GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials (Security 401)"
at http://www.giac.org/registration/gsec

Trusted Operating Systems
Charles Jacobs
May 14, 2001

Introduction

You've probably heard the hype around the OpenHack III challenge sponsored by eWEEK magazine. In case your not familiar with the Openhack challenge, for the last couple of year's, eWeek has set up a challenge for hackers to break into test systems. In the past, they have used the most common security model, perimeter defense. This year they used a different security structure called a Trusted Operating System. During OpenHack III, about 5.25 million attacks from some 200,000 hackers hit the servers, yet not one single breach scenario was achieved. The purpose of this paper is to define what a trusted operating system is, look at the pros and cons of trusted operating systems as compared to the traditional security model, and look at the future of security models.

Definition

What is a trusted or secure operating system? Trusted operating systems are enhanced versions of everyday operating systems such as Windows NT or Unix that are made more secure. The trusted operating system concept is nothing new. It was actually developed in the early 1980's and received evaluation from the National Security Agency in 1984. A trusted operating system generally involves four components. They are information compartmentalization, role compartmentalization, least privilege and kernel level enforcement. Let's look at these four principles a little more in depth.

Information compartmentalization restricts what information an application has access to. If one application on the system is broken into, this prevents access to another unrelated application. For instance, if the web server component is compromised, the attacker won't be able to get at the database component.

Role compartmentalization restricts the control a user has. There is no such thing as root access on a trusted operating system. Even adding users and other routine administrative tasks requires the use of more than one account. This would prevent an attacker from getting full control of the system.

Least privilege restricts what processes are able to perform. Processes should only have enough rights to perform their duty. For example, a web server process should not be able to modify an e-mail file or any other system files, just the web files that it uses.

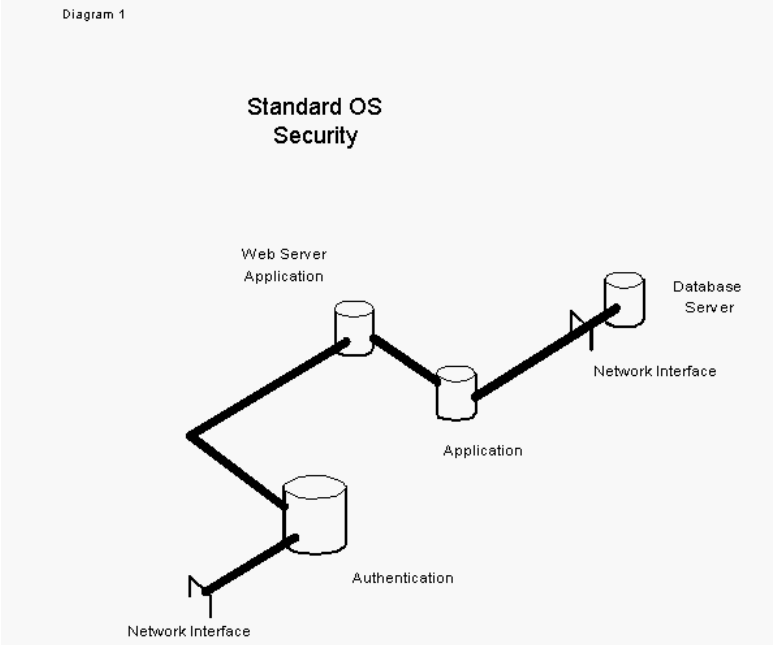
Kernel level enforcement ensures that security decisions are made at a low level where users or applications cannot interfere with them. This also reduces

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

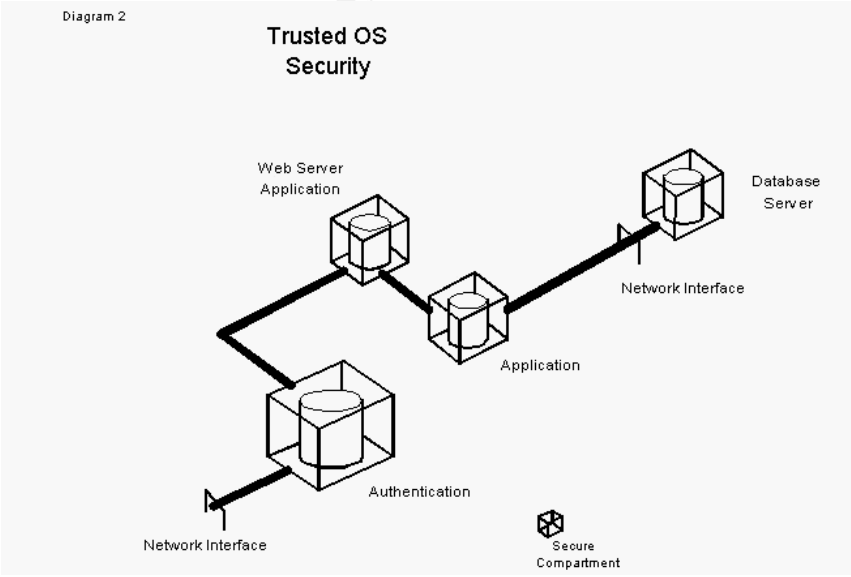
system overhead because the security decisions are close to the resources being protected.

Basically a trusted operating system is about separating elements from each other and making access between areas more difficult. It's like having a firewall and access list to each application and process.

A standard operating system's architecture is fairly open once the user is past the initial authentication(Diagram 1).



The trusted operating system because of it's information compartmentalization locks down the individual components(Diagram 2).



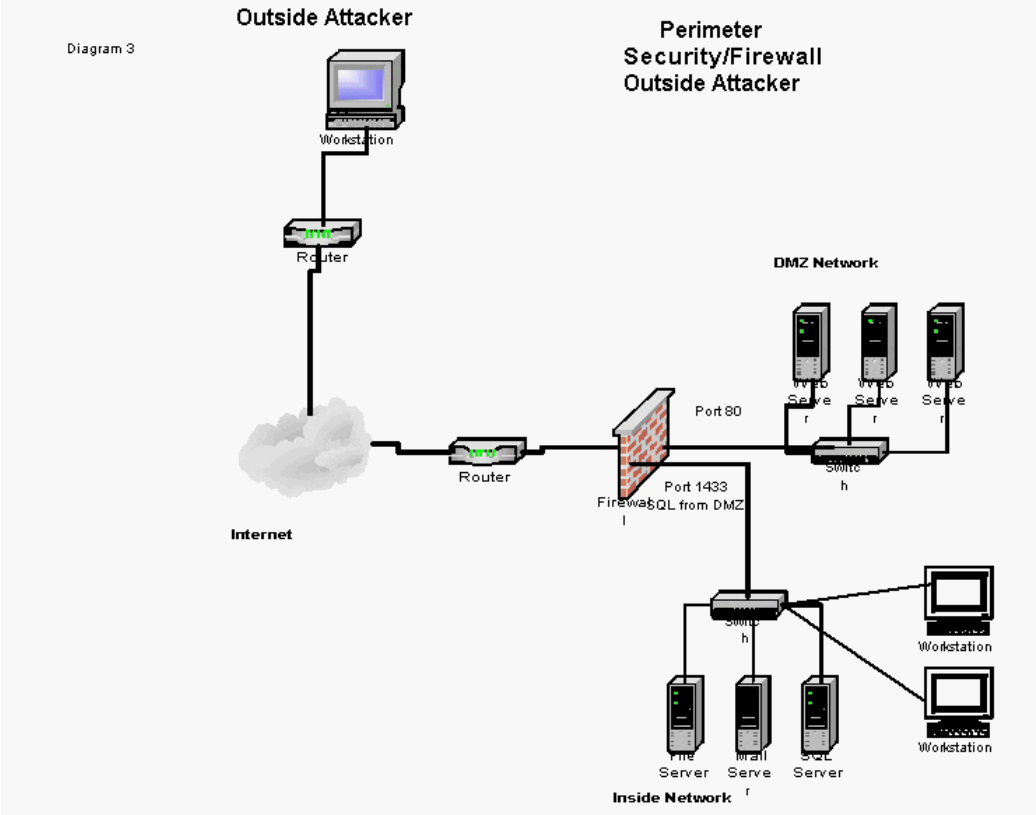
© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Advantages and disadvantages

Now that we have defined what a trusted operating system is, let's compare the advantages and disadvantages of the trusted OS and compare them to a traditional security model, the firewall with a dmz, inside, and outside network. There are four areas we will look at: security, administration and configuration, compatibility, and cost.

First is security and let's look at some scenarios of possible attacks. In these scenarios the trusted operating could be used in conjunction with a standard firewall and perimeter security. In the Openhack III challenge, no firewall or perimeter security was used.

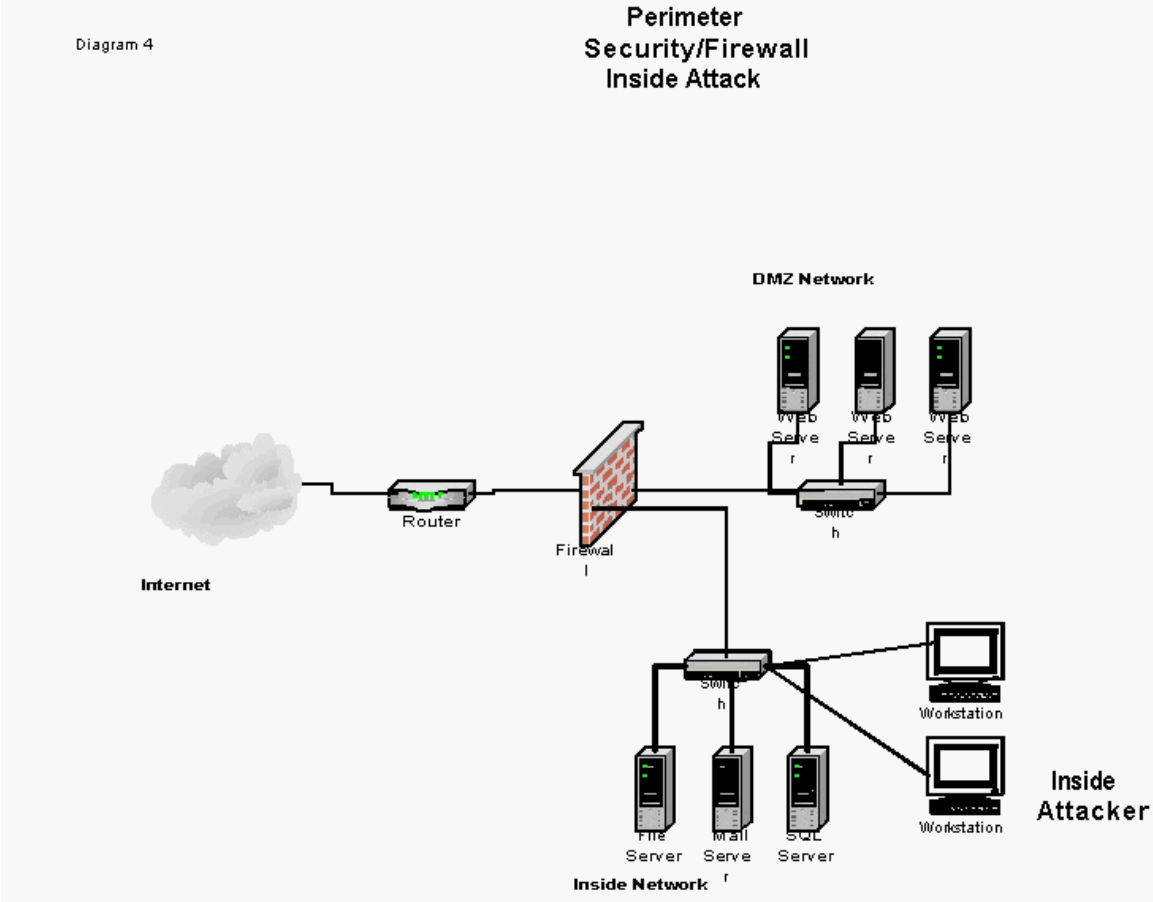
Suppose an attack is coming from the outside network(the internet). In the perimeter security model, it is common to put a web server behind the firewall on the DMZ network and it is tied to a database server on the inside network(Diagram 3). The only port open from the outside network to the web server is port 80(http). There is also a path open from the web server to the database server on some port depending on the database type. If the attacker finds a vulnerability on the web server and gains root access, the attacker then has full access to that system and possibly an easy path to the database server and the inside network.



© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

In this same scenario on a trusted OS, if the attacker gains access to web server, because of role compartmentalization and least privilege, the attacker would not have root access and would not have an easy path to the database or the inside network.

In the second scenario, let's assume an attack is coming from the inside network(Diagram 4). This is probably the biggest weakness of perimeter security because it assumes that users on the inside are good and would never do anything to cause damage. Because of this, many times there are holes that allow a malicious user to use the privileges they have to get into areas that they normally don't have access to. Because network traffic is usually open to all ports, passwords can be captured and cracked to gain higher access. Also, very common are machines that are non production machines, which are configured with no passwords or no security patches applied providing an easy path to a production server.



In the trusted OS, the same rules apply as the above scenario in that because of role compartmentalization and least privilege, the attacker would only have rights to what they were granted and would not be able to get root access.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Next let's look at configuration and administration. The trusted operating system is very complex. It requires a very well trained technical person to setup properly. Also it is very hard to administer because you can't just have full root access and change anything you want. What you pay for in very tight security, you lose in administration. In other words, if it's easy to administer, it's probably easy to break into. Because of the complexity, administrators will have a much bigger learning curve and will have to spend much more time administrating a trusted operating system. On the other side, firewall/perimeter security, is the de facto standard, and it is relatively easy to administer. The biggest configuration usually lies in securing the operating system.

Next compatibility is a big issue, most of the trusted OS's are based on the Unix flavor of operating systems. There are a few for Windows NT, but also on the compatibility issue are applications. Some applications won't work on a trusted OS. Perimeter security is very flexible in that any OS or application can be used.

Lastly, let's look at the costs between the two security models. There are many factors that contribute to the costs involved. There are hardware costs, software costs, and training and administration costs. Hardware costs between the two security models is very competitive. Trusted OS's don't require any additional hardware. As long as the operating system supports the hardware it will work. The software cost is the cost of the trusted OS or the firewall software. Most firewalls cost on average from $5000 to $15000. The trusted OS can be anywhere from $5000 per server to $50000 for an enterprise level system. So it appears that the traditional firewall is less expensive. As far as training and administration costs, the trusted OS, because of it's complexity, definitely takes more time to learn, configure, and administer which makes it more expensive.

Conclusions

From looking at these two scenarios, it looks like trusted operating systems are more secure that the traditional model of perimeter security. When I started looking at trusted operating systems as a topic, I was very impressed with what I've read. But I also believe that this technology hasn't been tested as much as perimeter security. Perimeter security is the standard, it's popular, which also means everyone understands it, especially attackers. Its weaknesses are well known and a lot of the blame falls on the operating system being used, as that is where most of the vulnerabilities are. Also, many existing setups have been done poorly and many administrators don't apply security patches to production systems.

As I was finishing my research, we've just recently seen that failure to apply security patches also affects the trusted operating system. Towards the end of April, 2001, another hack challenge called 'Argus Hacking Challenge', showed a weakness in a trusted OS. A system running Solaris with Argus's Pitbull

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

software was compromised because a security patch wasn't applied to the system. So although the trusted operating system model looks promising, there is still the issue of applying the latest security patches to keep it secure. I believe that the perimeter model will stay with us for quite a while, although I think future operating systems will incorporate the features of the trusted operating systems.

References

Robert L. Scheier, Computerworld
Trusted Operating Systems: The Ultimate Defense
http://www.computerworld.com/cwi/story/0,1199,NAV47_STO53293,00.html

Rodger O. Crockett, Business Week
Keep Out. We Mean It
http://www.argus-systems.com/press/news/cache/2000.10.23.shtml

Jim Rapoza, EWeek
Latest Pitbull attack holds lessons for IT
http://www.zdnet.com/eweek/stories/general/0,11011,2713689,00.html

Chet Dembeck, E-Commerce Times
Improve Internet Security or Face the Music
http://www.ecommercetimes.com/perl/story/?id=2492

Ronald L. Mendell, SecurityPortal
The Future of Operating Systems Security
http://securityportal.com/cover/coverstory20010115.html

Sue Hildreth, ebizQ.net
ASP Security: Why Firewalls Are Not Enough
http://b2b.ebizq.net/asp/hildreth_2.html

Peter A. Loscocco, Stephen D. Smalley, Patrick A. Muckelbauer, Ruth C. Taylor, S. Jeff Turner, John F. Farrell, National Security Agency
The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments
http://www.cs.utah.edu/flux/fluke/html/inevitability.htm

Tom Mills, Trusted Operating Systems
http://www.sei.cmu.edu/str/descriptions/trusted_body.html

eWeek's OpenHack III Challenge Survives 5.25 Million Attack Attempts
http://www.argus-systems.com/press/news/2001.02.01.eWEEK.shtml

Pitbull .comPack, Trusted OS Security: principles and practice
http://www.argus-systems.com/product/white_paper/pitbull/

© SANS Institute 2000 - 2002, Author retains full rights.

David Ferraiolo and Peter Mell
Operating System Security: Adding to the Arsenal of Security Techniques
http://www.itl.nist.gov/lab/bulletns/dec99.htm

© SANS Institute 2000 - 2002, Author retains full rights.