Network Perimeter Defense

Josef Pojsl, Martin Machácek
jp@tns.cz, mm@tns.cz

Trusted Network Solutions, Inc.
Czech Republic

Abstract

This paper outlines methods traditionally used for securing network perimeters. These methods are confronted with changes in the nature and amount of network usage. Even the original scope of ”local” network has expanded dramatically. Ever growing complexity of information systems is the key enemy of security.

New security technologies enable further measures to be deployed. Solutions implementing among others cryptography, content scanning and intrusion detection techniques are getting popular.

Security and instant adoption of new-born communication technologies are mutually exclusive. In this paper, we focus on security needs in high-risk environments and in this perspective emphasize some design requirements for securing data networks.

Keywords: perimeter defense, bastion host, proxy gateways, packet filtering gateways, NAT, DoS, VPNs, IPv6, IPsec, PKI, content filtering, IDS, vulnerability scanning.

1. Traditional approaches to perimeter defense

First firewalls were either single-homed bastion hosts on a screened subnet or mere screening routers. In the old days of Internet, it was perfectly acceptable for users to have to connect to a dedicated host that provided them with a special interface. Users were able to use mainly command-line tools like telnet or electronic mail.

In a very short time, firewall became a very popular term. Vendors hit the low-end market with many simple and cheap ”firewalls” and changed the meaning of the word completely. For the rest of this paper, we will use it in the original notion of a robust, expert device destined for network perimeter security.

1.1 Proxies versus packet filters

Two different approaches to firewall design have been fighting since their first appearance. The first approach started with improving the bastion host and making it a dual-homed proxy gateway. The idea is to analyze traffic on the application layer in order to decide competently whether the communication should be allowed or denied. Every single bit is processed by the network stack on the gateway, application data are reconstructed and examined and if authorized, sent out to its ultimate destination.

Today’s proxy gateways add a convenience feature called transparency. Users no longer need to connect to the proxy server explicitly; they even need not know about the proxy gateway at all. The most important feature of proxy gateways is that IP stacks of internal hosts are never exposed to IP packets generated by a device beyond control of the network administrator.

In the second approach, the bastion host was completely left out of the picture and the screening router was chosen to perform the job of securing the network. These types of firewalls are able to adapt to new protocols much easier. They are faster because they are usually kernel based and no user level processes are involved. There is one reason they have not eradicated proxy servers; they do not check application data (which is why they are faster), so their ability to provide fine-grained control is limited.

Popular packet filtering firewalls come up with an engine able to watch the communication more thoroughly; this is known as stateful filtering. Some vendors claim their filtering based products are able to collect data on the application layer and are thus as secure as proxy gateways. However, this task has a fixed complexity and it is not possible to perform it while avoiding its necessary overhead.

Even in the case of generic proxies that give up analyzing the application data, there are still some advantages over packet filtering:

- Proxies inherently secure IP stacks of internal hosts from attacks on the network and transport layers.
- With rare exceptions, proxies are user-level processes, which is a gain in terms of stability of the whole system.
- Proxies naturally translate addresses and thus avoid the hassle of performing network address translation (NAT).
- Assuming both proxies and packet filters collect application data, proxies are inherently more stable as they have a modular design; failure in one component is less likely to cause failures in other components of the system.

1.2 Multilevel security

In some environments, there are many different layers of security required. Implementing several zones of defense is an old and common design. Combinations of packet filters and proxy gateways on borders to public networks are very popular.

At the same time, network segmentation divides the organization’s electronic assets into groups of risk categories. These groups have their own internal borders that must be secured as well. The most sensitive assets are sometimes not connected to public networks at all. Quite often, internal network borders have different security requirements than borders to public networks.

2. Changing conditions

2.1 Perimeter expansion

The traditional notion of network perimeter is no more valid. It can be no longer viewed as a limited set of computers located in the same building or otherwise physically isolated. Some forecasts say that among hosts connected to the Internet, mobile devices will outnumber classic computers in a few years. Visions of a huge number of wireless devices simultaneously connected to the Internet are not exaggerating; experts have been working on new wireless and IP standards heavily.

A perimeter might include distant branches, roaming users, telecommuters, business partners and possibly others. All these groups of people have different needs, and we possess different levels of trust in them. The most important task is to deliver sensitive data to authorized users connected to the Internet.

2.2 Bandwidth

Proxy gateways have always been considered slower than packet filters. But as far as the Internet connection is measured in kbps, they are able to handle the traffic. Packet filters do not reconstruct data up to the application layer and therefore are typically faster. Of course, speed should not be the only (even not the major) argument when security is involved. A balance between those parameters must be found.

In the course of the last few years, new orders of bandwidth became real. The dawn of 1Gbps Ethernet brought even packet filters to their limits: currently, there is probably no packet filter that can handle 1 Gbps of traffic at wirespeed. As a way out, firewall bundles are going to be used in many networks.

The increase of bandwidth has some important consequences to security. Leaks of huge amounts of information may be performed in mere seconds. Therefore, access control must be more sound and intrusion detection faster. Required response time shortened so that systems must be able to prevent from unauthorized activity. In this perspective, actions like thorough logging, traffic analysis and fine-grained access control involved in any detail of communication should be performed.

2.3 Complexity

The number of network interactions grows exponentially. Internet is probably the most complex system humans have ever built.

Software complexity grows exponentially as well. According to [5], Windows 3.1 had 3 million lines of code, Windows 95 has 15 million and Windows 2000 has something between 35 and 60 million. The number of ways in which software components interact in a single computer must inevitably grow in the same way. It has become practically impossible to track all possible combinations of interactions in a single machine. This makes software testing comparable to fortune telling.

Every day, several new vulnerabilities are published and several patches are issued. It has become extremely expensive to track all of them and take appropriate measures. Many hosts stay unpatched and wait for their exploits to come. The Ramen worm that struck in January 2001 targeted several vulnerabilities for which patches had existed for several months. It is clear that many, if not most security incidents would not be possible if victims applied all available patches.

Also, the average quality of commercial software has decreased a lot. Vendors are under pressure and shorten life cycles of their software. What used to be a beta quality product is now released and accepted as standard commercial software. Its producers cannot afford to perform a thorough security design that would put the software's release date off. The number of features grows so that it is not possible to test all their possible combinations. Unintended configurations are likely to contain vulnerabilities that are very hard to discover.

2.4 Importance of information systems and communications

Many organizations have become vitally dependent on their computer and electronic communication systems. This makes potential attacks more harmful. Under certain circumstances, even denial-of-service (DoS) attacks may cause a lot of damage. These attacks are the most easy to perform and the most difficult to fight against. Broad availability of hacking tools means that even an unskilled person can hit and hurt.

In many cases, paperwork has been completely replaced with its electronic equivalents. If integrity of electronic data is violated, recovery and even detection is much harder. Setting a proper infrastructure for securing electronic data integrity is a challenging task. Automated integrity attacks are able to affect much bigger amounts of data than paper forgery. All this gives new opportunities to fraudulence, piracy and espionage.

3. Additional security techniques

Apart from classical measures like network segmentation, packet filters and proxy gateways, there are many other network security techniques. Among them are cryptography, content filtering, intrusion detection etc. A brief introduction to some of them together with a description of its effects on perimeter defense follows.

3.1 Cryptography

Cryptography is now widely recognized as a method to ensure data confidentiality, authentication and integrity. But it can never serve as the only method for data security. Even the most promising methods of public key cryptography have (often neglected) flaws that may lead to false sense of security. Care must be taken when deploying large-scale encryption.

For example, the world's number one certification authority VeriSign recently issued two certificates claiming to belong to Microsoft. Real owners are unknown cheats who now can distribute software apparently produced and digitally signed by Microsoft (see [8]). The moral here is that correct implementation of key management is a hard and complex task in every project.

An emerging standard for virtual private networks (VPNs) called IPsec and especially its key exchange mechanism named IKE are very complex systems. This makes them very susceptible to bugs (see [7]). In IPv4 networks, just one of many modes of IPsec has become popular: ESP in tunnel mode. This encrypted tunneling protocol is suitable for deployment as a tool to create VPNs between several distant networks over the Internet. Its other purpose, allowing traveling users to connect to the internal network, is even more popular but not very well designed. Up to now, there is no standard specifying how to assign internal network addresses to remote clients. Vendors either let external address space into the internal network or provide users with proprietary solutions. As nearly every border gateway needs to implement a means of roaming clients' VPN now, IPsec is a part of most firewalls.

A remote host equipped with a VPN connection to the internal network is a great threat. All machines on the internal network are somehow secured by the corporate firewall and possibly other systems. But this is not true for a remote client who can talk to hosts on the internal network as well as to the rest of the world. What if the client is infected with a trojan? Personal firewalls, hit of the last year, attempt to address this issue. They are lightweight, very limited packet filters, often bundled with VPN clients. They do not reach high security standards set by real firewalls.

Encryption may also break several security schemes. For example, content filtering cannot be performed on digitally signed data unless the scanner mounts a successful man-in-the-middle attack.

3.2 Content scanning

Content scanning and filtering is the most popular technique aside from firewalls. The main reason is the threat of viruses and worms. Content scanners are able to identify and also filter undesired content like applets, executable files etc. Quite often, the content scanning engine is equipped with a categorized list of destinations, which lets administrators filter out unwanted material.

All of these three functions are dependent on an up-to-date database of content samples and as such are quite imperfect. Additionally, it is unfeasible to precisely predict interpretation of data on internal hosts. Nevertheless, content coming into the protected network is potentially dangerous so that a means of content scanning and/or filtering is inevitable in most networks today.

3.3 Intrusion detection

Intrusion detection systems (IDS) try to identify security incidents. Their task is similar to house alarms: they watch valuable resources and discover hostile activities. In fact, it is a very hard job even for a skilled and experienced person. When converted into algorithms, they are even more limited. In many cases, the information provided by these automated IDS may not be relevant.

There are two basic types of IDS. The network-based IDS capture all network traffic and try hard to figure out what is going on. When a suspicious event occurs, administrators get informed. It is very difficult to interpret network packets and their exact effect on the target hosts.

Host-based IDS partially remove this weakness. More appropriate information, such as the state of the operating system and its memory, are available for them. They are far more useful then network-based systems, although their deployment and management are more difficult.

Given all the deficiencies of machine-made intrusion detection, the concept of active reaction based solely on automated systems is very dangerous. Simple DoS attacks may potentially disconnect the network completely.

3.4 Vulnerability scanning

Good vulnerability scanners provide useful information while verifying correctness and completeness of protection. They can point out misconfigurations resulting in vulnerabilities to known attacks but they are of no help in protecting against yet unknown attacks and provide only as accurate information as exhaustive and up-to-date their vulnerability database is. In most cases, penetration tests performed by humans are more accurate. However, automated vulnerability scanning should not be avoided in a complete security design.

3.5 IPv6

IPv6 itself is not a security measure but when deployed massively, it will influence security a lot (both positively a negatively). On the plus side, it assumes a large-scale deployment of IPsec. Disregarding the above-mentioned difficulties of key management, it can be a great step forward in securing network infrastructure because eavesdropping will be a much harder task. The enlarged address space will mean that there will be no need for NAT any more.

On the minus side, IPsec will break content scanning and filtering as well as network-based intrusion detection. IPv6 is also a very complex protocol (consider nearly unlimited header chaining) and its implementations are likely to contain bugs. Some other features, like improved source routing, are close to disaster from security perspective.

4. Network perimeter defense in a high-risk environment

Security is always a tradeoff and everyone has to find their own balance. In a high-risk environment, the need to communicate should not outweigh the need for security. Any promising technology is not going to be adopted unless it could be controlled and used securely.

In the rest of this paper, we will omit organizations with a low-risk profile. These entities are likely to accept less secure solutions because big investments required by advanced security measures would not be profitable for them.

Standard commercial solutions are rarely ready to adapt to conditions of today's communication needs easily. Most vendors attempt to claim that you can secure your network with a single click in their nice-looking GUI. However, security officers who recognize the complexity of their task must develop a better set of measures. Based on the above-mentioned conditions, we conclude some design principles that a network perimeter defense shall obey.

4.1 Services instead of products

Security is a moving target. As such, many of its functions may not be carried out by fixed elements. Consider a virus scanner: what would its value be without permanent application of new samples? This is not as clear with other network security components but the principle is the same. Not only new viruses, but also new vulnerabilities, exploits and patches emerge every day.

Generalized security measures are often not enough. Many organizations have their own customized information systems. If the scope of these systems gets expanded, special care should be taken and security systems should be customized as well. Security parameters differ from one organization to another in such a great extent that it is unlikely that a general-purpose product (or even a set o products) will cover, support and be well tested for all the parameters' combinations.

4.2 Open architecture

By open, we do not mean open-source software. Open in this sense means that the vendor has nothing to hide. This is a key part of many discussions: does hiding bring anything to security?

Briefly summarizing, advocates of closed source software argue that if adversaries do not have access to internals of your system, it is much harder for them to attack you. This approach relies on false assumption that the internals of the software will remain closed forever. It has been proved many times that secrets in IT world are never kept for too long. In addition, it is much easier for administrators to take even temporary measures if they know the system internals well or even if they possess its source code. Information hiding may sometimes bring a small value, but only if the risk is low. If something of high value is involved, more expensive attacks are likely and nothing is hidden well enough from them. On the other hand, if an open system survives thousands or more eyeballs looking into its internals, the chance is good that this system is quite safe. In 1998, even the American NSA accepted this approach and revealed their former secret encryption algorithms, SKIPJACK and KEA (see [9]).

One of the best ways to make a system well documented and verifiable is to deliver it with source code included. An open system must be easy to understand and its behavior must be predictable. In high-risk environments, strong emphasis on verification of every system's behavior yields the need for source code. This imperative is yet stricter when security systems are involved.

Compliance with open standards is essential where possible. In many cases, they are far from perfect. Some IETF standards suffer from high complexity because they are based on large consensus. Standards serve to ensure interoperability but they are useful in verification of a system's behavior as well, which is especially important for security systems. In the real world, however, we must also adapt to other systems that are often not compliant.

4.3 Minimalism

Being minimalist is the only efficient way to fight complexity. Anything not essential to a key function of the security design should be removed. In this way, the number of possible states and interactions inside the system gets decreased. The less functions a system or a software product has, the more secure it is. It is not possible to emphasize this principle enough.

4.4 Modularity

Any minimalist approach to (especially software) systems leads to modularity. A toolkit-like system is more suitable here. By definition of simple interfaces between modules, complexity is minimized and one can concentrate on security consequences of individual modules' behavior. Modules may be removed or replaced which contributes to a minimal, customized design.

Several key functions of perimeter defense are best achieved with modules. For example, access control should be as independent on other components as possible. Also, firewall bundles are more easily built with modular systems because distribution of its modules among the hosts involved in the bundle is much easier than distribution of the whole system.

4.5 Event logging and log analysis

Any action taken by the system must be logged in a way that enables to understand it. Definition of ”normal” behavior and detection of exceptions from this normal is only possible when full event logs are available. All relevant components of the system must log runtime information in a consistent, easy to process way. Without adequate level of event logging and its detailed analysis, a system cannot be considered secure.

The analysis should be very comprehensive and consider correlations between logs from different components of the system. It could be based on a set of rules as an expert system or rely only on statistics. Best results are achieved with a combination of both, together with a manual inspection.

4.6 Specialization

As the whole subject gets more and more complex, it is often far beyond the scope of a system or network administrator to keep track of security news, interpret them correctly and apply them with respect to his or her specific situation. As stated above, the number of security advisories is so high that many administrators fail to follow them.

It seems to be more effective to rely on some team of experts dedicated to security monitoring who are able to do that job. Specialized teams may be a part of the organization and operate in some form of internal outsourcing, or they may be contracted from another company. These teams should never fulfill the daily tasks of network administration that shall still be the administrators' responsibility. When an initial security design is finished, the experts should be continuously redefining its components and procedures to adapt to new facts and conditions.

5. Conclusions

Many functions required to maintain desired level of security are continuous processes. As such, they can be delivered better as services rather than as (software) products. Clearly, quality software products are required as building blocks that can be used to create infrastructure for implementing security measures. However, without expert configuration and continuous maintenance they are rarely able to provide desired level of security in high-risk environments.

Open architecture of security components and customizable tools are required under risky circumstances. Network security elements, made out of these tools, should be as transparent and easy to understand as possible. Source code delivery appears to be the best way to achieve these goals.

Minimalist approach and modular design decrease complexity that has shown to be the biggest enemy of security. Customizing, fine-grained access control and clustering are also best achieved with a set of modules instead of a monolithic product.

Different environments have different security needs and capabilities to address them. The volume of security news, advisories, bugs, exploits and incidents is very high. In many cases, the best solution is to rely on an expert team specialized in information security that would suggest how to apply new findings to a specific security model.

6. References

[1] Schneier, B.: The Security Patch Treadmill, Crypto-Gram electronic newsletter, March 15, 2001 issue, http://www.counterpane.com/crypto-gram-0103.html.

[2] Mudge: Mudge Ado About Security, CIO Magazine, March 1, 2001 issue, http://www.cio.com/archive/030101/mudge.html.

[3] Loshin, P.: Open Source Under The Hood, Information Security Magazine, March 2001 issue, http://www.infosecuritymag.com/articles/march01/features1_open_source_sec.shtml.

[4] Schneier, B.: Secrets and Lies: Digital Security in a Networked World, John Wiley & Sons, Inc., ISBN 0-471-25311-1, 2000.

[5] Schneier, B.: Software Complexity and Security, Crypto-Gram electronic newsletter, March 15, 2000 issue, http://www.counterpane.com/crypto-gram-0103.html.

[6] Ranum, M. J.: Thinking About Firewalls V2.0: Beyond Perimeter Security, originally presented at SANSII in Washington, DC, 1993, http://pubweb.nfr.net/~mjr/pubs/think/index.htm, 1997.

[7] Ferguson, N. Schneier, B.: A Cryptographic Evaluation of IPsec, http://www.counterpane.com/ipsec.html, 2000.

[8] Microsoft Corporation: Microsoft Security Bulletin MS01-017, March 28, 2001 issue, http://www.microsoft.com/technet/security/bulletin/MS01-017.asp.

[9] NIST: SKIPJACK and KEA algorithms, http://csrc.nist.gov/encryption/skipjack/skipjack-kea.htm, 1998.