamento. Os habitantes podem achar que estão seguros quando não estão. Isto pode ser pior do que não ter p ra raio nenhum.

PACOTES DE CONTROLE DE ACESSO Estes produtos não se preocupam diretamente com vírus, mas garantem que nenhum desconhecido ligar e usar a máquina. Eles se baseiam em complexas tabelas de

usuários e de passwords, como suportes a um sofisticado processo de "sign-on". Aumentando a "higiene"no uso da máquina, estes programas garantem uma maior segurança quanto aos vírus

PACOTES DE SEGURANÇA GENÉRICOS Identificam e avisam sobre as seguintes ocorrências que estejam acontecendo no micro. Exemplos: a) Um programa não autorizado querendo ganhar residência na memória b) Tentativas de escrever em programas do tipo COM e EXE c) Tentativas de renomear módulos. Exemplos de programas deste tipo: Supervisor PC (3I inform tica) Capoeira (Módulo) etc.

EXEMPLO REAL: CURIÓ Este programa produzido pela MÓDULO Consultoria do Rio de Janeiro, garante ao DOS todas as ferramentas de segurança necessárias para garantir um uso só por pessoas, programas e arquivos autorizados.

Inicialmente o programa necessita um cadastro de usuários (pessoas) autorizadas a usá-lo. Estes usuários se dividem em 4 categorias:

-Gerente: O maior nível de autorização, é quem instala o software e determina seu ambiente nativo. Ele pode fazer o que quiser, sem nenhuma restrição.

-Administrador: É o segundo nível, e mantém uma hierarquia maior que os usuários comuns, podendo realizar a maioria das tarefas.

-Auditor: Tem a habilidade de entrar nas bibliotecas dos usuários comuns, verificando o que e como está sendo feito.

-Normal: É o usuário comum.

Os arquivos (e conseqüentemente os diretórios), podem ser protegidos em 3 níveis: leitura, escrita e execução. Desta forma, arquivos/diretórios que não possam ser lidos, fazem com que o seu conteúdo seja protegido contra vazamento e pirataria. Arquivos/diretórios que não possam ser gravados estão protegidos (em princípio) contra vírus. A proteção contra execução é similar e impede que programas executáveis sejam rodados.

Os drives "A"e "B"podem também ter seu uso restrito. Além de proteções similares (leitura, escrita e execução), mediante a troca da ROM da máquina, pode-se inibir a máquina de tentar carregar o sistema operacional a partir de disquete. Isto torna realmente o ambiente seguro.

O administrador ou o gerente podem determinar para cada um dos usuários, quais comandos do DOS podem ser usados. Assim, permite-se a criação de ambientes especiais para cada um dos usuários do computador. Usualmente os comandos DATE e TIME estão inibidos, fazendo com que o recurso de DATA se comporte tal como nos mainframes, isto é não podendo ser alterado pelos usuários. Também pode ser estabelecido pelo administrador, qual a cota do disco rígido que caber a cada pessoa que usar o sistema. Quando esta quota for atingida, o usuário passar a receber o status de DISK FULL, embora ainda exista espaço no disco.

O Curió, admite também o estabelecimento de senhas que precisam ser fornecidas para a liberação de recursos. Identicamente arquivos podem ser cifrados, de tal maneira a serem irreconhecíveis fora do ambiente CURIÓ.

Existe a facilidade de exigir do usuário a realização de back ups. Enquanto eles não forem feitos, a máquina não ser liberada para uso normal. programas disponíveis CHECK-UP: Fabricado pela Unidados Informática de S.P., este programa verifica a existência dos vírus: Ping-Pong, Israeli, Screen e Pakistani Brain. O programa apenas sugere a existência dos vírus, o que no caso de sistemas operacionais que não o MS ou PC DOS, pode resultar em alarme falso. No caso do Israeli, ele procura o SumSDos, o que pode trazer problemas em caso de alteração. Para executá-lo basta chamar CHECK-UP <enter>. As opções de análise são a memória, o drive "A"e o winchester. O programa é conversacional.

ANTISUM: Produzido por Anderson Cunha em maio de 1989, este programa restaura infecções e múltiplas infecções de ISRAELI. Não se deixa enganar pela troca da mensagem. Após a recuperação, o programa mantém algumas diferenças em relação ao original, mas aparentemente volta a funcionar. Basta executar ANTISUM, e o drive padrão ser pesquisado para encontrar todos os arquivos COM e EXE, e estes serão pesquisados para encontrar o ISRAELI. O disco deve ser colocado no drive SEM proteção. Basta chamar o programa, na forma: ANTISUM <enter>.

CHKVIRUS Produzido pela Embratel, ele pesquisa a existência dos vírus conhecidos. Existe o compromisso de que novas versões do produto pesquisem novos vírus que venham a aparecer. Apenas sugere a existência do vírus e recomenda atitudes. Permite a livre cópia e uso, desde que sem fins lucrativos. Deve ser chamado assim: CHKVIRUS <drive> <enter>.

RMAP Programa distribuído pela PC Magazine (domínio público) e produzido por Adrian Evans, permite conhecer o mapa de memória da máquina em um dado instante. Uma vez executado, ele fica residente, e pode ser acionado a qualquer momento com a pressão de CTRL e ESC. Embora não seja específico de vírus, o mapa pode dar informações preciosas. Para executar o programa, basta rodar RMAP <enter>.

ITALIANO Programa com mensagens em italiano, informa se o vírus ping-pong está ou não presente no disco.

CHKMEM Também produzido pela Embratel, este programa informa quais os endereços de interrupção foram alterados. Mapeia a memória do micro e sugere a existência ou não de vírus. Pode ser distribuído, desde que sem fim comercial. Para rodar, fazer CHKMEM <enter>.

LIMPA Também produzido pela Embratel, este programa limpa as áreas não usadas no disco. Para usar, LIMPA <drive> <opção-opcional>. Se a opção é escrita (qualquer caractere) apenas é verificada a presença de vírus. Pode ser distribuído, desde que sem fim comercial.

20 CASOS REAIS: PETROBRAS

Nesta empresa, uma das maiores usuárias de informática do País, os primeiros casos aconteceram em dezembro de 1988. A primeira infestação ocorreu em dez micros de departamento comercial. Como a empresa tinha na época um universo de 5.000 Pcs, o alastramento da epidemia teria efeitos danosos. Assim, a empresa agiu em duas frentes:

a) Campanha de esclarecimento, formada por palestras, artigos no PETROINFO e cartazes que se espalharam pela empresa, e b) Criação de uma brigada de combate a vírus

Na seqüência, houve uma restrição de acesso dos chamados "grupos de risco". Foram eles: a) Técnicos de manutenção b) Estagiários c) Prestadores de serviço d) Softwares demonstração

Finalmente a PETROBRAS passou a desenvolver uma série de protetores para uso dentro da organização.

20.1 EMBRATEL

Na Embratel, que é uma empresa de telecomunicações, a virose atacou de maneira descentralizada. Mais ou menos na mesma época apareceram infestações em Rio, São Paulo, Vitória e Recife.

O que originalmente foi diagnosticado como lentidão excessiva dos PCs, foi rapidamente caracterizado como vírus. O universo da Embratel era composto de 700 máquinas do tipo PC.

Após a descoberta do vírus Israeli, todas as máquinas infectadas foram colocadas de quarentena. Isto significa não rodar nada e principalmente não copiar nada.

Para enfrentar a situação, a EMBRATEL desenvolveu 3 programas protetivos (1800 linhas em C), que testam todas as características do PC (buffers, interrupções, diretórios, clusters vazios, fim de arquivos etc.