System Structure

Designing a System

- We have lots of tools
- Tools are rarely interesting by themselves
- Let’s design a system. . .

Some of Our Tools

- Encryption
- Authentication mechanisms
- Access control mechanisms
- Confinement mechanisms
- Lots of computers—hardware is relatively cheap

What Should We Build?

- Let’s build a large e-commerce site
- Lots of machines
- Lots of processes

What Are the Pieces?

- Web server itself
- A *replicated* web server, for load-sharing and reliability
- Databases
- Firewalls (well, we haven’t covered those)

More Pieces

- Customer care
- Mail servers
- Links to suppliers, banks, shipping companies, etc.
- NOC
- System adminstrators
- Webmasters
- Development machines

Even More Pieces

- Developer access
- Tier N customer care—often your top developers
- Backup servers
- Geographically diverse machines?
- Environmental control systems – air conditioning, power, backup generators
- Console servers
- Personnel machines (stay tuned)
- DNS servers
- KDC and/or CA
- Probably a lot more

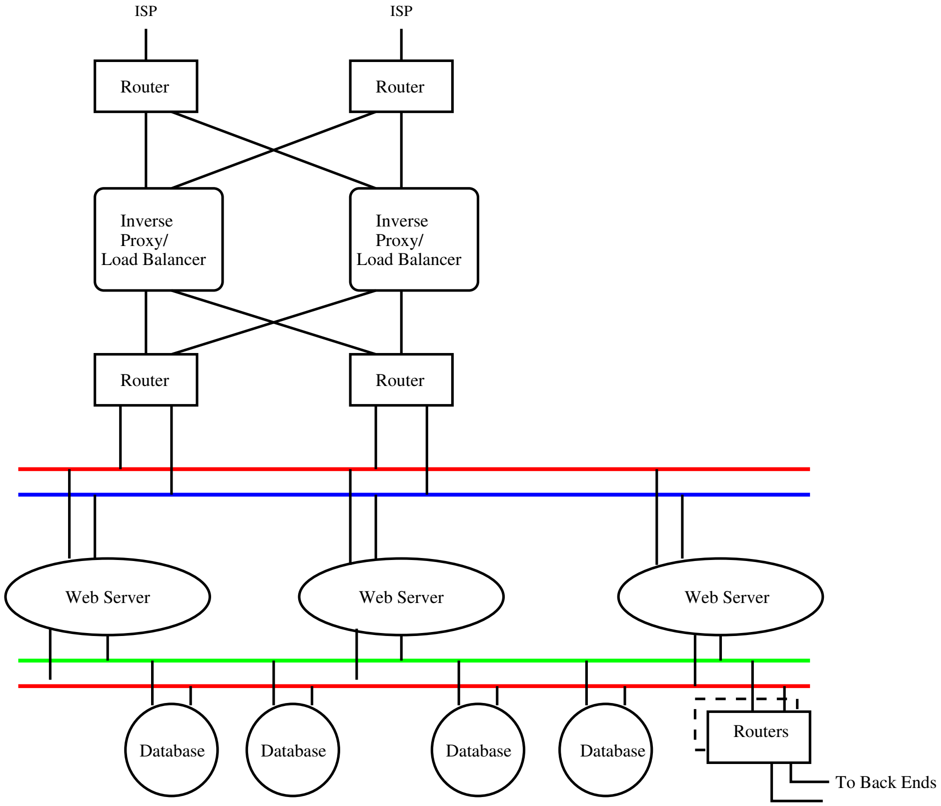
How Do We Connect These?

- First—which pieces go on separate machines?
- Does that even matter?
- Yes. . .

A Real-World Example

- I looked at the internal documentation for a billing system
- Four different databases
- 18 other processing elements
- Transaction and web inputs; external link to credit card processor

The Web Server Complex



Why So Complex?

- Availability—primarily against ordinary failures
- *Everything* is replicated: routers, links, Ethernets, servers, etc.
- The only security feature of the redundancy is some protection against DDoS attacks if your two routers connect to different ISPs

Other Security Functionality

- The inverse proxies are effectively firewalls – they only pass ports 80 and 443
- The database servers are not accessible from the outside—you have to hack through a web server to get any access at all

What are the Danger Points?

- How are these devices *managed*?
- How does the NOC talk to the routers? How is software upgraded on the Web servers?
- *Something* is missing
- The link to the back-end systems—how is that protected?

Managing the Network Elements

- Some NOC machine has to be able to talk to the network elements
- The top (“north”) routers and the inverse proxies are exposed to the public Internet
- Must use strong cryptographic authentication
- Add network access control to limit sites that can talk to them
- What about the middle routers?
- Does the inverse proxy permit access to them? That’s a slight weakening of the firewall functionality
- Are they reached from the north LANs? Does that weaken the protection of the database servers?

How Many NOCs?

- The NOC really needs to see all network elements
- To talk to the south and middle routers, it has to be on the inside
- To talk to the north routers, it has to be able to reach the outside
- Some problems are most easily diagnosed if you have the ability to connect to all of the elements
- Conclusion: we need a special firewall for the NOC machines

What are Our Goals?

- The usual trilogy: confidentiality, integrity, availability
- Ah—but what resources need what type of security?
- What are the consequences of failures?

If the Web Server is Hacked. . .

- Embarrassment—see, for example, `http://www.zone-h.org/archive`
- Passwords from active users
- Access to the database machines?

☞ Passwords for your userbase are stored in a database

How to Protect Web Servers?

- Use strategies already discussed
- Major advantage—these are dedicated machines, with no ordinary users
- Can we use separate UIDs?

Separate UIDs on the Web Server

- In general, it’s a good idea
- Principle of least privilege—protect different functions from each other
- Example: login CGI script runs as a different user than the browsing CGIs, to protect the user password database
- Often harder than it seems—functions interact a lot

Should We Chroot the Web Server?

- What do we protect if we do that?
- The rest of the system? Maybe—but there's nothing else on the system
- It doesn't hurt, but it isn't that big a help
- This machine is a web server appliance

The Database Machines

- These are the crown jewels of the company
- If the database is tampered with, very bad things can happen, including loss of lots of money
- How are the database machines attacked?
  - Hacking attack—first the web server falls, then the database machine
  - Database queries and changes—the web servers have access to the database; therefore, the hackers on that machine do, too
  - SQL injection attacks through the web server

Databases for a Simple Configuration

- User information: login, password, credit card numbers, shipping and billing addresses, etc.
- Orders: active orders, past orders, etc.
- Inventory: stock on hand, prices, etc.

Hacking Attacks

- This is nothing special—it’s just another machine to lock down
- The database machine runs very different software than the web server does; probably no common mode failures
- Not a major threat—but don’t forget to lock it down

Database Query Attacks

- The web server can perform lots of database operations
- How do we stop the hacker from doing them?
- Answer: have the customer log in to the database!
- That is, all customer-type operations must be accompanied by a customer-specific authenticator
- A compromised web server machine can only modify database records for *active* users

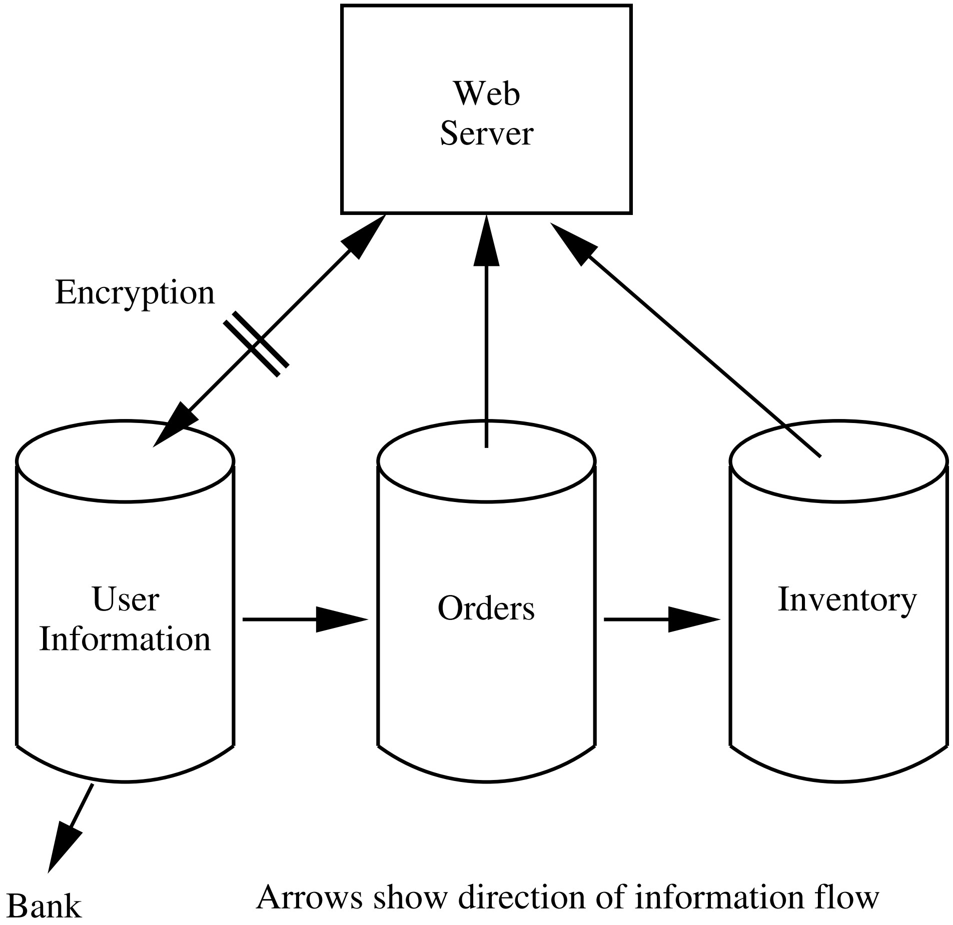
More Generally

- Let’s adjust the control and information flow
- Only let the web server write what it has to
- Thus: the user database creates the order, *not* the web server
- Why? Because the web server is much more exposed and hence vulnerable

What About the Inventory Database?

- When a customer buys something, the inventory needs to be adjusted
- That isn't customer-specific—how do we secure that database?
- Use a separate database—and database machine—for orders; the order database can adjust the inventory
- Put sanity-checking and customer limit enforcement into that machine, too
- (This is an oversimplification; the shipping database also needs to adjust the inventory in may situations)

Information Flow Among the Databases



Arrows show direction of information flow

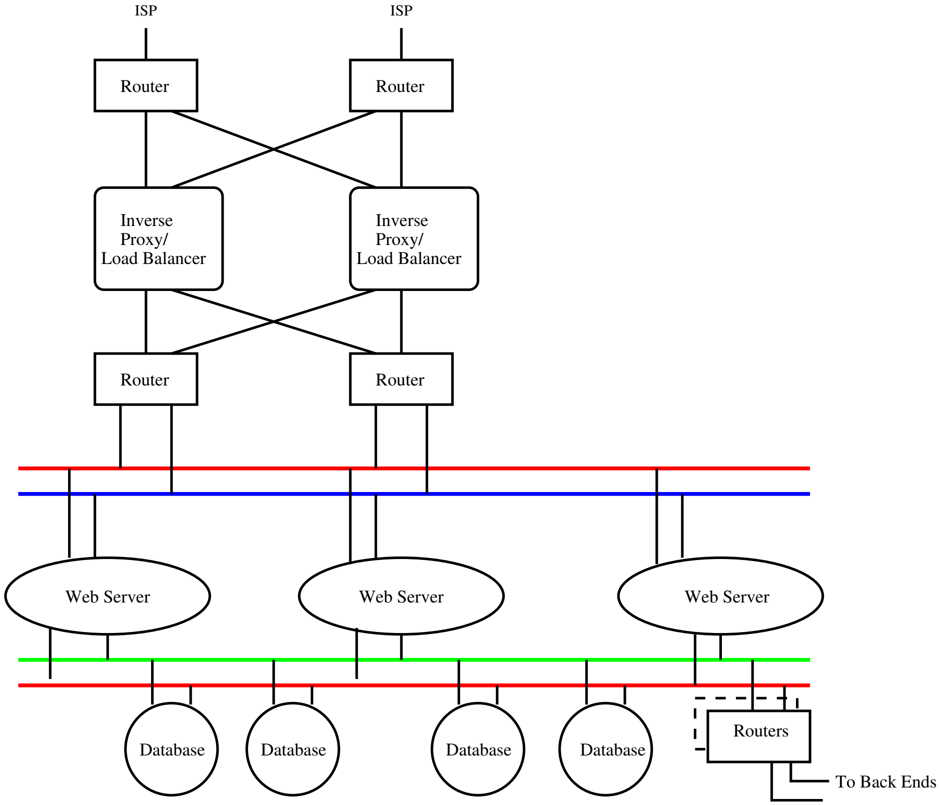
Protecting Information

- The databases will only reveal certain information to certain other machines
- Example: credit card numbers are not readable by the web server or even the order server.
- The web server can tell the user information machine to send a debit message to the bank

What About a Small Site?

- Divide the functions up the same way
- Use chroot() and equivalent to isolate the different components
- We are using chroot() here as a weaker version of separate machines

The Biggest Weakness?



That Link to the Back End

- The link in the southeast to back-end systems is the most dangerous part of the diagram
- What is the protection from the rest of the company?
- What essential functions are there?

Two Major Classes

- Normal operations
- Unusual circumstances

Normal Operations

- Customer care
- Outside links
- Mail servers
- These go on a LAN with reasonable access to the server complex—customer care, for example, has to be able to read and write several databases

Protecting Us from Customer Care

- What do we do about a rogue customer care agent?
- Prevent access to some kinds of information (i.e., credit card numbers)
- Log everything; audit periodically

☞ Route all requests through a logging proxy

- “This call may be monitored for quality assurance purposes”

Isolating Customer Care

- Customer care is often outsourced or done from home
- How do you protect home computers?
- Supply agents with locked-down laptops
- Use a VPN and firewall that go *only* to the customer care proxy/logging machine
- Be ready to ship replacement machines in case of trouble

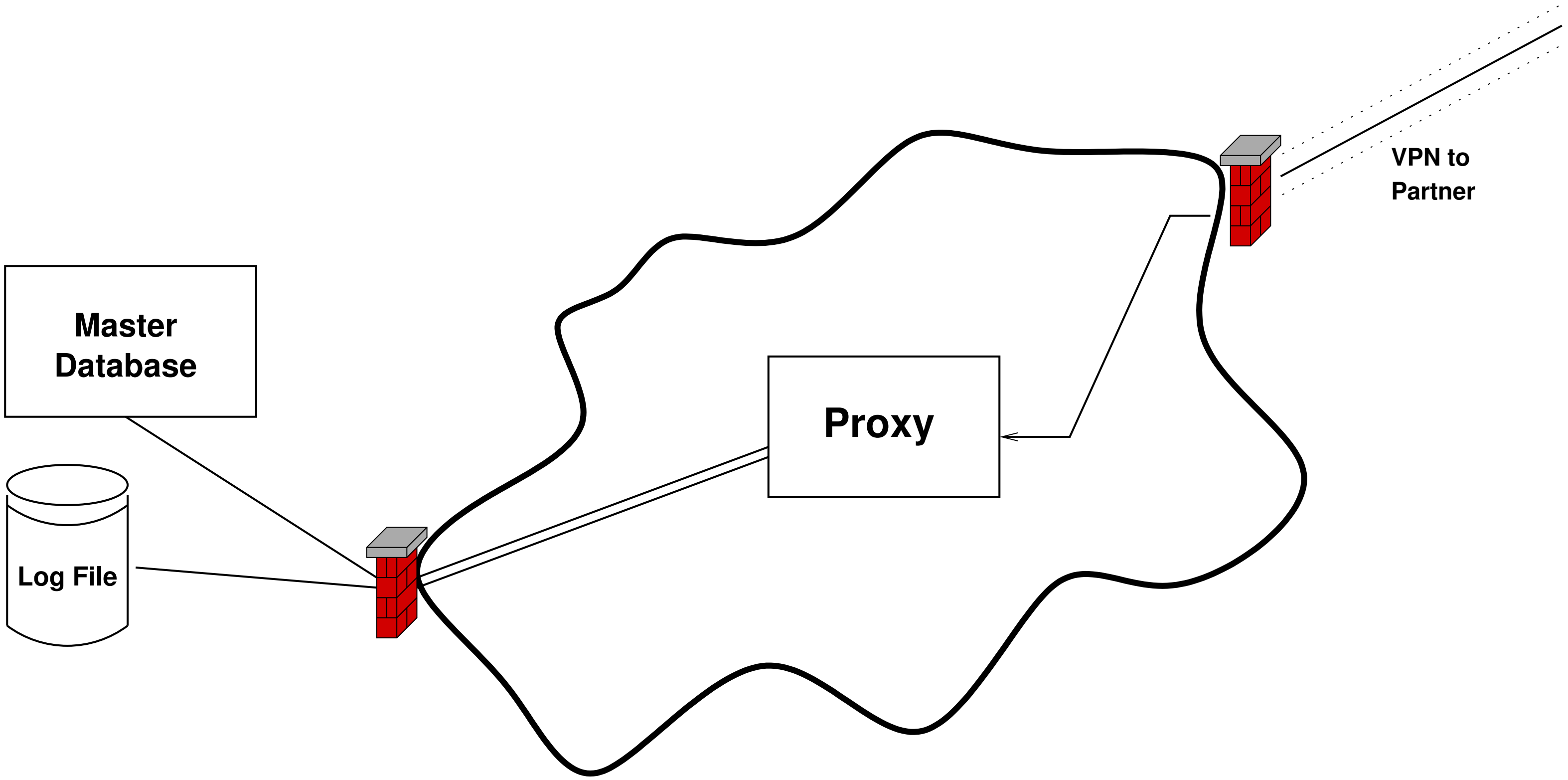
Outsourced Functions

- Many other functions might be outsourced
- If there are interconnections, you can be attacked that way

☞ According to press reports, Target was hacked via their HVAC contractor

- Isolate the networks
- Use enclaves with proxies and logging

Enclaves



Unusual Circumstances

- Maintenance
- Problem recovery
- Out-of-hours emergencies
- Put a gateway between these functions and the server complex

Maintenance

- On a production system, maintenance is a scheduled activity
- This means that access controls can be relaxed during that window only
- Be careful about relaxing the controls only to the extent necessary

Problem Recovery

- Problem don't occur on a nice, neat schedule
- Often, development staff has to do the repairs
- How do we mediate access?

Remote Access

- Often, development staff has to do the repairs at night, from home
- Sometimes, the development staff is in another country
- We can’t institute a “physical access only” rule; it’s not realistic

Two Types of Access

- VPN access to the site
- (This may be normal anyway.)
- (Does your answer change if you use an off-site hosting company?)
- Authenticated access to the server complex

Forms of Authentication

- What types of authentication should be used?
- Differs by function
- The NOC can’t use anything that requires access to external servers; they have to talk when the network isn’t working well
- Customer care agents can use more or less anything to log in to their machines—but their machines would have credentials to permit database access
- Scheduled maintenance can use anything, but something secure should be used to relax access controls
- Emergency maintenance personnel need cryptographic keys for the VPN, and something secure—a token?—to get to the server complex. Note, of course, that the server complex may be experiencing connectivity problems. . .

Revoking Access

- What do you do when an employee leaves?
- This is especially serious for employees with special access rights
- Solution: link the HR database to authentication servers
- Have some other way, driven by the HR database, to revoke access to other resources
- Caution: this means giving the HR machine a lot of access

General Principles

- There’s no one solution; a lot depends on context and budget
- Strive for separation of function
- Reserve your strongest controls for the most valuable pieces
- Log everything (but we talked about that already. . .)

The Final

The Final

- Wednesday, December 23, 1:10-4:00
- Open book
- Open notes
- No computers
- Same style as the midterm
- *Cumulative*, but a stress on the second half; some questions may span both halves
- Nominally 170 minutes, but I'm going to aim for significantly less