Global Information Assurance Certification Paper

Protecting the Average Consumer – What's wrong with Firewalls.
Thomas Hauer
GSEC 1.4b
January 18, 2003

Introduction

Internet Security is critical to Internet safety. As broadband achieves greater penetration into the home market, the risk from insecure computers is increasing. Security products for the home user are either too difficult to properly set up, or too trusting to be properly secure. Given the development of multi-vector worms and other threats, an integrated defense is becoming essential. The improving feature sets of recent versions of personal firewalls confirm this. This paper will review the operation of firewalls and discuss what makes them user un-friendly. It will then offer a possible direction for improvement.

Scope

In June of 2001, estimates put the number of business and residential broadband users in the US at 9.6 million[1]. This is a large number of potential targets, however it is a small fraction of the full potential. Residential broadband of all forms is expected to exceed 38 million households by 2006[2]. Considering that there are approximately 115 million homes in the US with access to a cable system[3], it is only going to get worse. The consumer's PC is not the true target. If an attack can compromise a PC, the victim can then be used to attack other systems. The number of systems attacking determines the strength of the attack launched on the final target.

In the paper "How to Own the Internet in Your Spare Time" Stainford, Paxson and Weaver state: "If you can control a million hosts on the Internet, you can do enormous damage. First, you can launch distributed denial of service (DDOS) attacks so immensely diffuse that mitigating them is well beyond the state-of-the-art for DDOS traceback and protection technologies. Such attacks could readily bring down e-commerce sites, news outlets, command and coordination infrastructure, specific routers, or the root name servers. "[4]

One million is approximately 10% of today's residential broadband users and the percentage is falling rapidly. Using the above projections it will decrease to about 2.6% in 2006. Does anyone seriously believe that all residential users have anti-virus software and stay current with their virus definitions? Many users install anti-virus software, think they're protected and don't need to do anything else. Finding one million vulnerable systems should be comparatively easy.

Anti-virus programs are mature and have "benefited" from various well-publicized attacks such as the Melissa virus, the "I Love You" virus and the Anna Kournikova virus. Even with this level of public exposure, anti-virus vendors

have found it necessary to automate the update process and take the task away from consumers. Firewalls are still relatively unknown and do not enjoy the same maturity, market penetration or familiarity.

New techniques are constantly being developed to slip past firewalls. In August 2002, the Setiri architecture was described. This technique takes over Microsoft's Internet Explorer using features designed for convenience and turns them into avenues of attack. It will be very difficult to protect against this type of attack because it is using a legitimate application.

Firewalls - A Quick Review

A firewall is designed to police the communications between computers. It examines each message going to or from a computer and compares it to a set of rules. If it finds a rule that matches, it performs the appropriate action. For example, if your computer tries to send an email to another computer, your firewall looks at the list of rules to see if email is permitted or not. If it can't find a rule saying "no", it will let the message pass. This is a bad idea, since everything will get through unless you know what to block. A more secure approach is to add a default rule that says "block everything". In this way, nothing can get through unless you allow it. Unfortunately, not every firewall takes this approach, particularly on the consumer level. Many programs have a default rule of "ask the user". These programs will inform the user that an application has attempted to use the network and ask if it should be allowed to proceed. The firewall provides some information about what program made the request in order to help the user decide. Some firewalls provide information about what IP addresses protocols and ports are being used but others provide the information only if you ask for it. Most will offer to configure a rule for you.

Firewalls fall into two basic categories, packet filtering and application level. There are refinements on both, most notably stateful [packet] firewalls, but their underlying methods are still the same.

Packet filters examine the raw packets at the transport level; Level 4 in the OSI model, or the TCP level in TCP/IP. "Packet filters enable the administrator to permit or prohibit the transfer of data based on the following controls: the physical network interface that the packet arrives on; the source IP address the data is coming from; the destination IP address the data is going to; the type of transport layer; the transport layer source port, and the transport layer destination port."[5] The chief benefit of packet filters is speed. They operate on the information in the header and do not have to open or examine the payload. Unfortunately, safety is sacrificed for speed, since many dangers can be hidden in the payload. "Even newer state based firewalls still only look at packet information contained in the IP, TCP, or UDP headers. They tend not to look at specific data contained in those packets beyond the headers, and tend not to discern anything related to a specific protocol."[6]

Application level firewalls are slower because they have to track connections (in order to properly combine the packets), assemble the payload, and then examine it to see if the contents are in order. For example, a HTTP packet should contain commands like PUT and GET. If everything is in order, the firewall passes it to the application. Since the actual data is examined, instead of just the header, more thorough checks can be made. However, this works only for protocols it understands. Any new or unknown protocols cannot be checked.

An extension of the application level firewall is evolving. It is called an application firewall. It is targeted at a specific use of the Internet, such as HTTP. All communications on the appropriate port are compared to the relevant RFC's and anything that doesn't meet those specifications is discarded. The intention is to stop known and unknown attacks, for example, any buffer overflows whether caused by new or existing attacks. The effect is to insulate the application from known bad data, relieving some of the risk from poorly written applications. This type of product can be useful in the "defense in depth" strategy.

Firewalls and the Consumer

Firewalls are mysterious creatures outside the education and experience of the average Internet user. Ask a consumer what port 27374 is used for and you will likely get a blank stare instead of the answer "Sub-Seven". Most users don't know what an IP address, port or packet is. After all, DNS and other systems are designed to make it easier for users, insulating them from the underlying structure of the Internet. Even the "raw" Internet is too wild for many, as demonstrated by the popularity of AOL, CompuServe, Genie and similar services. Unfortunately this isolation and lack of education makes it nearly impossible for the average user to configure a firewall. If a dedicated consumer decides to read the manual before trying to configure the firewall, they probably still won't succeed. As we are well aware, IP Fundamentals covers two sections of the GSEC course. You can't squeeze hours of course work into a few pages or few kilobytes of document. Even with the best manuals, true understanding only comes with real world examples and experience.

Most users don't see past the icons on their desktop. They know how to start and use applications, but don't understand what's happening "under the hood". Few understand that a single application is made of many modules, several of which may be designed to access the network. A firewall that reaches the "ask user" default rule will usually provide the name of the module in question. However, they rarely (if ever) provide the name of the controlling application, so the user is frequently left without any idea of what the module is or does. This makes the "permit or deny" decision even more difficult. Combine that with the lack of understanding of protocols, etc. and most firewall rules end up being "permit the module to access to any port to/from any address". Put enough of these together and we seriously undermine the function of the firewall.

Now imagine what happens when the same user is being scanned for open ports. The thought process is probably something like this: “Here’s another message from that pesky firewall. I don’t know why it keeps bothering me. I’m not trying to install anything. I’ll just click on ‘configure a rule’ and make it go away.” The computer now belongs to the attacker, not the user.

Few programs tell you what networking resources they are going to use. This makes defining rules more difficult, even for professionals. Unless the documentation is exceptionally clear or other users have documented and shared their experience, a firewall administrator has to deny everything and add rules one at a time until the product works properly. This can be a time consuming and frustrating process. If the user discovers a new feature in the application, or an upgrade is applied, the process starts all over again.

Given all these aspects, it is unreasonable to expect the average user to configure a firewall and so firewalls must evolve. Early anti-virus software provides a reasonable model for improving firewalls. Anti-virus has improved over the years and users have come to expect the software to configure itself and keep the definitions current. Typically the software checks for an Internet connection. When it finds one, it periodically downloads and installs new virus definitions and then notifies the user that the process is complete. Earlier versions required the user to start the update process after they connected to the Internet by clicking on an “update” button. Before that, the users had to be aware there was an update, then locate it on a web page or BBS, download, extract and install it. Firewalls haven’t even made it this far. To continue the anti-virus comparison, it’s like asking the user to write virus signatures.

Anti-virus software takes a “permit unless denied” approach that makes sense given the unpredictable nature of data being sent through the Internet. Every data file, instant message and email is different (except spam). Therefore, you must look for the “bad guys” because you know what they look like. The downside to this approach is the window of opportunity between virus release and signature development. A firewall’s “deny unless permitted” approach is possible because there are a limited number of protocols and ports. Communication doesn’t vary as much as the data does. Email programs use the same protocols (SMTP, POP3, IMAP, MIME, etc.) from one computer to the next, but the emails the users write can be as different as night and day. While the programs can be customized and modules added, email is still sent on specific ports. In spite of these known standards, each user must define and maintain their firewall rules, effectively forcing each user to “re-invent the wheel”.

In summary, firewalls in the home are weakened by four main factors. Each user has to build a firewall from scratch; they are too difficult for the average user to configure due to incomplete or unclear information about the applications; default rules are too permissive; and it is difficult to determine how

an application communicates with other computers. In order for firewalls to gain acceptance by consumers these areas must be improved.

So Where Do We Go From Here?

The consumer needs software that provides a reasonable degree of protection with little to no work on their part. It needs to be automatically configured for installed applications and constantly update itself. It needs to plug in and work right from the beginning.

Users may feel that since a computer is used only for games (or email, or browsing) it's not worth protecting. An unguarded computer can provide a base of operations for intruders just as an unguarded house can. In fact, physical security provides a good model for data security. We do not let people enter our houses and do what they want, nor should we let applications do the same to our computers. We must start with the assumption that the house (or computer) itself is worth protecting and that we cannot trust anyone else in the neighborhood. Since the Internet is worldwide, bad neighbors are everywhere.

First, we hire a security guard from a reputable firm. As we invite service people into our house to do things, it is the guard's responsibility to make sure each person is authorized to be there, to make sure they do their job and nothing else, and to keep track of what's been going on over time. Our guard has a lot of work to do. Here is a list of what the guard must do to properly protect our home.

Identification - Get the service person's name and job function.
Authentication – Check with the service company to confirm this is the person they sent.
Permission – Find out what work was ordered and confirm it was properly authorized.
Logging – Record when the person arrives, what they do on site and when they depart.
Inspection – examine all parcels carried into and out of the building.
Verification – watch the work being done; confirm it is appropriate for the problem and up to established standards.
Asset Management – watch the service person to make sure they don't take or damage anything.
Completion Tracking – make sure the service person cleans up, doesn't leave anything behind, logs out and exits the building.
Enforcement – stop the service person from working outside of the job parameters; throw the person out if they violate rules.
Reporting – notify management of transgressions; notify the service company and law enforcement if so directed by management; log violations and periodically examine the logs for patterns.

This amount of work will require a team of guards. In a computer the team will be a "security suite", a group of applications working together to protect the computer. Now consider each application to be a service person. Each has specific skills and a specific job. You would not let a plumber work on electrical wiring, nor should an email program be controlling another PC remotely. Here is the list again, with the requirements for implementing it against software.

Identification - Get the application's name and job function. In this case the job function is an "Application Signature" telling the security suite how the program operates normally and includes a list of ports and protocols the application will use. It should also list what computer resources (directories, etc.) it will use. This list could be provided with the application, or pulled from secure servers maintained by the Security Suite and application manufacturers. The signature needs to be tamper-proof.

Authentication – Check with the manufacturer to confirm this is the application they sent. This can be as simple as checking the MD5 hashes for the program and the application signature.

Permission – Report to the user what this program will do, and confirm that they want to install it. A general explanation should be offered with more detail available upon request. For example, "sends and receives email" or "grants other people access to music files stored on your computer".

Logging – Records when the program starts, stops and all communications it sends and receives. Current firewalls do this well if they are properly configured. All data should be captured but the user should be able to select the level of detail reported. The default level should report warnings and errors. Capturing all the details also requires that the security suite monitors remaining space on the media, and has a utility for archiving/purging old data.

Inspection – This is the current anti-virus function. It should cover all communications, where possible – email, files, messaging, etc. Unfortunately encrypted traffic will be immune to inspection.

Verification – This is the traditional firewall merged with an application firewall. The security suite monitors all network traffic for two things; 1) RFC compliance – the firewall blocks and logs non-compliant traffic; 2) Application Signature compliance - the firewall will detect and log any attempts to work outside the application's stated protocol and port usage. This function is closely integrated with Enforcement.

Asset Management – monitor files accessed by the application. Make sure the application uses only those directories listed in its Application Signature.

Completion Tracking – confirm that all modules loaded by the program are unloaded when it closes. Confirm all installed files are deleted when the application is removed.

Enforcement – stop applications that are flagged by the Verification function. This module could be designed to spoof the offending application in an effort to learn more about an attack and try to track it.

Reporting – log the reasons for all terminations and report them to user. It should have the ability to report the violation to various security sites such as anti-virus vendors, security organizations, bug-tracking organizations, and the manufacturers of the firewall and of the application. This is an area of conflicting needs. Internet security calls for reporting violations to organizations that can use the information but privacy concerns argue against it. This should be the area where the user has the greatest input; each user determining just how much information they wish to share. The security suite should periodically review the logs, looking for patterns of "misbehavior".

Most of these behaviors are currently available in one form or another. For example, some firewalls and products like Tripwire do an excellent job of monitoring files for changes (Asset Management); most firewalls prompt the user for unknown software (Permission) and most have the ability to log transactions (Logging). Some expansion and refinement of these abilities will fulfill most of the roles discussed. However, current security products implicitly trust the applications being installed, and have no idea of what the applications are supposed to do (Identification, Authentication and Verification). The idea of an "Application Signature" addresses these areas.

The Case for Application Signatures (AppSigs)

By now most people have seen the screen stating that an application does (or does not) have Microsoft approval and an appropriate digital signature. This is where Application Signatures start, but they must go further. The signature needs to do several things: 1) Uniquely identify the software including version information; 2) uniquely identify the manufacturer; 3) provide proof that the software was not tampered with; 4) provide a list of all modules included; 5) provide a list of the protocols, ports and other resources each module is allowed to use; 6) provide a layout of acceptable packets for any proprietary formats (RFC equivalent); 7) be resistant to spoofing/hacking.

The problems that create the need for a solution like this will not go away. There will always be people who will attack computers and communications, whatever their motive. This imposes an additional requirement. AppSigs should not be vendor or platform specific, permitting the concept to adapt to changing technology. Intel envisions the "e-home" of the future with a diverse array of Internet enabled devices[7]. Given this view, there is no way to know what

hardware or what operating system it is running on. When an application is installed it should report the signatures to the suite without worrying about the underlying platform.

This is a huge project that will require a group effort, as much to defray the costs as to avoid it becoming proprietary. It could be developed as a standard or as a "branded" program. Any application that passes a certification can be labeled "SecuritySuite Ready".

How do application signatures address the four weaknesses identified earlier? The signature makes the firewall "modular". The user no longer has to build the Security Suite from scratch since each application automatically extends the suite as it is installed. The signature contains all the protocol and port information, eliminating the need for the user to configure anything beyond the privacy settings. The remaining issues are eliminated by the rules in the signature that list exactly what the application will use. A default "block everything" rule will block anything that isn't registered.

Application Signatures can be good for firewall manufacturers as well. In theory a signature shouldn't change unless a new version is released but practice is rarely so clean. Following the example of anti-virus software, firewall companies could offer subscription services. The service could include automatic download of attack signatures, updating AppSigs, confirming that existing AppSigs haven't been tampered with and even a review of the firewall log. The manufacturer could compare log activity against a database of AppSigs and flag any unexpected activity. This would help speed discovery and developing signatures for new attacks.

Application Signatures are not a complete solution. They are an additional layer of the "defense in depth" strategy. Security Suites should continue to use attack signatures that block known ports, such as Sub-seven's 27374. The AppSig certification process could prevent any application from using known attack vectors for legitimate purposes.

Hybrid threats could still pose a problem. A virus could be developed that hijacks a legitimate application and use it to launch an attack, similar to the Setiri architecture mentioned earlier. Checking RFC compliance and the AppSig equivalent may make this harder to put into practice. However, the ability to hijack an application is due to the application's security flaws. Guarding against poorly written applications is very difficult and may be impossible for a Security Suite. Once a vulnerability is confirmed, the suite manufacturer can push an attack signature through the subscription service, providing a temporary patch until the vulnerable application can be corrected.

Summary

Firewalls in the home are still evolving. In the past few months new versions of several firewalls have been released. New features include: running applications in a "sandbox" so that it can be stopped if it misbehaves; learning modes where the firewall monitors an application recording the protocols and ports it uses to help build rules; virus scanning; privacy monitoring (cookies); and log analysis. The addition of these features is a good indication of what the firewall will become. The features of firewalls, Intrusion Detection Systems, anti-virus software and privacy software will merge into a "Security Suite". Early versions of suites are already out. To be successful, these products must automate the process of security. Home users need a "Check Engine" light for their security system. It should be invisible, until it detects something is wrong.

Security suites with these capabilities will benefit businesses as well as consumers. Obviously, decreasing the number of vulnerable systems will decrease the numbers of attacks. However, the technology of automatic updates would be welcome in the corporate world. How many hours could be saved if a central policy server could push approved AppSigs to the desktop. VPN connections could be denied to remote systems that don't have up-to-date or valid AppSigs. The VPN connection could push a set of temporary limits to the user's security suite, extending the corporate policy to the home computer for the duration of the session. An integrated defense will benefit all participants.

Bibliography

Cable-Modems.Org. "Cabel Modem Market Share and Shipments" 2002 (Cable-Modems.Org)
URL: www.cable-modems.org/articles/market

Internet Security Systems. "Response Strategies for Hybrid Threats" 2001 (Internet Security Systems, Atlanta GA)
URL: http://documents.iss.net/whitepapers/HybridThreat.pdf

Lonestar Broadband. "Background: National Story" May 31, 2002 (LonestarBroadband.org)
URL: http://www.lonestarbroadband.org/background/nationalstory.htm
[The link may not work directly. Go to the home page, then "background", then "National Story"]

Merrick, Craig et al. "Extending the PC in the Home" August 23, 2002 (Intel Corporation)
URL: http://www.intel.com/technology/itj/q22001/articles/art_3.htm

Permeh, Ryan. "The Use of Application Specific Security Measures in a Modern Computing Environment" (Eeye Digital Security)
URL: http://www.eeye.com/html/Research/Papers/DS20010322.html

Romanofski, Ernest. “A comparison of Packet Filtering Vs. Application Level Firewall Technology” March 18, 2001 (SANS Institute)
URL: http://rr.sans.org/firewall/app_level.php

Sapiro, Benjamin. “Application Level Content Scrubbers” August 22, 2001 (SANS Institute)
URL: http://rr.sans.org/firewall/scrubbers.php

Skoudis, Ed. “SANS Threat Update Briefing” September 19,2002 (SANS Institute)

Staniford, Stuart et al. “How to Own the Internet in Your Spare Time” May 14, 2002 (This paper appears in the Proceedings of the 11th USENIX Security Symposium (Security '02))
URL: http://www.icir.org/vern/papers/cdc-usenix-sec02/index.html

Zwicky, Elizabeth D. “Building Internet Firewalls, Second Edition” June 2000 (O’Reilly & Associates Inc.)
URL: http://www.oreilly.com/catalog/fire2/

Endnotes

[1] Lonestar Broadband. “Background: National Story” May 31, 2002
URL: www.lonestarbroadband.org/backgroud/nationalstory.htm

[2] Cable-Modems.Org. “Cabel Modem Market Share and Shipments” 2002
URL: www.cable-modems.org/articles/market

[3] Cable-Modems.Org. “Cabel Modem Market Share and Shipments” 2002
URL: www.cable-modems.org/articles/market

[4] Staniford, Stuart; Paxson, Vern; Weaver, Nicholas. “How to Own the Internet in Your Spare Time”
URL: http://www.icir.org/vern/papers/cdc-usenix-sec02/index.html

[5] Romanofski, Ernest. “A comparison of Packet Filtering Vs. Application Level Firewall Technology” March 18, 2001
URL: http://rr.sans.org/firewall/app_level.php
[6] Permeh, Ryan. “The Use of Application Specific Security Measures in a Modern Computing Environment”
URL: http://www.eeye.com/html/Research/Papers/DS20010322.html
[7] Merrick, Craig et al. “Extending the PC in the Home” August 23, 2002
URL: http://www.intel.com/technology/itj/q22001/articles/art_3.htm

Last Updated: April 22nd, 2018

Upcoming Training

SANS Seattle Spring 2018	Seattle, WA	Apr 23, 2018 - Apr 28, 2018	Live Event
Mentor Session - AW SEC401	Detroit, MI	May 01, 2018 - May 17, 2018	Mentor
SANS Security West 2018	San Diego, CA	May 11, 2018 - May 18, 2018	Live Event
SANS Northern VA Reston Spring 2018	Reston, VA	May 20, 2018 - May 25, 2018	Live Event
SANS Atlanta 2018	Atlanta, GA	May 29, 2018 - Jun 03, 2018	Live Event
Community SANS New York SEC401	New York, NY	Jun 04, 2018 - Jun 09, 2018	Community SANS
SANS London June 2018	London, United Kingdom	Jun 04, 2018 - Jun 12, 2018	Live Event
SANS Rocky Mountain 2018	Denver, CO	Jun 04, 2018 - Jun 09, 2018	Live Event
Community SANS Bethesda SEC401	Bethesda, MD	Jun 04, 2018 - Jun 09, 2018	Community SANS
SANS Oslo June 2018	Oslo, Norway	Jun 18, 2018 - Jun 23, 2018	Live Event
Community SANS Portland SEC401	Portland, OR	Jun 18, 2018 - Jun 23, 2018	Community SANS
Community SANS Madison SEC401	Madison, WI	Jun 18, 2018 - Jun 23, 2018	Community SANS
SANS Crystal City 2018	Arlington, VA	Jun 18, 2018 - Jun 23, 2018	Live Event
SANS Cyber Defence Japan 2018	Tokyo, Japan	Jun 18, 2018 - Jun 30, 2018	Live Event
Minneapolis 2018 - SEC401: Security Essentials Bootcamp Style	Minneapolis, MN	Jun 25, 2018 - Jun 30, 2018	vLive
Community SANS Nashville SEC401	Nashville, TN	Jun 25, 2018 - Jun 30, 2018	Community SANS
SANS Minneapolis 2018	Minneapolis, MN	Jun 25, 2018 - Jun 30, 2018	Live Event
SANS Cyber Defence Canberra 2018	Canberra, Australia	Jun 25, 2018 - Jul 07, 2018	Live Event
SANS Vancouver 2018	Vancouver, BC	Jun 25, 2018 - Jun 30, 2018	Live Event
SANS London July 2018	London, United Kingdom	Jul 02, 2018 - Jul 07, 2018	Live Event
SANS Charlotte 2018	Charlotte, NC	Jul 09, 2018 - Jul 14, 2018	Live Event
SANS Cyber Defence Singapore 2018	Singapore, Singapore	Jul 09, 2018 - Jul 14, 2018	Live Event
SANSFIRE 2018	Washington, DC	Jul 14, 2018 - Jul 21, 2018	Live Event
SANS Malaysia 2018	Kuala Lumpur, Malaysia	Jul 16, 2018 - Jul 21, 2018	Live Event
SANSFIRE 2018 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 16, 2018 - Jul 21, 2018	vLive
Mentor Session - SEC401	Jacksonville, FL	Jul 17, 2018 - Aug 28, 2018	Mentor
Community SANS Bethesda SEC401	Bethesda, MD	Jul 23, 2018 - Jul 28, 2018	Community SANS
SANS Pittsburgh 2018	Pittsburgh, PA	Jul 30, 2018 - Aug 04, 2018	Live Event
SANS Hyderabad 2018	Hyderabad, India	Aug 06, 2018 - Aug 11, 2018	Live Event
SANS San Antonio 2018	San Antonio, TX	Aug 06, 2018 - Aug 11, 2018	Live Event
SANS Boston Summer 2018	Boston, MA	Aug 06, 2018 - Aug 11, 2018	Live Event