

Firewall Selection

- **Single Purpose Router or a General Purpose Computer?**
 - Packet filtering should be only activity on the device
 - Combinations of proxy servers and/or bastion hosts may be implemented on routing device
 - **Serious increase in hardware performance requirements**
- **Simple specification of rules**
 - Packet filtering is complicated to begin with because the protocols are complex, rule implementation should not add complexity.
- **It should allow rules based on any header or meta-packet criteria**
 - Header information is in the packet
 - Meta-packet information are those things routers recognize outside of the header



Applying filtering rules

- **Apply rules in the order specified**
 - Reordering makes it more difficult to analyze what is going on
 - Any quirks or bugs in the rule set may be obscured
 - Reordering rules can break a rule set that would otherwise work correctly
 - **Example**
 - Rule A permits the university network to reach your research subnet
 - Rule B locks out a hostile subnet at the university out of everything else
 - Rule C disallows Internet access to your subnet
 - **Rule order ABC**
 - Packet from hostile subnet allowed to research subnet (rule A)
 - **Rule order BAC**
 - Packet from hostile subnet denied access to research subnet (rule B)
 - Rule may have limited granularity



More packet filtering guidelines

- **Allow rules to be applied separately to incoming and outgoing packets on a per-interface basis**
 - provide maximum flexibility
 - when only outgoing packets can be viewed then:
 - **The filtering system is always “outside” of its filters**
 - **More difficult to detect forged packets**
 - Forgery is most easily detected when the packet enters from outside the system
 - Routers can generate packets themselves and sometimes process internal packets (due to fixed paths for example).
 - **Filtering outgoing packets only is more complicated when the router has multiple ports**
- **Allow option to log accepted or dropped packets**
- **Support good testing and validation capabilities**



HTTP Security Concerns

- **What can a malicious client do to you HTTP server?**
- **What can a malicious HTTP server do to your clients?**
- **What else can come in tunneled over HTTP?**



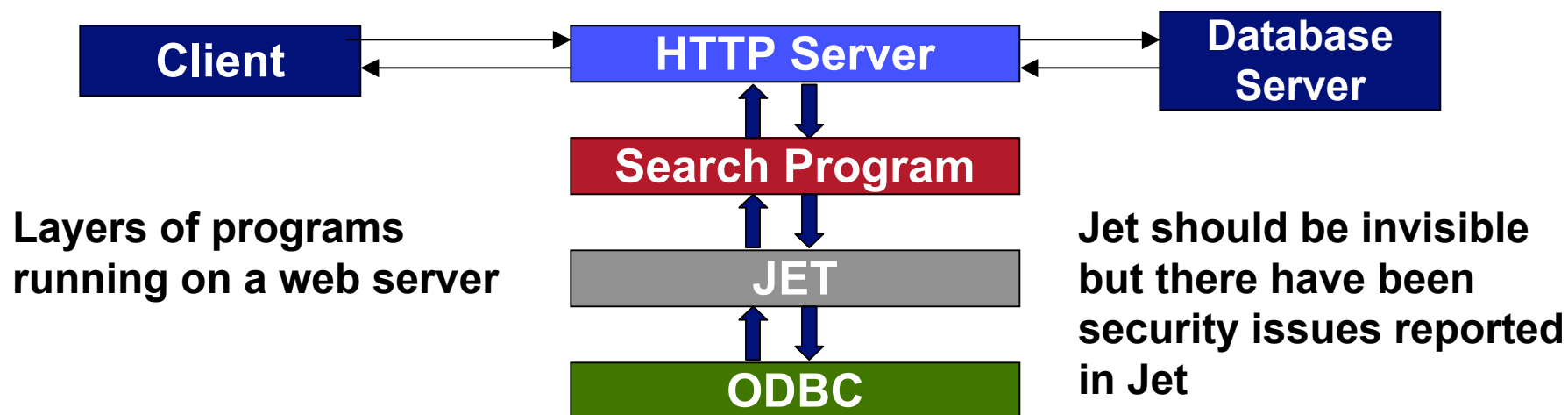
Minimize Server Exposure

- Carefully configure the security and access control features of your server to restrict its capabilities and what users can access with it.
- Run the server as an unprivileged user.
- Use filesystem permissions to be sure that the server cannot read files it is not supposed to provide access to.
- Under UNIX, use the chroot mechanism to restrict the server's operation to a particular section of your filesystem hierarchy.
- Minimize the amount of sensitive information on the machine.
- Limit the number of people who can put data on the externally visible web sites;
 - educate people carefully about the implications of publishing data.
- Maintain a clear distinction between production and development servers and specify a cleanup stage before data is moved to the production servers.



Minimize External Programs

- Ex. counter programs
 - “You are visitor #5031”
 - Must allow read/write access somewhere
- Some counter programs can be manipulated into reading or writing any file they have appropriate permissions for, and overwriting arbitrary files with counter information can do a lot of damage



Minimize External Program Risks

- **Install external programs only after you have considered their security implications and tested them on a protected machine.**
- **Run as few external programs as possible**
- **Run external programs with minimal permissions**
- **Don't assume that programs will be accessed from pages or CGI forms you provide.**
- **Develop special bastion host configurations for external programs, going through and removing all unneeded files and double-checking all permissions and placements of files that are read and written.**



Common Errors

- **Taking a development tool or web server package and installing it on a production machine without removing sample programs or other development features.**
 - These are often proof-of-concept or debugging tools that provide very broad access.
- **Running external programs with too many permissions, either by using an overly powerful account**
 - (root, under UNIX, for instance),
 - or the same account for a number of different external programs,
 - or a default account provided by a software vendor with normal user access to the entire system
 - (such accounts also often have a known name and password).



Recommendations Summary for HTTP

- **If you're going to run an HTTP server**
 - use a dedicated bastion host if possible.
 - carefully configure the HTTP server to control what it has access to.
 - in particular, watch out for ways that someone could upload a program to the system (via mail or FTP for example) and then trick the HTTP server into executing it.
 - carefully control the external programs your HTTP server can access.
- **Proxying HTTP is easy, and a caching proxy server offers network bandwidth benefits as well as security benefits.**
- **Do not allow external connections to HTTP proxy servers.**
- **Configure your HTTP clients carefully and warn your users not to reconfigure them based on external advice.**

