Global Information Assurance Certification Paper

Layered Security: A ISP Case Study with Cisco and Solaris

Rockie Brockway
Oct 19, 2000

I. Introduction

The purpose of this paper is to present a structured process and design of how to layer the security of a large scale internet presence. Our examples will focus primarily on an ISP scenario running Cisco routers and switches, appliance firewalls and servers running Solaris 8 on Sun boxes. In reality, an entire paper should be spent on each layered level, but we'll try to just get a general summary of the entire process.

II. Overview

To simplify our network design we will choose to show one border router connected to one or more uplinks (in reality there will usually be more than one border router multi-homed to several upstream providers linked into the core switch(es) in a redundant manner using HSRP for starters). Providing redundant systems in this network design is out of the scope of this paper, but is highly recommended for obvious reasons.

Our example border router will be a Cisco 7200VXR running 12.1 Service provider IOS. From the internal side of our border router we will connect that into a core layer 3 switch VLAN support. We will choose as our example core switch a Cisco 6509 with the Multi-layer Switch Feature Card (MSFC). The MSFC is set up to route between the various VLANs configured in the 6509 itself. We will break down our VLANs into server groups that we need to segment: web servers, mail servers, news servers and name servers will be our basic sample set of server groups. One great feature of layer 3 switches such as the 6509 is the ability to do ISL trunking. ISL trunking allows us to designate a port of the 6509 as a trunk port, which then connects to another ISL supported switch (like a 1900, 2900 or 3500 series Cisco switch). This trunked uplink port then carries any and all VLAN info you desire from the 6509 to the satellite switch, which can then in turn be programmed to have certain VLANs on certain ports, just like the mother switch 6509. However, although ISL trunking greatly reduces the number of flat switches necessary in the all around network design by allowing multiple VLANs to be configured on each switch, this feature set does not lend itself well to our highly layered secure design. This is because we WANT one point of entry for each VLAN that can be firewalled. If we have multiple switches in multiple areas of our network that each have a port on VLAN4, for example, it then becomes much more difficult to monitor and filter traffic for that particular VLAN.

So our design continues with one port on our core 6509 switch per server VLAN. Each port then plugs directly into a small 100Mbit hub, where we can hang a NIDS sensor. The hub then plugs directly into the outside interface of a Cisco Pix firewall (a 515UR for our example, but depending on throughput requirements this can be a 506 to 535). The inside interface of the Pix connects to a 100Mbit switch where we have our specific Solaris server farms connected. Let's take a detailed look at each layer.

III. Layer 1 – Border Routers

Every router and switch in your enterprise should have the following list of things done immediately prior to turning up live:

- Synchronize the clocks with NTP
- Set up logging to an external syslog server
- Remove default SNMP community strings and replace with your own
- Set up AAA through a remote Tacacs/Radius server
- Login banners
- Access lists on the vty interfaces to filter inbound telnet/ssh access
- Each active interface should have the following preventive measures as well
  - no ip redirects – disallows ip redirection
  - no ip directed-broadcasts – hampers DOS attacks like smurf
  - no ip proxy-arp – does not listen to arp if previous knowledge of MAC address

Your enterprises border routers are those routers that connect your autonomous system to the neighboring autonomous systems of your upstream providers. They are where initial filtering and defensive systems should be set up, and will typically be talking BGP to the uplink providers in order to get full internet routing tables. So where do we begin?

Well, let's start with some basics. We know that according to RFC 1918 (Private address space) that there are subnets which are for internal use only and should not traverse the internet, so we will filter these not only from our upstream providers, but to them just in case. We will achieve this easiest via BGP distribute lists which list the private networks we want to filter (127/8, 10/8, 192.168/16, etc) going both inbound and outbound on our BGP links.

We also know that traffic from outside of our autonomous system should not enter our network saying that it is originating from our network. This is a sure sign of spoofing. Additionally, the reverse is true. We should not allow traffic to leave our autonomous system that says it is originating from network not our own. This is achieved by creating an access-list that denies traffic inbound that says it originates from us, and another that denies outbound traffic that says it does not originate from us. They are applied to each uplink interface on the router.

It may also be prudent as well to gather a list of ports that we know should not be allowed to enter our AS from the internet. Echo, chargen, portmapper, netbios, snmp, NFS and X are good for starters. These definitions would be applied to the same inbound access list that we created to prevent inbound spoofing above.

Our edge border routers are also a very good place to attempt to prevent such DOS attacks as SYN-floods. Cisco's tcp intercept should be enabled.

IV. Layer 2 – Core Layer 3 Switch

Our core Layer 3 switch is the, fittingly, the center of our enterprise. It is here where we create our individual VLANs and assign them to ports on the 6509. Security on the switch itself, on top of the list of things which should be set up in every router and switch in section III, is basically limited to port security and ip permit lists for inbound telnet/snmp/ssh. Port security allows the configuration of a specific MAC address to a specific port, preventing the port from working if that MAC changes. The ip permit lists can be set up for allowing inbound telnet, snmp or ssh into the switch from only specified IP addresses.

The ability to route between VLANs is provided by the MSFC. Configuring the MSFC is just like configuring any IOS based Cisco router. This means we have the ability to use such things as access-lists on interfaces. So that's exactly what we'll do. Only we're not going to concentrate much on filtering at this level because we want to preserve processor time on our core device(s). Here we will just do a little more anti-spoofing on each server interface VLAN to make sure no spoofed packets come in or out and let the actual packet filtering happen at the 3rd layer, the PIX firewall.

Our Core switch will be set up to do one more thing. We will set up one or more span ports on the switch, which will be connected to high performance Network Intrusion Detection Systems, such as the Cisco Secure IDS, RealSecure or a UNIX box running snort/shoadow/NFR. This particular device must have very high throughput, especially if you are going to span all VLANs to this port and expect to look at each packet from every subnet.

V. Layer 3 – Appliance Firewall

From each individual VLAN port on our core switch, we connect directly to a 100Mbit hub so we can hang an individual NIDS box per VLAN (These NIDS sensors can again be any kind of IDS you prefer.) Also connected to the hub is this VLAN's appliance firewall. I am partial to appliance firewalls over software firewalls for one huge reason – You don't have to take more time and harden the underlying OS and then potentially worry about blowing that hardening by upgrading and patching the OS in the future. The appliance firewall, while perhaps not as user friendly as some software firewalls, usually do not need any further TLC than assigning it an IP address, setting up some filters and plugging in.

So we begin by determining what services we want available from the internet on our servers behind our pixes. Set up the ACLs (conduits are being replaced with standard ACLs as of PIX OS 5.x) and ssh server (PIX OS 5.2). PIX firewalls also perform a limited amount of protocol command filtering in the fixup commands by default. Make sure we're logging timestamps and to an internal syslog server and we're ready to move on to our last layer.

VI. Layer 4 – Host Based Defenses

Our final layer is our host based layer of defense. As stated before, our server scenario is populated with Sun boxes running Solaris 8. After our initial OS install we need to do many more things before plugging the server into the network.

First and foremost is installing the latest maintenance and security patches directly from sun. These patches are updated daily so as of this install you should have the most recent versions of every package on installed on the box.

Next is to harden the OS. There a couple of very good packages available to help harden the Solaris OS. Titan and YAPPS are the two that I like the best. You may want to write your own set of automated scripts that do things such as modify kernel parameters to help prevent DOS attacks, disallowing IP redirects, remove unnecessary init scripts, add strong TCP sequence number support, add additional log files, etc. Remove all unnecessary services from inetd (preferably all of them and remove it from it's startup script /etc/rc2.d/S72inetsvc). This is a very important step so think it through and try to avoid allowing things left on the box that is really unnecessary.

At this point I like to install some additional packages – gcc, ssh, perl_5.6, tcpwrappers, secure RPCbind, RCS, sudo, snort, nmap and tripwire. Set up tcpwrappers, make sure your sshd uses libwrap and replace rpcbind to with Weitse Venema's secure RPCbind, which uses libwrap as well. Also, I generally have a package of very useful system and network tools that are not found in the default installation of solaris that gets installed here as well – mtr, lsof, libpcap, tcpdump, top, etc. are all included in this. These tools are all instrumental in helping track down break-ins if they ever actually happen. Syslog should be configured to log to an external server to hopefully prevent log modification in the event of a break-in.

Finally install whichever server software the box will be running – apache, sendmail, innd, bind, etc. Be sure you have the latest versions and patches of these as well. If not, then all the precautions in the previous layers are moot, since we are allowing traffic to reach the ports that there servers are running on. If they themselves are vulnerable, big problems usually occur shortly thereafter.

Once we have all of our software installed, reboot and then do an initial tripwire run. The tripwire database should be moved off and preferably burned to a CDROM so it cannot be modified, but storing it on another machine for backup will suffice as well. Set up a cron to run tripwire nightly and mail you the results. You can also write your own scripts to check other things nightly such as comparing an SUID/GUID list, finding odd file names, etc.

If your servers have enough resources, you can run a host-based IDS, like snort or shadow on each box for further security info. I have a series of perl scripts that go out every hour and download each server's snort.alert file to a main snort box and the run

snortsnarf on them for hourly reports and alerts. One more layer of information does not hurt, and many say host based IDS is even more important than Network IDS.

Once finished, we can then plug our server into the VLAN switch behind our Pix and have a complete multi-layered security implementation.

VII. Conclusion

Since it is extremely important, I will state the obvious: A multi-layered security implementation is of utmost importance, and not just because we have been told this in a course. Starting from the border routers of your enterprise through the core switched network to the appliance firewalls and finally the host servers themselves, taking the appropriate measure at each level, ensuring that you are not only implementing good barriers, but also sound auditing systems and recovery policy. It is important not to forget the importance of backup policy and router and switch revision control. They themselves are another layer to keeping your enterprise secure and enable you as the administrator to recover quickly and successfully from not only break-in attempts but even standard crashes.

It is also crucial to remember that even after these steps have been made towards a secure internet presence, our work as administrators is not done. We must continue to be aware of any and all changes on our systems, monitor log files, run audits regularly, keep up on the latest security forums and patch the systems regularly.

These guidelines can assist in properly securing many different scenarios, not just the one cited here. All products cited here are replaceable and are only meant to illustrate each layer in a knowledgeable manner.

VIII. References

"Essential IOS Features Every ISP Should Consider." Version 2.6.5. 12.04.98. http://www.cisco.com/public/cons/isp/documents/IOSEssentialsPDF.zip

"Solaris Security Guide." 06.23.99. http://www.sabernet.net/papers/Solaris.html

"Configuration Guide for the Cisco Secure PIX Firewall Version 5.2." http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v52/config/

"Catalyst 6000 Family Software Configuration Guide." http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sft_6_1/configgd

Chouanard, Jean. "YASSP." 07.20.00. http://yassp.parc.xerox.com/

Venema, Weitse. "Wietse's tools and papers." ftp://ftp.porcupine.org/pub/security