IQware's Approach to Software and IT Security Issues

The Need for Security

Security is essential in business intelligence (BI) systems since they have access to critical and proprietary enterprise information. The expenses associated with cyber attacks fall into four categories:

1. Damage cleanup

2. Business disruption

3. Enterprise data theft and/or corruption

4. Discovering and defusing a software “time bomb”

The proper goal of IT and software security is to ensure that the business operations are tamper-proof and that they will not be interrupted by a cyber attack. Cyber attacks and viruses are problematic because of the potential harm that they can cause. If the IT system - and the component software products - are properly architected, then cyber attacks and viruses will do no damage. This is what IQware does. IQware's application software will work properly - even during a cyber attack - without any data loss or interruption in business operations.

In late January 2003, a cyber attack originated outside the U.S. and did extensive damage. The worm, called “SQL Slammer” plugged network channels with excessive traffic, interfered with Bank ATM communication, and even interfered with Microsoft's internal network. This “worm” exploited weaknesses in Microsoft's operating systems and exploited weaknesses in SQL server and MDSW 2000, which are popular Microsoft applications.

Clearly, such cyber attacks are increasing in frequency and in severity. Future attacks will exploit other “holes” in popular desktop operating systems and applications. Further, the “best-of-breed” anti-virus software, firewalls and intrusion detection systems that failed to defend against SQL Slammer in January 2003 will again be insufficient to defend against the next attack. The last two years have seen too many successful cyber attacks to mention them individually - they're in the news all the time.

IQware Uses a Secure Operating System (O/S)

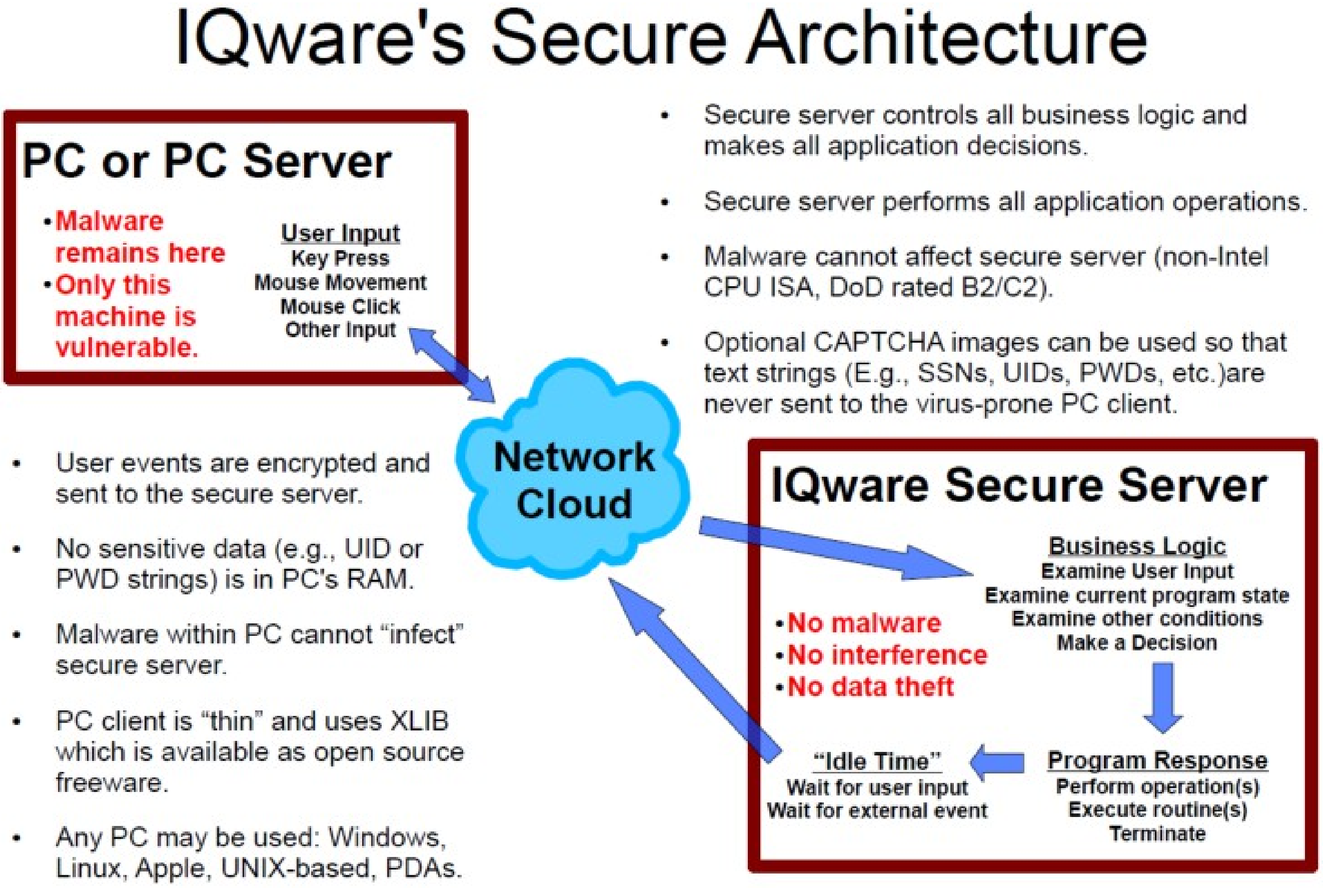
IQware gains its capabilities from its architecture. The significant components are:

- Department of Defense rating of B2; National Information Assurance Partnership (NIAP) Common Criteria (ISO 15408) minimum rating of EAL5.

- A “Reference Monitor” mediates attempts by a subject to gain access to an object. An access control list is maintained as well as a tamper-proof audit trail of security-related events.

- An authorization database serves as a repository of subject and object security attributes, including access modes and allowed operations.
- The IQware application is layered on the secure O/A, using the reference monitor architecture to implement the security policy while providing full accountability, tracking and assurance. This design ensures that the combination of the application and O/S will operate in accordance with the Department of Defense Secure System standards.

IQware uses this Reference Monitor and a secure operating system to create a secure software system. The secure server uses the reference monitor to ensure that the system security policy is implemented properly. All critical software functions and system operations are handled by the IQware application operating on the secure server.



IQware's Secure Architecture

- Secure server controls all business logic and makes all application decisions.
- Secure server performs all application operations.
- Malware cannot affect secure server (non-Intel CPU ISA, DoD rated B2/C2).
- Optional CAPTCHA images can be used so that text strings (E.g., SSNs, UIDs, PWDs, etc.)are never sent to the virus-prone PC client.

PC or PC Server

- Malware remains here
- Only this machine is vulnerable.

User Input
Key Press
Mouse Movement
Mouse Click
Other Input

Network Cloud

- User events are encrypted and sent to the secure server.
- No sensitive data (e.g., UID or PWD strings) is in PC's RAM.
- Malware within PC cannot "infect" secure server.
- PC client is "thin" and uses XLIB which is available as open source freeware.
- Any PC may be used: Windows, Linux, Apple, UNIX-based, PDAs.

IQware Secure Server

- No malware
- No interference
- No data theft

Business Logic
Examine User Input
Examine current program state
Examine other conditions
Make a Decision

Program Response
Perform operation(s)
Execute routine(s)
Terminate

"Idle Time"
Wait for user input
Wait for external event

IQware also uses a "thin client" architecture that puts the critical functionality on the secure server where the majority of the IQware secure application executes. This segregation of software functionality allows IQware to work with common desktop (non-secure) clients without creating any security issues. Any security breach occurring on a client machine will not propagate to the IQware application and will not interfere in any way with its operation. This unique functional assignment between client and server lets IQware provide the best of both worlds: a secure software environment with the convenience of the desktop machine.

All critical data and functional are executed in a secure environment. The clients can be any common desktop, even ones with virus-prone operating systems and/or applications. Computer virus “infections” on the client machines cannot get into the secure environment and will not damage files or interfere with the proper execution of the IQware application. IQware's secure and patent-pending architecture is shown in in the diagram.

The client handles the non-critical user interface functions and miscellaneous interface operations where tampering will not affect the proper execution of the IQware application. This functional segregation is an important part of protecting an IT systems against cyber attacks. Further, this distribution of software functionality lets IQware work with any client device, including PDAs, Linux, Mac OS, Windows, cell phones, etc. As technology changes, this level of software flexibility is the only way that businesses can incorporate new technology without disturbing operations.

Both the operating system and application software must be architected, coded and deployed properly in order to be secure. Patching is ineffective. True-to-form, the SQL Slammer worm that hit in late January 2003 exploited weaknesses in Microsoft's operating systems, SQL Server 2000 and MDSE 2000 software. Even though Microsoft had a patch for this particular security flaw, most IT managers simply did not install it. Patching is so commonplace that it has become impractical for IT managers to install every one of them because they would be modifying their IT systems more than once per day. This “change rate” is impossible to sustain and to manage. Besides, future viruses will surely exploit other existing software flaws for which patches are not yet available. Another question is “how did this worm get through businesses' best-of-breed anti-virus software and firewalls?”

What Secure Systems Must Do

Security research has been ongoing since the 1960s. The research has shown that “putting out security fires” by patching code can never be 100% successful. Rather, software systems must be architected from their foundations in accordance with a secure system model. This is the only way to create a secure and tamper-proof software system.

Secure systems must control access to information and information processing operations. Only properly authorized people are allowed to read, write, create, edit or delete information. Further, only properly authorized processes are allowed to read, write, create, edit or delete information.

The three main characteristics of a secure system are policy, accountability and assurance. These three characteristics must be present in order for a software system to be secure. Each of these critical characteristics is explained in the following paragraphs.

Policy - Secure systems must have a well-defined, clear and practically enforceable security policy. This policy includes object marking so that the sensitivity and allowed access modes of each object are clear and computable. Further, each “object” in the software system (e.g., files, directories, RAM locations, external devices, etc.) must have an access control label that summarizes this information attached to it in a secure way. This access control label is critical to a secure system. The lack of this “object marking” is one of the reasons that popular, virus-prone desktop software cannot be made secure by simply adding other software to it.

Accountability - Secure systems must have accountability so that any data processing actions may be accurately traced to the responsible party. This accountability includes a secure identification method so that system uses can be associated with their authorizations in a secure manner. An audit trail is also required so that all security-related actions can be accurately and securely traced to the responsible party and then recorded for later review and analysis. This audit trail must be continuously maintained and protected.

Assurance - Secure systems must periodically evaluate the “trusted” hardware and software security mechanisms to provide assurance that the security policy is enforced properly. The corresponding mechanisms that handle accountability must also be periodically (and independently) evaluated. These hardware and software mechanisms for policy and accountability must also be continuously protected from tampering and free from external observation. For example, if an unauthorized user (or process) attempts to access a protected set of files or directories, the mechanism that presents this cannot tell the users or process “why” they were denied access. Such an explanation might provide too many clues to hackers about how to get around the mechanism. The table below summarizes the three critical secure system characteristics.

Secure System Characteristics (Summary)

Policy

- **Security Policy** - System must enforce a well-defined security policy
- **Marking** - System must associate all objects with access control labels (sensitivity & access modes).

Accountability

Identification - System must identify individuals and their corresponding authorizations in a secure manner.

Audit Trail - System must keep & protect audit trail so actions may be traced to responsible party.

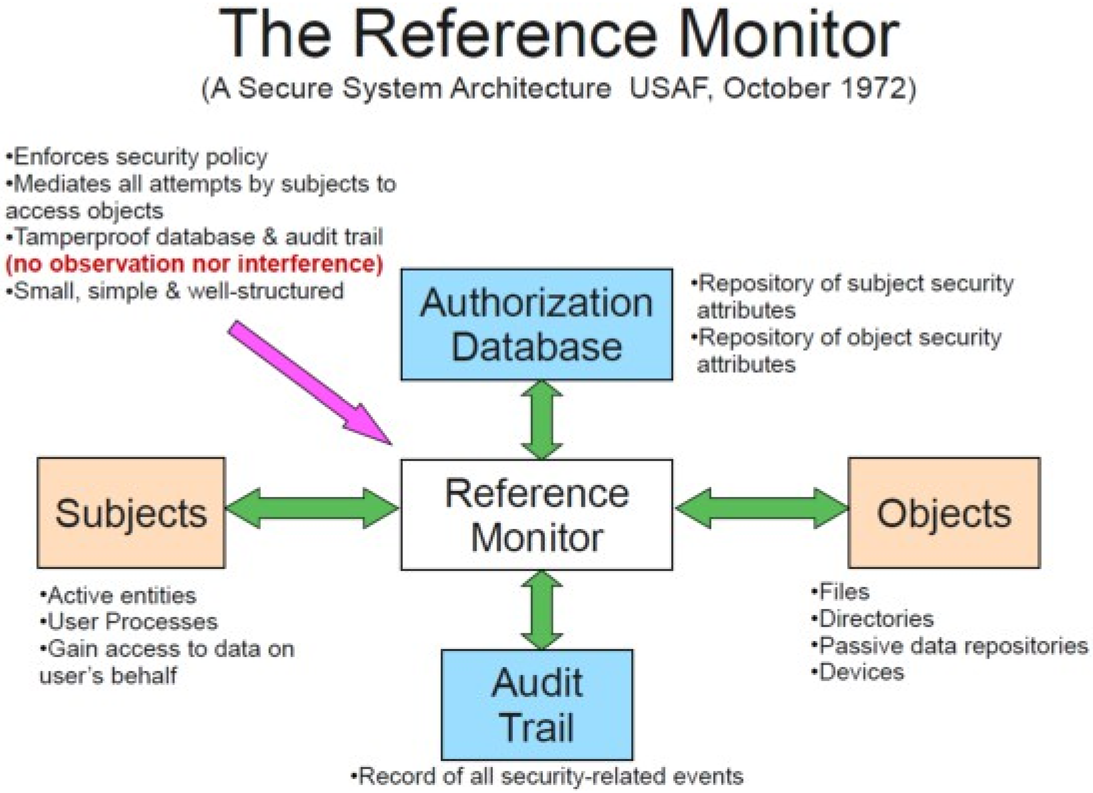
Assurance

Evaluation - System must have hardware/software mechanisms that can be independently evaluated to assure that policy & accountability are enforced.

Continuous Protection - System must continuously protect trusted mechanisms that enforce policy & accountability from tampering.

The Reference Monitor - A Mechanism for a Secure System

A trusted mechanism for a secure system is known as the Reference Monitor. It is an architecture that can be implemented through software, hardware or a combination of the two. This structure implements the security policy by controlling the access of every subject (users, processes, etc.) to every object (files, directories, RAM locations, external devices, etc.) in the software system.


The Reference Monitor
(A Secure System Architecture USAF, October 1972)
•Enforces security policy
•Mediates all attempts by subjects to access objects
•Tamperproof database & audit trail
(no observation nor interference)
•Small, simple & well-structured
Authorization Database
•Repository of subject security attributes
•Repository of object security attributes
Subjects
Reference Monitor
Objects
•Active entities
•User Processes
•Gain access to data on user's behalf
•Files
•Directories
•Passive data repositories
•Devices
Audit Trail
•Record of all security-related events

The reference monitor maintains an authorization database, which contains security, attributes of all subjects and objects. The reference monitor also maintains an audit trail of all security-related events. As mentioned earlier, the reference monitor can be implemented in a variety of ways but it must be tamper-proof and non-observable.

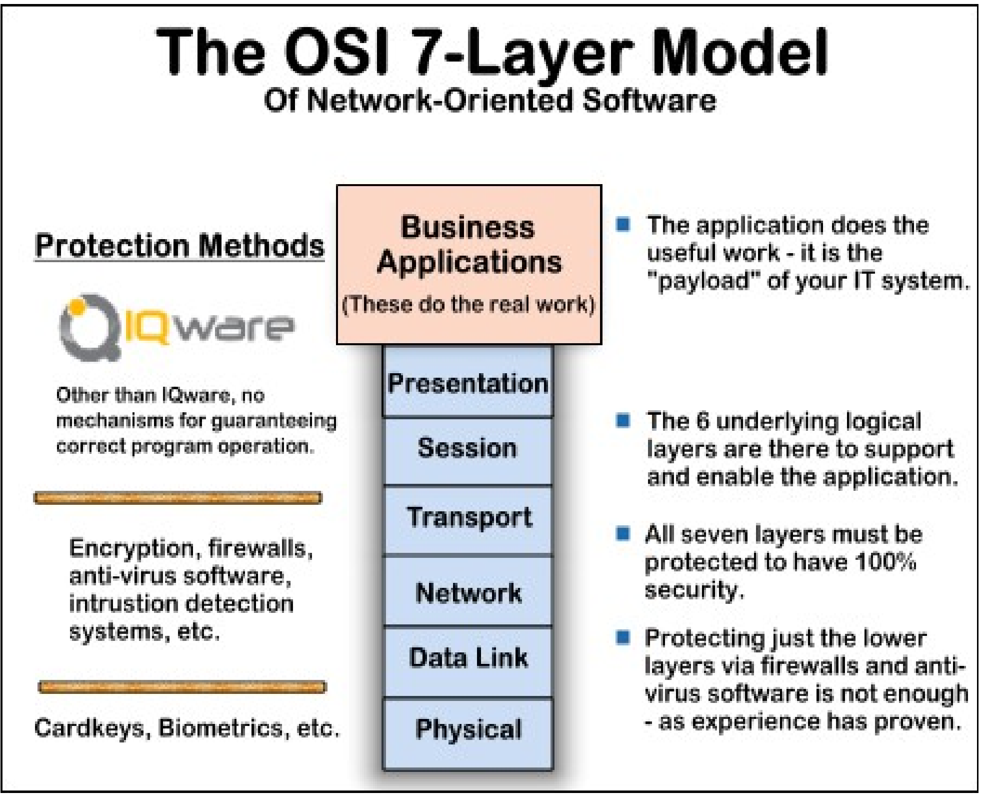
As an example, viruses are generally composed of script files, executable images or a combination of the two. The reference monitor ensures that such virus script files and executable images cannot access the system resources (including files, directories, RAM, hardware devices, etc.).

This architecture and approach prevents the virus from operating - even if it is a new and ever-seen-before virus. Malicious software cannot spawn or launch other applications nor can it propagate itself via replication and retransmission. As another example, common virus scripting languages such as VisualBasicT will not execute in IQware's secure server environment because the security policy implemented by the reference monitor prevents it. Access to email list, addresses and data transfer channels is also prevented so viruses cannot spread. The reference monitor also prevents harmful application macros from executing, which closes another door to cyber attacks and hackers.

The Seven Layer Software Security Model

The critical issue of software security is one that IQware addresses within this new IT environment. IQware's unique architecture - and its use of a secure server environment - makes IQware software desktop virus immune and provides the maximum possible IT system protection.

Security is the cornerstone of IQware Software. Other IT systems use code patches and security updates, which only prevent further infections of the same viral strain. These approaches do nothing to clean up the damage from the initial attack nor do they inoculate against unknown, future virus attacks.



Other than IQware, no mechanisms for guaranteeing correct program operation.

Encryption, firewalls, anti-virus software, intrustion detection systems, etc.

- The application does the useful work - it is the "payload" of your IT system.
- The 6 underlying logical layers are there to support and enable the application.
- All seven layers must be protected to have 100% security.
- Protecting just the lower layers via firewalls and anti-virus software is not enough - as experience has proven.

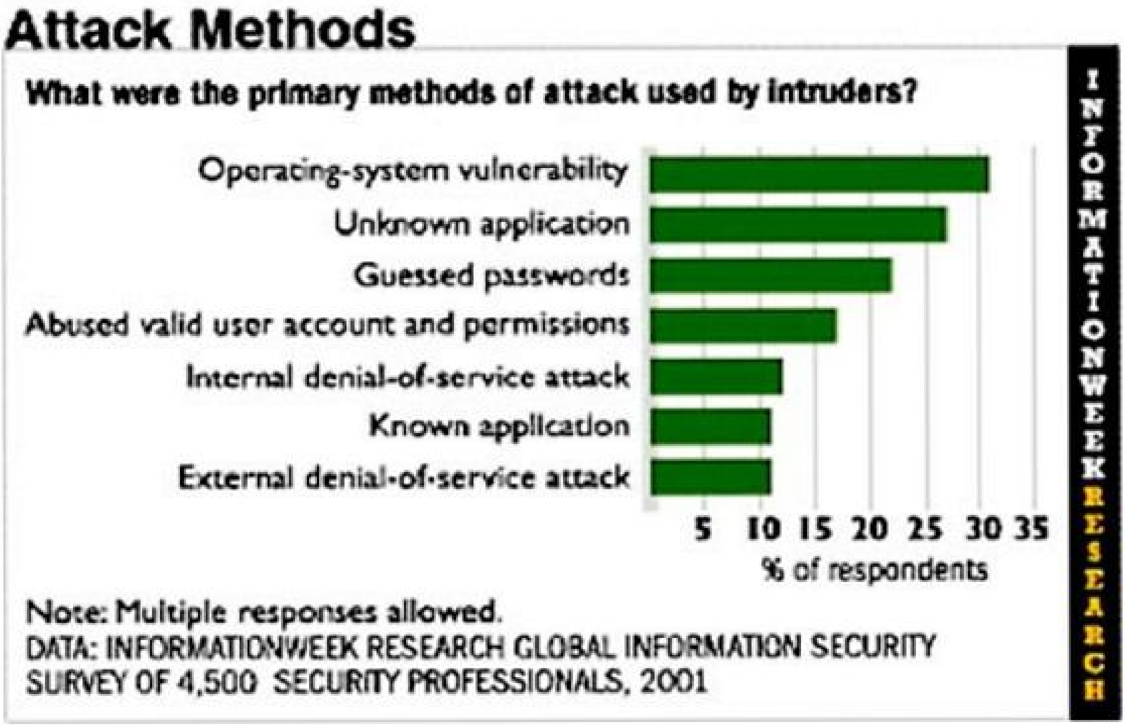
From a security perspective, IQware is quite complementary to the various firewall and anti-virus products on the marketplace. As previous diagrams have shown, software can be modeled as a 7-layer structure. These layers are logical layers that perform different functions. Each layer performs function to the layer above it in the diagram. The top layer is the application layer - it's the layer that does the useful work. The underlying six layers are only there to support the top, or application layer.

As is clear from the seven-layer software model, firewalls, anti-virus software, encryption and intrusion detection systems only protect the lower logical layers of the software models. Both research and experience have shown that this amount of cyber defense is not enough. IQware works to protect the higher layers - which is essential for 100% security. Using IQware software ensures that the software application - whatever it may be - will operate correctly even in the presence of a cyber attack.

A total security solution for a customer, which would protect all seven layers of the software model, would include firewall(s), anti-virus software and IQware's secure BI applications. *This allows IQware to position itself cooperatively and complementarily with these vendors.*

Cyber Attack Methods

IQware's security is based upon its architecture, coding and deployment. All software applications, including IQware's execute in the context of the operating system (O/S) and related run-time systems. A secure system are results from the proper architecture, coding and deployment of the operating system and application.



Note: Multiple responses allowed.
DATA: INFORMATIONWEEK RESEARCH GLOBAL INFORMATION SECURITY SURVEY OF 4,500 SECURITY PROFESSIONALS, 2001

A survey showed that the primary attack method used by hackers and cyber terrorists is to exploit weaknesses within the operating system. Another significant attack method was to exploit existing weaknesses in the application layer. IQware's approach directly addresses this significant problem by using a secure operating system on the server side in its client-server architecture. This technique lets IQware applications safely interact with common, off-the-shelf (COTS) desktop computers that are very vulnerable to cyber attacks and viruses. IQware's unique and patent-pending architecture lets customers have the best of both worlds by combining a secure environment with inexpensive and popular desktop technology.

Limitation of Anti-Virus Software and Firewalls

As the seven-layer software model clearly shows, defense techniques such as firewalls (FWs), anti-virus software (AVS), intrusion detection systems (IDS), etc. are effective only in the lower layers of the model. These techniques do nothing to address the critical issue of ensuring that your application software operates properly even in the presence of viruses and cyber attacks.

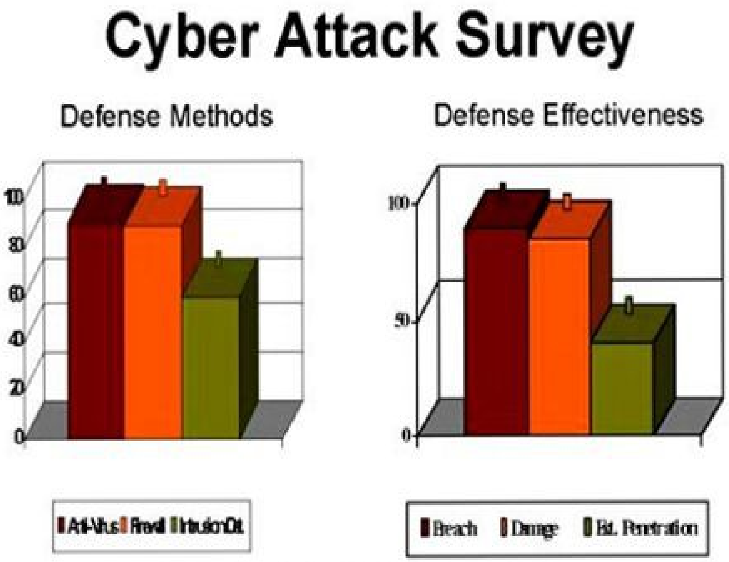
Anti-virus software can only defend against known viral strains. Firewalls help protect systems from network-transmitted viruses. Unfortunately, anti-virus software and firewalls cannot do the following critical functions:

Cannot prevent damage from known viral strains
Cannot clean up damage from cyber attacks
Cannot prevent critical data loss from cyber thieves
Cannot prevent damage from user errors
Cannot ensure that 100% correct operation application software
Cannot make an IT system tamper-proof
Cannot provide 100% security

Man-in-the-middle attacks, IP-spoofing, etc. are all techniques that can defeat firewalls and other defense schemes. Entire websites are devoted to the tools and techniques of hacking, including footprinting, scanning and enumeration. Some example websites that feature these tools and techniques include www.cultdeadcow.com, www.phrack.org, www.foundstone.com/rdlabs and www.nwpsw.com.

Cyber Defense Ineffectiveness

If firewalls, anti-virus software, intrusion detection systems and the like were truly effective, then networks and IT systems would be impregnable and there would be no concern about cyber attacks. Unfortunately, the exact opposite is true. Both experience and research have shown that firewalls, anti-virus software and intrusion detection systems cannot provide anywhere near 100% protection.



Source: Computer Security Institute, Computer Crime and Security Survey, April 7, 2002

The Computer Security Institute conducted a survey of cyber defense effectiveness in April 2002. The survey results clearly show the ineffectiveness of these defense mechanisms. Nearly all organizations that were surveyed had experienced breaches, damage and external penetration of their IT systems - *even though they had installed the best cyber defense systems.* These defense mechanisms included anti-virus software, firewalls and intrusion detection systems. The Computer Crime and Security Survey results are shown in the bar graphs.