Catalog

ELECTRONIC MAIL SECURITY

- PGP is an open-source, freely available software package for e-mail security. It provides authentication through the use of digital signature, confidentiality through the use of symmetric block encryption, compression using the ZIP algorithm, and e-mail compatibility using the radix-64 encoding scheme.
- PGP incorporates tools for developing a public-key trust model and public-key certificate management.
- S/MIME is an Internet standard approach to e-mail security that incorporates the same functionality as PGP.
- DKIM is a specification used by e-mail providers for cryptographically signing e-mail messages on behalf of the source domain.

PRETTY GOOD PRIVACY

- 1. It is available free worldwide in versions that run on a variety of platforms, including Windows, UNIX, Macintosh, and many more. In addition, the commercial version satisfies users who want a product that comes with vendor support.
- 2. It is based on algorithms that have survived extensive public review and are considered extremely secure. Specifically, the package includes RSA, DSS, and Diffie-Hellman for public-key encryption; CAST-128, IDEA, and 3DES for symmetric encryption; and SHA-1 for hash coding.
- 3. It has a wide range of applicability, from corporations that wish to select and enforce a standardized scheme for encrypting files and messages to individuals who wish to communicate securely with others worldwide over the Internet and other networks.
- 4. It was not developed by, nor is it controlled by, any governmental or standards organization. For those with an instinctive distrust of "the establishment," this makes PGP attractive

NOTATION

K_s = session key used in symmetric encryption scheme
PR_a = private key of user A, used in public-key encryption scheme
PU_a = public key of user A, used in public-key encryption scheme
EP = public-key encryption
DP = public-key decryption
EC = symmetric encryption
DC = symmetric decryption

H = hash function
|| = concatenation
Z = compression using ZIP algorithm
R64 = conversion to radix 64 ASCII format[1]

SUMMARY OF PGP SERVICES

Function	Algorithms Used	Description
Digital signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message.
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman or RSA	A message is encrypted using CAST-128 or IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie-Hellman or RSA with the recipient's public key and included with the message.
Compression	ZIP	A message may be compressed for storage or transmission using ZIP.
E-mail compatibility	Radix-64 conversion	To provide transparency for e-mail applications, an encrypted message may be converted to an ASCII string using radix-64 conversion.

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

OPERATIONAL DESCRIPTION

- The actual operation of PGP, as opposed to the management of keys, consists of four services: authentication, confidentiality, compression, and e-mail compatibility

AUTHENTICATION

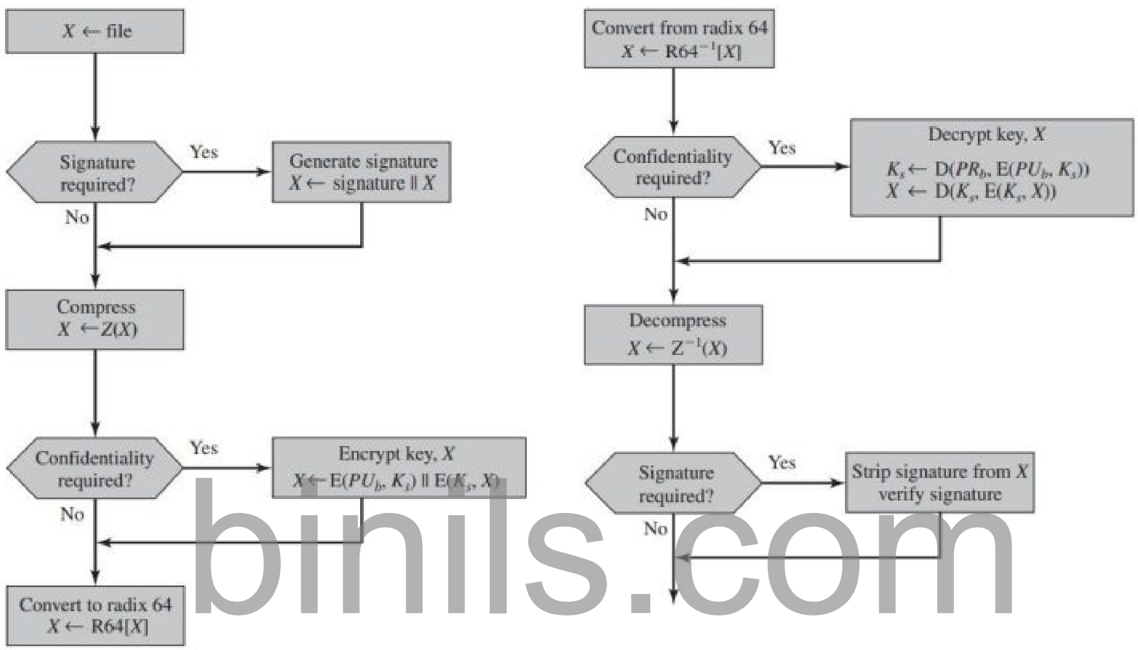
- 1. The sender creates a message.
- 2. SHA-1 is used to generate a 160-bit hash code of the message.
- 3. The hash code is encrypted with RSA using the sender's private key, and the result is prepended to the message.
- 4. The receiver uses RSA with the sender's public key to decrypt and recover the hash code.
- 5. The receiver generates a new hash code for the message and compares it with the decrypted hash code. If the two match, the message is accepted as authentic

CONFIDENTIALITY

- 1. The sender generates a message and a random 128-bit number to be used as a session key for this message only.
- 2. The message is encrypted using CAST-128 (or IDEA or 3DES) with the session key.

- 3. The session key is encrypted with RSA using the recipient's public key and is prepended to the message.
- 4. The receiver uses RSA with its private key to decrypt and recover the session key.
- 5. The session key is used to decrypt the message.

TRANSMISSION AND RECEPTION OF PGP MESSAGES



(a) Generic transmission diagram (from A)

(b) Generic reception diagram (to B)

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

S/MIME

- Secure/Multipurpose Internet Mail Extension (S/MIME) is a security enhancement to the MIME Internet e-mail format standard based on technology from RSA Data Security.
- Although both PGP and S/MIME are on an IETF standards track, it appears likely that S/MIME will emerge as the industry standard for commercial and organizational use, while PGP will remain the choice for personal e-mail security for many users.
- S/MIME is defined in a number of documents—most importantly RFCs 3370, 3850, 3851, and 3852.

RFC 5322

- RFC 5322 defines a format for text messages that are sent using electronic mail.

- It has been the standard for Internet-based text mail messages and remains in common use. In the RFC 5322 context, messages are viewed as having an envelope and contents.
- The envelope contains whatever information is needed to accomplish transmission and delivery. The contents compose the object to be delivered to the recipient.
- The RFC 5322 standard applies only to the contents. However, the content standard includes a set of header fields that may be used by the mail system to create the envelope, and the standard is intended to facilitate the acquisition of such information by programs.
- The overall structure of a message that conforms to RFC 5322 is very simple. A message consists of some number of header lines (the header) followed by unrestricted text (the body).
- The header is separated from the body by a blank line. Put differently, a message is ASCII text, and all lines up to the first blank line are assumed to be header lines used by the user agent part of the mail system
- A header line usually consists of a keyword, followed by a colon, followed by the keyword's arguments; the format allows a long line to be broken up into several lines.
- The most frequently used keywords are From,To, Subject, and Date. Here is an example message:


```
Date: October 8, 2009 2:15:49 PM EDT
From: "William Stallings" <ws@shore.net>
Subject: The Syntax in RFC 5322
To: Smith@Other-host.com
Cc: Jones@Yet-Another-Host.com

Hello. This section begins the actual
message body, which is delimited from the
message heading by a blank line.
```

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

MULTIPURPOSE INTERNET MAIL EXTENSIONS

- Multipurpose Internet Mail Extension (MIME) is an extension to the RFC 5322 framework that is intended to address some of the problems and limitations of the use of Simple Mail Transfer Protocol (SMTP), defined in RFC 821, or some other mail transfer protocol and RFC 5322 for electronic mail.
- Limitations of the SMTP/5322 scheme.
- 1. SMTP cannot transmit executable files or other binary objects. A number of schemes are in use for converting binary files into a text form that can be used by SMTP mail systems, including the popular UNIX UUencode/UUdecode scheme. However, none of these is a standard or even a de facto standard.

- 2. SMTP cannot transmit text data that includes national language characters, because these are represented by 8-bit codes with values of 128 decimal or higher, and SMTP is limited to 7-bit ASCII.
- 3. SMTP servers may reject mail message over a certain size.
- 4. SMTP gateways that translate between ASCII and the character code EBCDIC do not use a consistent set of mappings, resulting in translation problems.
- 5. SMTP gateways to X.400 electronic mail networks cannot handle nontextual data included in X.400 messages.
- 6. Some SMTP implementations do not adhere completely to the SMTP standards defined in RFC 821. Common problems include:
- Deletion, addition, or reordering of carriage return and linefeed
- Truncating or wrapping lines longer than 76 characters
- Removal of trailing white space (tab and space characters)
- Padding of lines in a message to the same length
- Conversion of tab characters into multiple space characters

OVERVIEW

- The MIME specification includes the following elements.
- 1. Five new message header fields are defined, which may be included in an RFC 5322 header. These fields provide information about the body of the message.
- 2. A number of content formats are defined, thus standardizing representations that support multimedia electronic mail.
- 3. Transfer encodings are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system.

HEADER FIELDS

- The five header fields defined in MIME are
- MIME-Version: Must have the parameter value 1.0. This field indicates that the message conforms to RFCs 2045 and 2046.
- Content-Type: Describes the data contained in the body with sufficient detail that the receiving user agent can pick an appropriate agent or mechanism to represent the data to the user or otherwise deal with the data in an appropriate manner.
- Content-Transfer-Encoding: Indicates the type of transformation that has been used to represent the body of the message in a way that is acceptable for mail transport.

- Content-ID: Used to identify MIME entities uniquely in multiple contexts.
- Content-Description: A text description of the object with the body; this is useful when the object is not readable (e.g., audio data).

MIME CONTENT TYPES

Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

S/MIME FUNCTIONALITY

- In terms of general functionality, S/MIME is very similar to PGP. Both offer the ability to sign and/or encrypt messages

FUNCTIONS

- Enveloped data: This consists of encrypted content of any type and encrypted content encryption keys for one or more recipients.
- Signed data: A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer. The content plus

Type	Subtype	Description
Text	Plain	Unformatted text; may be ASCII or ISO 8859.
	Enriched	Provides greater format flexibility.
Multipart	Mixed	The different parts are independent but are to be transmitted together. They should be presented to the receiver in the order that they appear in the mail message.
	Parallel	Differs from Mixed only in that no order is defined for delivering the parts to the receiver.
	Alternative	The different parts are alternative versions of the same information. They are ordered in increasing faithfulness to the original, and the recipient's mail system should display the "best" version to the user.
	Digest	Similar to Mixed, but the default type/subtype of each part is message/rfc822.
Message	rfc822	The body is itself an encapsulated message that conforms to RFC 822.
	Partial	Used to allow fragmentation of large mail items, in a way that is transparent to the recipient.
	External-body	Contains a pointer to an object that exists elsewhere.
Image	jpeg	The image is in JPEG format, JFIF encoding.
	gif	The image is in GIF format.
Video	mpeg	MPEG format.
Audio	Basic	Single-channel 8-bit ISDN mu-law encoding at a sample rate of 8 kHz.
Application	PostScript	Adobe Postscript format.
	octet-stream	General binary data consisting of 8-bit bytes.

signature are then encoded using base64 encoding. A signed data message can only be viewed by a recipient with S/MIME capability.

- Clear-signed data: As with signed data, a digital signature of the content is formed. However, in this case, only the digital signature is encoded using base64.As a result, recipients without S/MIME capability can view the message content, although they cannot verify the signature.
- Signed and enveloped data: Signed-only and encrypted-only entities may be nested, so that encrypted data may be signed and signed data or clear-signed data may be encrypted

INTRUDERS

A significant security problem for networked systems is hostile, or at least unwanted, trespass being unauthorized login or use of a system, by local or remote users; or by software such as a virus, worm, or Trojan horse.

One of the two most publicized threats to security is the intruder (or hacker or cracker), which Anderson identified three classes of:

• Masquerader: An individual who is not authorized to use the computer (outsider)

• Misfeasor: A legitimate user who accesses unauthorized data, programs, or resources (insider)

• Clandestine user: An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection (either)

Intruder attacks range from the benign (simply exploring net to see what is there); to the serious (who attempt to read privileged data, perform unauthorized modifications, or disrupt system).

The intruder threat has been well publicized, particularly because of the famous "Wily Hacker" incident of 1986–1987, documented by Cliff Stoll. Intruder attacks range from the benign to the serious. At the benign end of the scale, there are many people who simply wish to explore internets and see what is out there. At the serious end are individuals who are attempting to read privileged data, perform unauthorized modifications to data, or disrupt the system.

One of the results of the growing awareness of the intruder problem has been the establishment of a number of computer emergency response teams (CERTs). These cooperative ventures collect information about system vulnerabilities and disseminate it to systems managers.

The techniques and behavior patterns of intruders are constantly shifting, to exploit newly discovered weaknesses and to evade detection and countermeasures. Even so, intruders typically follow one of a number of recognizable behavior patterns, and these patterns typically differ from those of ordinary users.

Examples of Intrusion

GRAN04] lists the following examples of intrusion:

• Performing a remote root compromise of an e-mail server

• Defacing a Web server

• Guessing and cracking passwords

• Copying a database containing credit card numbers

• Viewing sensitive data, including payroll records and medical information, without authorization

• Running a packet sniffer on a workstation to capture usernames and passwords

• Using a permission error on an anonymous FTP server to distribute pirated software and music files

• Dialing into an unsecured modem and gaining internal network access

• Posing as an executive, calling the help desk, resetting the executive's e-mail password, and learning the new password

• Using an unattended, logged-in workstation without permission

Hackers

Traditionally, those who hack into computers do so for the thrill of it or for status. The hacking community is a strong meritocracy in which status is determined by level of competence. Thus, attackers often look for targets of opportunity, and then share the information with others. Benign intruders might be tolerable, although they do consume resources and may slow performance for legitimate users. However, there is no way in advance to know whether an intruder will be benign or malign. Consequently, even for systems with no particularly sensitive resources, there is a motivation to control this problem.

Intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) are designed to counter this type of hacker threat. In addition to using such systems, organizations can consider restricting remote logons to specific IP addresses and/or use virtual private network technology.

One of the results of the growing awareness of the intruder problem has been the establishment of a number of computer emergency response teams (CERTs). These cooperative ventures collect information about system vulnerabilities and disseminate it to systems managers. Unfortunately, hackers can also gain access to CERT reports. Thus, it is important for system administrators to quickly insert all software patches to discovered vulnerabilities.

Hacker Behavior Example

The techniques and behavior patterns of intruders are constantly shifting, to exploit newly discovered weaknesses and to evade detection and countermeasures. Even so, intruders typically follow one of a number of recognizable behavior patterns, and these patterns typically differ from those of ordinary users. Table 20.1a, based on [RADC04] summarizes an example of the behavior of hackers. This example is a break-in at a large financial institution.

The intruder took advantage of the fact that the corporate network was running unprotected services, some of which were not even needed. In this case, the key to the break-in was the pcAnywhere application. The manufacturer, Symantec, advertises this program as a remote control solution that enables secure connection to remote devices. But the attacker had an easy time gaining access to pcAnywhere; the administrator used the same three-letter username and password for the program. In this case, there was no intrusion detection system on the 700-node corporate network. The intruder was only discovered when a vice president walked into her office and saw the cursor moving files around on her Windows workstation.

1. select target using IP lookup tools
2. map network for accessible services
3. identify potentially vulnerable services
4. brute force (guess) passwords
5. install remote administration tool
6. wait for admin to log on and capture password
7. use password to access remainder of network

Criminal Enterprise

Organized groups of hackers have become a widespread and common threat to Internet-based systems. These groups can be in the employ of a corporation or government, but often are loosely affiliated gangs of hackers. Typically, these gangs are young, often Eastern European or Russian hackers who do business on the Web. They meet in underground forums with names like DarkMarket.org and theftservices.com to trade tips and data and coordinate attacks. A common target is a credit card file at an e-commerce server. Attackers attempt to gain root access. The card numbers are used by organized crime gangs to purchase expensive items, and are then posted to carder sites, where others can access and use the account numbers; this obscures usage patterns and complicates investigation.

Whereas traditional hackers look for targets of opportunity, criminal hackers usually have specific targets, or at least classes of targets in mind. Once a site is penetrated, the attacker acts quickly, scooping up as much valuable information as possible and exiting.

IDSs and IPSs can also be used for these types of attackers, but may be less effective because of the quick in-and-out nature of the attack. For e-commerce sites, database encryption should be used for sensitive customer information, especially credit cards. For hosted e-commerce sites (provided by an outsider service), the e-commerce organization should make use of a dedicated server (not used to support multiple customers) and closely monitor the provider's security services.

- organized groups of hackers now a threat
    - corporation / government / loosely affiliated gangs
    - typically young
    - often Eastern European or Russian hackers
    - often target credit cards on e-commerce server
- criminal hackers usually have specific targets
- once penetrated act quickly and get out
- IDS / IPS help but less effective
- sensitive data needs strong protection

Criminal Enterprise Behavior

1. act quickly and precisely to make their activities harder to detect
2. exploit perimeter via vulnerable ports
3. use trojan horses (hidden software) to leave back doors for re-entry
4. use sniffers to capture passwords
5. do not stick around until noticed
6. make few or no mistakes.

Insider Attacks

Insider attacks are among the most difficult to detect and prevent. Employees already have access and knowledge about the structure and content of corporate databases. Insider attacks can be motivated by revenge of simply a feeling of entitlement. An example of the former is the case of Kenneth Patterson, fired from his position as data communications manager for American Eagle Outfitters. Patterson disabled the company's ability to process credit card purchases during five days of the holiday season of 2002.

As for a sense of entitlement, there have always been many employees who felt entitled to take extra office supplies for home use, but this now extends to corporate data. An example is that of a vice president of sales for a stock analysis firm who quit to go to a competitor. Before she left, she copied the customer database to take with her. The offender reported feeling no animus toward her former employee; she simply wanted the data because it would be useful to her.

Although IDS and IPS facilities can be useful in countering insider attacks, other more direct approaches are of higher priority. Examples include: enforcing least privilege, monitor logs, protect sensitive resources with strong authentication, on termination delete employee's computer and network access and make a mirror image of employee's hard drive before reissuing it.

- among most difficult to detect and prevent
- employees have access & systems knowledge
- may be motivated by revenge / entitlement
  - when employment terminated

- taking customer data when move to competitor

- IDS / IPS may help but also need:
  - least privilege, monitor logs, strong authentication, termination process to block access & mirror data

Insider Behavior Example

1. create network accounts for themselves and their friends
2. access accounts and applications they wouldn't normally use for their daily jobs
3. e-mail former and prospective employers
4. conduct furtive instant-messaging chats
5. visit web sites that cater to disgruntled employees, such as f'dcompany.com
6. perform large downloads and file copying
7. access the network during off hours

Intrusion Techniques

The objective of the intruder is to gain access to a system or to increase the range of privileges accessible on a system. Most initial attacks use system or software vulnerabilities that allow a user to execute code that opens a back door into the system. Alternatively, the intruder attempts to acquire information that should have been protected. In some cases, this information is in the form of a user password. With knowledge of some other user's password, an intruder can log in to a system and exercise all the privileges accorded to the legitimate user.

Knowing the standard attack methods is a key element in limiting your vulnerability. The basic aim is to gain access and/or increase privileges on some system. The basic attack methodology list is taken from McClure et al "Hacking Exposed".

- aim to gain access and/or increase privileges on a system
- often use system / software vulnerabilities
- key goal often is to acquire passwords

- so then exercise access rights of owner
- basic attack methodology
  - target acquisition and information gathering
  - initial access
  - privilege escalation
  - covering tracks

Password Guessing

Password guessing is a common attack. If an attacker has obtained a poorly protected password file, then can mount attack off-line, so target is unaware of its progress. Some O/S take less care than others with their password files. If have to actually attempt to login to check guesses, then system should detect an abnormal number of failed logins, and hence trigger appropriate countermeasures by admins/security. Likelihood of success depends very much on how well the passwords are chosen. Unfortunately, users often don't choose well.

- one of the most common attacks
- attacker knows a login (from email/web page etc)
- then attempts to guess password for it
  - defaults, short passwords, common word searches
  - user info (variations on names, birthday, phone, common words/interests)
  - exhaustively searching all possible passwords
- check by login or against stolen password file
- success depends on password chosen by user
- surveys show many users choose poorly

Password Capture

There is also a range of ways of "capturing" a login/password pair, from the low-tech looking over the shoulder, to the use of Trojan Horse programs (eg. game program or nifty utility with a covert function as well as the overt behaviour), to sophisticated network monitoring tools, or extracting recorded info after a successful login - say from web history or cache, or last number dialed memory on phones etc. Need to educate users to be aware of whose around, to check they really are interacting with the computer system (trusted path), to beware of unknown source s/w, to use secure network connections (HTTPS, SSH, SSL), to flush browser/phone histories after use etc.

- another attack involves **password capture**
    - watching over shoulder as password is entered
    - using a trojan horse program to collect
    - monitoring an insecure network login
        - eg. telnet, FTP, web, email
    - extracting recorded info after successful login (web history/cache, last number dialed etc)
- using valid login/password can impersonate user
- users need to be educated to use suitable precautions/countermeasures

IP SECURITY

- IP security (IPsec) is a capability that can be added to either current version of the Internet Protocol (IPv4 or IPv6) by means of additional headers.
- IPsec encompasses three functional areas: authentication, confidentiality, and key management.
- Authentication makes use of the HMAC message authentication code. Authentication can be applied to the entire original IP packet (tunnel mode) or to all of the packet except for the IP header (transport mode).
- Confidentiality is provided by an encryption format known as encapsulating security payload. Both tunnel and transport modes can be accommodated.
- IKE defines a number of techniques for key management.

IP SECURITY OVERVIEW

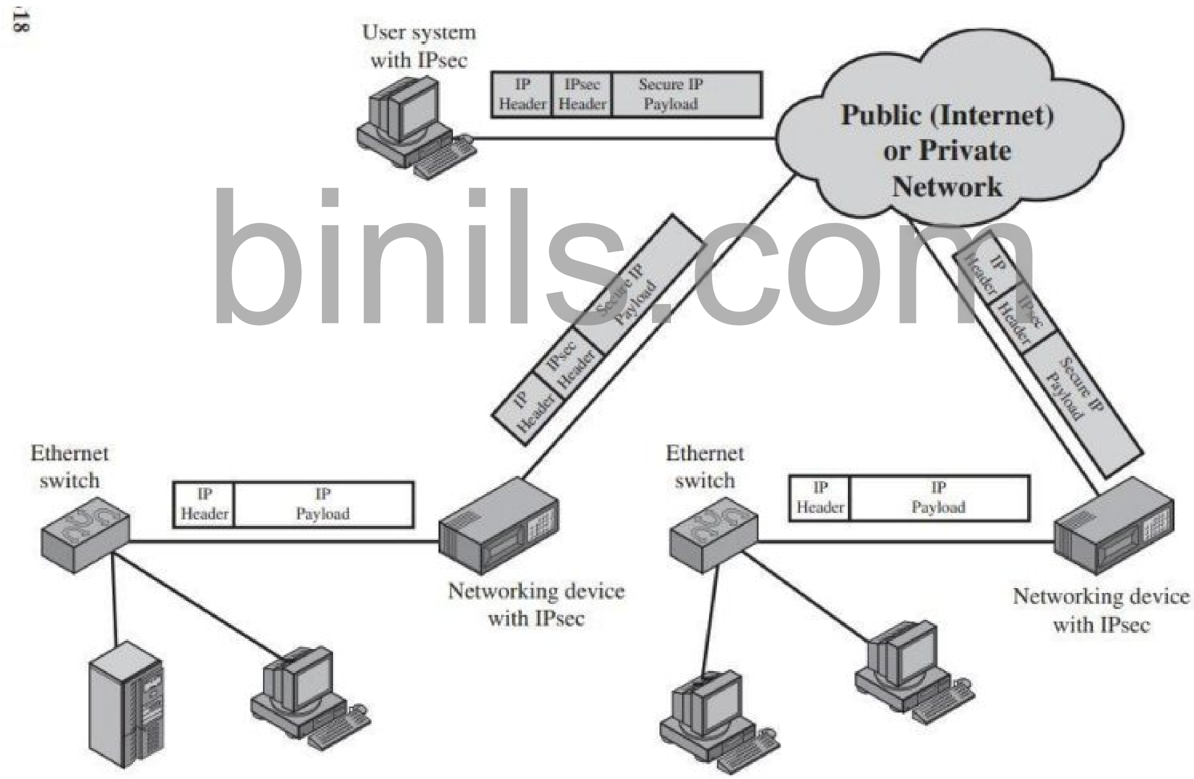
- In 1994, the Internet Architecture Board (IAB) issued a report titled "Security in the Internet Architecture" (RFC 1636).
- The report identified key areas for security mechanisms.
- Among these were the need to secure the network infrastructure from unauthorized monitoring and control of network traffic and the need to secure end-user-to-end-user traffic using authentication and encryption mechanisms.
- To provide security, the IAB included authentication and encryption as necessary security features in the next-generation IP, which has been issued as IPv6.
- Fortunately, these security capabilities were designed to be usable both with the current IPv4 and the future IPv6.
- This means that vendors can begin offering these features now, and many vendors now do have some IPsec capability in their products. The IPsec specification now exists as a set of Internet standards.

APPLICATIONS OF IPSEC

- IPsec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet.
- Examples of its use include:
- Secure branch office connectivity over the Internet: A company can build a secure virtual private network over the Internet or over a public WAN. This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead.

- Secure remote access over the Internet: An end user whose system is equipped with IP security protocols can make a local call to an Internet Service Provider (ISP) and gain secure access to a company network.This reduces the cost of toll charges for traveling employees and telecommuters.
- Establishing extranet and intranet connectivity with partners: IPsec can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing a key exchange mechanism.
- Enhancing electronic commerce security: Even though some Web and electronic commerce applications have built-in security protocols, the use of IPsec enhances that security. IPsec guarantees that all traffic designated by the network administrator is both encrypted and authenticated, adding an additional layer of security to whatever is provided at the application layer.

AN IP SECURITY SCENARIO



Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

BENEFITS OF IPSEC

- Some of the benefits of IPsec:

- When IPsec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter. Traffic within a company or workgroup does not incur the overhead of security-related processing.
- IPsec in a firewall is resistant to bypass if all traffic from the outside must use IP and the firewall is the only means of entrance from the Internet into the organization.
- IPsec is below the transport layer (TCP, UDP) and so is transparent to applications. There is no need to change software on a user or server system when IPsec is implemented in the firewall or router. Even if IPsec is implemented in end systems, upper-layer software, including applications, is not affected.
- IPsec can be transparent to end users. There is no need to train users on security mechanisms, issue keying material on a per-user basis, or revoke keying material when users leave the organization.

IPsec can provide security for individual users if needed. This is useful for offsite workers and for setting up a secure virtual subnetwork within an organization for sensitive applications.

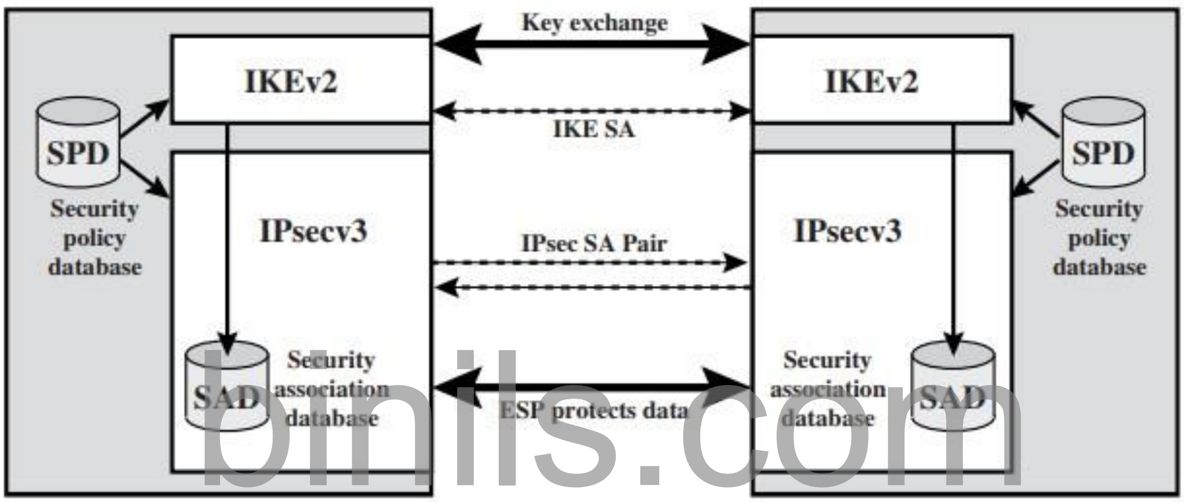
IPSEC DOCUMENTS

- Architecture: Covers the general concepts, security requirements, definitions, and mechanisms defining IPsec technology. The current specification is RFC 4301, Security Architecture for the Internet Protocol.
- Authentication Header (AH): AH is an extension header to provide message authentication. The current specification is RFC 4302, IP Authentication Header. Because message authentication is provided by ESP, the use of AH is deprecated. It is included in IPsecv3 for backward compatibility but should not be used in new applications. We do not discuss AH in this chapter.
- Encapsulating Security Payload (ESP): ESP consists of an encapsulating header and trailer used to provide encryption or combined encryption/authentication. The current specification is RFC 4303,IP Encapsulating Security Payload (ESP).
- Internet Key Exchange (IKE): This is a collection of documents describing the key management schemes for use with IPsec. The main specification is RFC 4306,Internet Key Exchange (IKEv2) Protocol, but there are a number of related RFCs.
- Cryptographic algorithms: This category encompasses a large set of documents that define and describe cryptographic algorithms for encryption, message authentication, pseudorandom functions (PRFs), and cryptographic key exchange.
- Other: There are a variety of other IPsec-related RFCs, including those dealing with security policy and management information base (MIB) content.

IPSEC SERVICES

- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets (a form of partial sequence integrity)
- Confidentiality (encryption)
- Limited traffic flow confidentiality

IP SECURITY POLICY



Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

- A security association is uniquely identified by three parameters.
- Security Parameters Index (SPI): A bit string assigned to this SA and having local significance only. The SPI is carried in AH and ESP headers to enable the receiving system to select the SA under which a received packet will be processed.
- IP Destination Address: This is the address of the destination endpoint of the SA, which may be an end-user system or a network system such as a firewall or router.
- Security Protocol Identifier: This field from the outer IP header indicates whether the association is an AH or ESP security association. Hence, in any IP packet, the security association is uniquely identified by the Destination Address in the IPv4 or IPv6 header and the SPI in the enclosed extension header (AH or ESP).

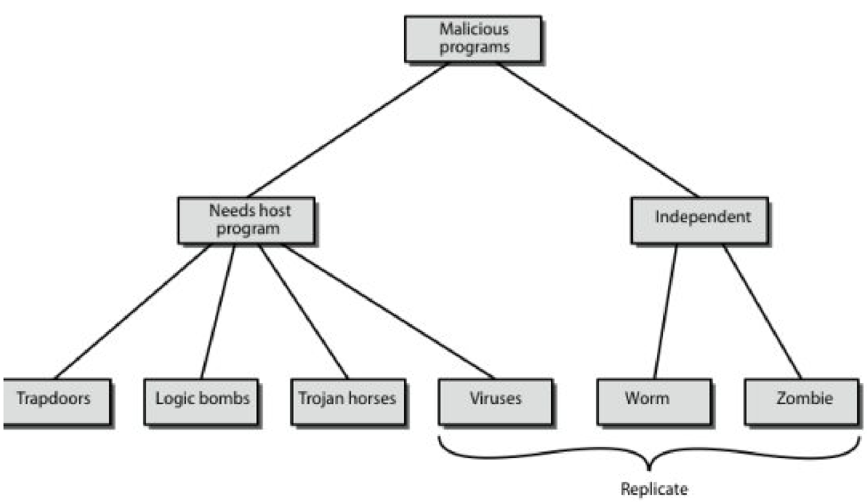
MALICIOUS SOFTWARE

This chapter examines malicious software (malware), especially viruses and worms, which exploit vulnerabilities in computing systems. These have been given a lot of (often uninformed) comment in the general media. They are however, of serious concern, and are perhaps the most sophisticated types of threats to computer systems. We begin with a survey of various types of malware, with a more detailed look at the nature of viruses and worms. We then turn to distributed denial-of-service attacks.

Viruses and Other Malicious Content

- computer viruses have got a lot of publicity
- one of a family of **malicious software**
- effects usually obvious
- have figured in news reports, fiction, movies (often exaggerated)
- getting more attention than deserve
- are a concern though

Malicious Software



Reference :William Stallings, Cryptography and Network Security: Principles and Practice, PHI 3rd Edition, 2006

The terminology used for malicious software presents problems because of a lack of universal agreement on all terms and because of overlap. Stallings Table 21.1, and this diagram from 3/e, provide a useful taxonomy. It can be divided into two categories: those that need a host program (being a program fragment eg virus), and those that are independent programs (eg worm); alternatively you can also differentiate between those software threats that do not replicate (are activated by a trigger) and those that do (producing copies of themselves). Will now survey this range of malware.

Backdoor or Trapdoor

A backdoor, or trapdoor, is a secret entry point into a program that allows someone that is aware of it to gain access without going through the usual security access procedures. Have been used legitimately for many years to debug and test programs, but become a threat when left in production programs, allowing intruders to gain unauthorized access. It is difficult to implement operating system controls for backdoors. Security measures must focus on the program development and software update activities.

Logic Bomb

One of the oldest types of program threat, predating viruses and worms, is the logic bomb. The logic bomb is code embedded in some legitimate program that is set to "explode" when certain conditions are met. Examples of conditions that can be used as triggers for a logic bomb are the presence or absence of certain files, a particular day of the week or date, or a particular user running the application. Once triggered, a bomb may alter or delete data or entire files, cause a machine halt, or do some other damage.

Trojan Horse

A Trojan horse is a useful, or apparently useful, program or command procedure (eg game, utility, s/w upgrade etc) containing hidden code that performs some unwanted or harmful function that an unauthorized user could not accomplish directly. Commonly used to make files readable, propagate a virus or worm or backdoor, or simply to destroy data.

Mobile Code

Mobile code refers to programs (e.g., script, macro, or other portable instruction) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics. The term also applies to situations involving a large homogeneous collection of platforms (e.g., Microsoft Windows). Mobile code is transmitted from a remote system to a local system and then executed on the local system without the user's explicit instruction. Mobile code often acts as a mechanism for a virus, worm, or Trojan horse to be transmitted to the user's workstation. In other cases, mobile code takes advantage of vulnerabilities to perform its own exploits, such as unauthorized data access or root compromise. Popular vehicles for mobile code include Java applets, ActiveX, JavaScript, and VBScript. The most common ways of using mobile code for malicious operations on local system are cross-site scripting, interactive and dynamic Web sites, e-mail attachments, and downloads from untrusted sites or of untrusted software.

Multiple-Threat Malware

Viruses and other malware may operate in multiple ways.

A **multipartite** virus infects in multiple ways. Typically, the multipartite virus is capable of infecting multiple types of files, so that virus eradication must deal with all of the possible sites of infection.

A **blended** attack uses multiple methods of infection or transmission, to maximize the speed of contagion and the severity of the attack. Some writers characterize a blended attack as a package that includes multiple types of malware. An example of a blended attack is the Nimda attack, erroneously referred to as simply a worm. Nimda has worm, virus, and mobile code characteristics. Blended attacks may also spread through other services, such as instant messaging and peer-to-peer file sharing.

Viruses

A virus is a piece of software that can "infect" other programs by modifying them; the modification includes a copy of the virus program, which can then go on to infect other programs. A virus can do anything that other programs do. The difference is that a virus attaches itself to another program and executes secretly when the host program is run. Once a virus is executing, it can perform any function, such as erasing files and programs. Most viruses carry out their work in a manner that is specific to a particular operating system and, in some cases, specific to a particular hardware platform. Thus, they are designed to take advantage of the details and weaknesses of particular systems. During its lifetime, a typical virus goes through the following four phases:

• **Dormant phase:** The virus is idle. The virus will eventually be activated by some event, such as a date, the presence of another program or file, or the capacity of the disk exceeding some limit. Not all viruses have this stage.

• **Propagation phase:** The virus places an identical copy of itself into other programs or into certain system areas on the disk. Each infected program will now contain a clone of the virus, which will itself enter a propagation phase.

• **Triggering phase:** The virus is activated to perform the function for which it was intended. As with the dormant phase, the triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself.

• **Execution phase:** The function is performed, which may be harmless, e.g. a message on the screen, or damaging, e.g. the destruction of programs and data files

Virus Structure

A computer virus has three parts [AYCO06]:

• Infection mechanism:The means by which a virus spreads, enabling it to replicate. The mechanism is also referred to as the infection vector.

• Trigger: event or condition determining when the payload is activated or delivered.

• Payload: What the virus does, besides spreading. The payload may involve damage or may involve benign but noticeable activity.

A virus can be prepended or postpended to an executable program, or it can be embedded in some other fashion. The key to its operation is that the infected program, when invoked, will first execute the virus code and then execute the original code of the program.

Once a virus has gained entry to a system by infecting a single program, it is in a position to infect some or all other executable files on that system when the infected program executes. Thus, viral infection can be completely prevented by preventing the virus from gaining entry in the first place. Unfortunately, prevention is extraordinarily difficult because a virus can be part of any program outside a system. Thus, unless one is content to take an absolutely bare piece of iron and write all one's own system and application programs, one is vulnerable. The lack of access controls on early PCs is a key reason why traditional machine code based viruses spread rapidly on these systems. In contrast, while it is easy enough to write a machine code virus for UNIX systems, they were almost never seen in practice due to the existence of access controls on these systems prevented effective propagation of the virus.

Virus Structure

```
    program V :=

{goto main;
    1234567;

    subroutine infect-executable :=
        {loop:
        file := get-random-executable-file;
        if (first-line-of-file = 1234567)
           then goto loop
           else prepend V to file; }

    subroutine do-damage :=
        {whatever damage is to be done}

    subroutine trigger-pulled :=
        {return true if some condition holds}

main:   main-program :=
        {infect-executable;
        if trigger-pulled then do-damage;
        goto next;}

next:

}
```


A very general depiction of virus structure is shown in Figure. In this case, the virus code, V, is prepended to infected programs, and it is assumed that the entry point to the program, when invoked, is the first line of the program. An infected program begins with the virus code and works as follows. The first line of code is a jump to the main virus program. The second line is a special marker that is used by the virus to determine whether or not a potential victim program has already been infected with this virus. When the program is invoked, control is immediately transferred to the main virus program. The virus program first seeks out uninfected executable files and infects them. Next, the virus may perform some action, usually detrimental to the system. This action could be performed every time the program is invoked, or it could be a logic bomb that triggers only under certain conditions. Finally, the virus transfers control to the original program. If the infection phase of the program is reasonably rapid, a user is unlikely to notice any difference between the execution of an infected and uninfected program.

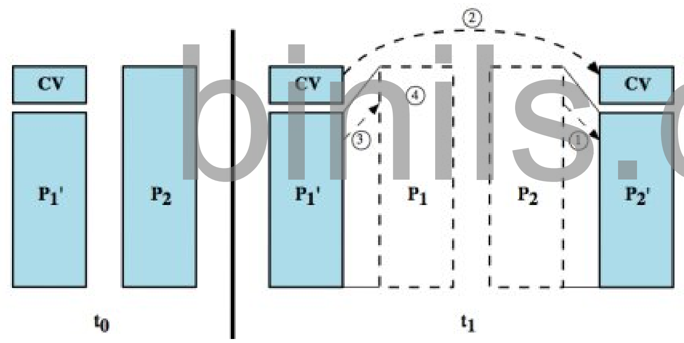
Compression Virus

```
    program CV :=

{goto main;
    01234567;

    subroutine infect-executable :=
          {loop:
               file := get-random-executable-file;
          if (first-line-of-file = 01234567) then goto loop;
        (1)     compress file;
        (2)     prepend CV to file;
        }

main:   main-program :=
          {if ask-permission then infect-executable;
        (3)     uncompress rest-of-file;
        (4)     run uncompressed file;}
        }
```



A virus such as the one just described is easily detected because an infected version of a program is longer than the corresponding uninfected one. A way to thwart such a simple means of detecting a virus is to compress the executable file so that both the infected and uninfected versions are of identical length. The code shown from Figure 21.2 shows in general terms the logic required. The key lines in this virus are numbered, and Figure 21.3 illustrates the operation. In this example, the virus does nothing other than propagate. As in the previous example, the virus may include a logic bomb. We assume that program P$_1$ is infected with the virus CV. When this program is invoked, control passes to its virus, which performs the following steps:

1. For each uninfected file P$_2$ that is found, the virus first compresses that file to produce , which is shorter than the original program by the size of the virus.

2. A copy of the virus is prepended to the compressed program.

3. The compressed version of the original infected program, , is uncompressed.

4. The uncompressed original program is executed

Virus Classification

There has been a continuous arms race between virus writers and writers of antivirus software since viruses first appeared. As effective countermeasures have been developed for existing types of viruses, new types have been developed. A virus classification by target includes the following categories:

• Boot sector infector:Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus.

• File infector: Infects files that operating system or shell consider to be executable.

• Macro virus: Infects files with macro code that is interpreted by an application.

A virus classification by concealment strategy includes the following categories:

• Encrypted virus: the virus creates a random encryption key, stored with the virus, and encrypts the remainder of the virus. When an infected program is invoked, the virus uses the stored random key to decrypt the virus. When the virus replicates, a different random key is selected.

• Stealth virus: A form of virus explicitly designed to hide itself from detection by antivirus software. Thus,the entire virus, not just a payload is hidden.

• Polymorphic virus: A virus that mutates with every infection, making detection by the "signature"of the virus impossible.

• Metamorphic virus: As with a polymorphic virus ,a metamorphic virus mutates with every infection. The difference is that a metamorphic virus rewrites itself completely at each iteration, increasing the difficulty of detection. Metamorphic viruses may change their behavior as well as their appearance.

Macro Virus

In the mid-1990s, macro viruses became by far the most prevalent type of virus. Macro viruses are particularly threatening for a number of reasons:

1. A macro virus is platform independent. Virtually all of the macro viruses infect Microsoft Word documents. Any hardware platform and operating system that supports Word can be infected.

2. Macro viruses infect documents, not executable portions of code. Most of the information introduced onto a computer system is in the form of a document rather than a program.

3. Macro viruses are easily spread. A very common method is by electronic mail.

Macro viruses take advantage of a feature found in Word and other office applications such as Microsoft Excel, namely the macro. In essence, a macro is an executable program embedded in a word processing document or other type of file. Typically, users employ macros to automate repetitive tasks and thereby save keystrokes. The macro language is usually some form of the Basic programming language. A user might define a sequence of keystrokes in a macro and set it up so that the macro is invoked when a function key or special short combination of keys is input. Successive releases of Word provide increased protection against macro viruses. For example, Microsoft offers an optional Macro Virus Protection tool that detects suspicious Word files and alerts the customer to the potential risk of opening a file with macros. Various antivirus (A/V) product vendors have also developed tools to detect and correct macro viruses. As in other types of viruses, the arms race continues in the field of macro viruses, but they no longer are the predominant virus threat.

E-Mail Viruses

A more recent development in malicious software is the e-mail virus. The first rapidly spreading e-mail viruses, such as Melissa, made use of a Microsoft Word macro embedded in an attachment. If the recipient opens the e-mail attachment, the Word macro is activated. Then the e-mail virus sends itself to everyone on the mailing list in the user's e-mail package, and also does local damage.

At the end of 1999, a more powerful version of the e-mail virus appeared. This newer version can be activated merely by opening an e-mail that contains the virus rather than opening an attachment. The virus uses the Visual Basic scripting language supported by the e-mail package.

Thus we see a new generation of malware that arrives via e-mail and uses e-mail software features to replicate itself across the Internet. The virus propagates itself as soon as activated (either by opening an e-mail attachment of by opening the e-mail) to all of the e-mail addresses known to the infected host. As a result, whereas viruses used to take months or years to propagate, they now do so in hours. This makes it very difficult for antivirus software to respond before much damage is done. Ultimately, a greater degree of security must be built into Internet utility and application software on PCs to counter the growing threat.

Virus Countermeasures

The ideal solution to the threat of viruses is prevention: Do not allow a virus to get into the system in the first place. This goal is, in general, impossible to achieve, although prevention can reduce the number of successful viral attacks. The next best approach is to be able to do the following:

- **Detection:** Once the infection has occurred, determine that it has occurred and locate the virus.
- **Identification:** Once detection has been achieved, identify the specific virus that has infected a program.
- **Removal:** Once the specific virus has been identified, remove all traces of the virus from the infected program and restore it to its original state. Remove the virus from all infected systems so that the disease cannot spread further.

If detection succeeds but either identification or removal is not possible, then the alternative is to discard the infected program and reload a clean backup version.

Anti-Virus Evolution

Advances in virus and antivirus technology go hand in hand. Early viruses were relatively simple code fragments and could be identified and purged with relatively simple antivirus software packages. As the virus arms race has evolved, both viruses and, necessarily, antivirus software have grown more complex and sophisticated. [STEP93] identifies four generations of antivirus software:

A **first-generation** scanner requires a virus signature to identify a virus. The virus may contain "wildcards" but has essentially the same structure and bit pattern in all copies. Such signature-specific scanners are limited to the detection of known viruses.

A **second-generation** scanner uses heuristic rules to search for probable virus infection, e.g to look for fragments of code that are often associated with viruses.. Another second-generation approach is integrity checking, using a hash function rather than a simpler checksum.

Third-generation programs are memory-resident programs that identify a virus by its actions rather than structure in an infected program. These have the advantage that it is not necessary to develop signatures / heuristics, but only to identify the small set of actions indicating an infection is attempted and then intervene.

Fourth-generation products are packages consisting of a variety of antivirus techniques used in conjunction. These include scanning and activity trap components. In addition, such a package includes access control capability, which limits the ability of viruses to penetrate a system and then limits the ability of a virus to update files in order to pass on the infection.

Generic Decryption

More sophisticated antivirus approaches and products continue to appear. In this subsection, we highlight some of the most important.

Generic decryption (GD) technology enables the antivirus program to easily detect even the most complex polymorphic viruses, while maintaining fast scanning speeds. In order to detect encrypted viruses, executable files are run through a GD scanner:

• **CPU emulator:** A software-based virtual computer that interprets instructions in an executable file rather than executing them on the underlying processor.

• **Virus signature scanner:** scans target code looking for known virus signatures.

• **Emulation control module:** Controls the execution of the target code.

At the start of each simulation, the emulator begins interpreting instructions in the target code, one at a time. Thus, if the code includes a decryption routine that decrypts and hence exposes the virus, that code is interpreted. In effect, the virus does the work for the antivirus program by exposing the virus. Periodically, the control module interrupts interpretation to scan the target code for virus signatures. During interpretation, the target code can cause no damage to the actual personal computer environment, because it is being interpreted in a completely controlled environment. The most difficult design issue with a GD scanner is to determine how long to run each interpretation. Typically, virus elements are activated soon after a program begins executing, but this need not be the case. The longer the scanner emulates a particular program, the more likely it is to catch any hidden viruses. However, the antivirus program can take up only a limited amount of time and resources before users complain.