# Web Security
# CS 236
# On-Line MS Program
# Networks and Systems Security
# Peter Reiher

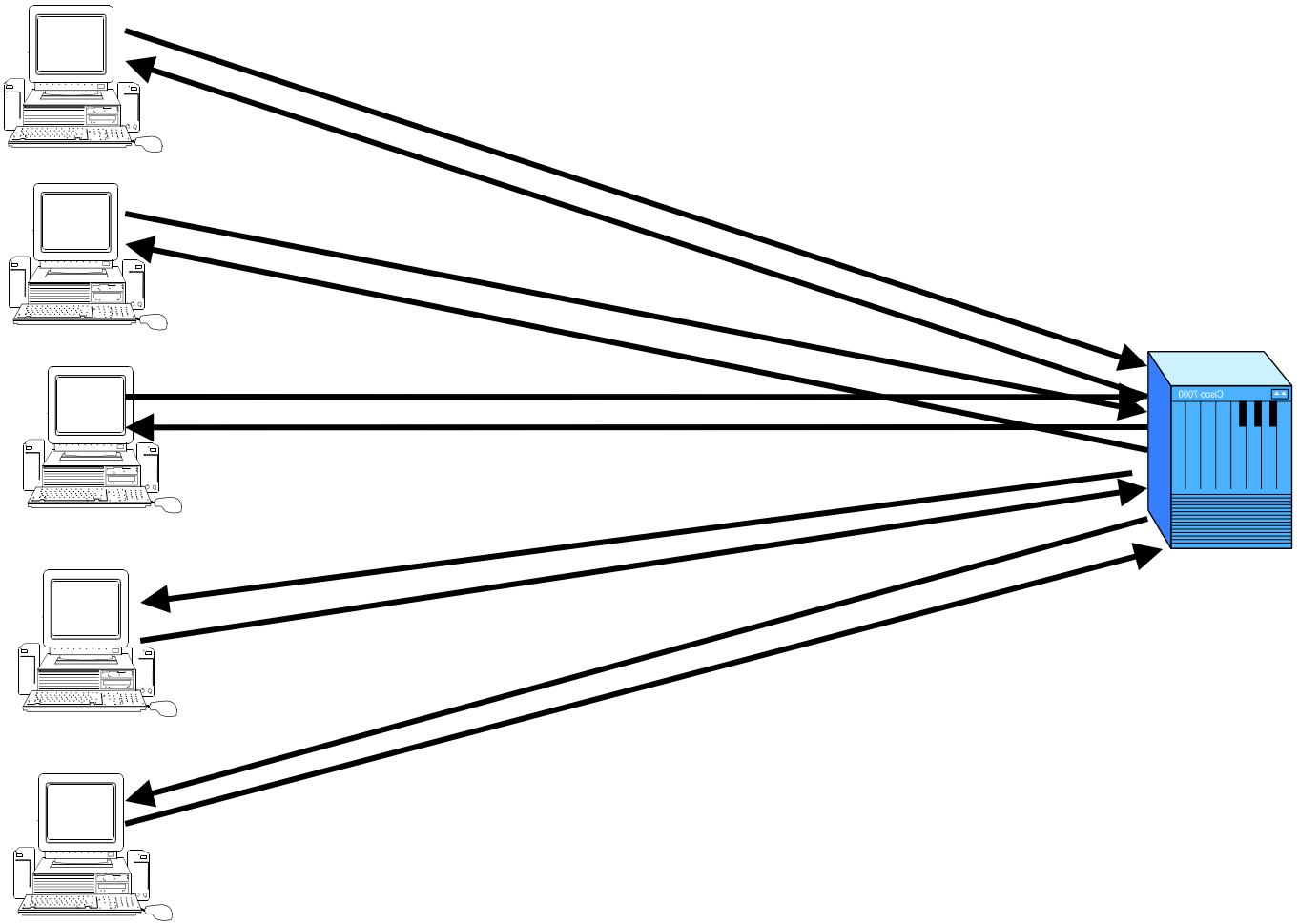
# Web Security

- Lots of Internet traffic is related to the web
- Much of it is financial in nature
- Also lots of private information flow around web applications
- An obvious target for attackers

# The Web Security Problem

- Many users interact with many servers
- Most parties have little other relationship
- Increasingly complex things are moved via the web
- No central authority
- Many developers with little security experience
- Many critical elements originally designed with no thought to security
- Sort of a microcosm of the overall security problem

# Aspects of the Web Problem



# Who Are We Protecting?

Everyone

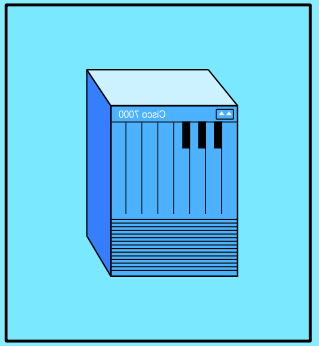
From the client

The clients

From the network

A client's interaction with one server

From the server



From his interaction with another server

The client

The server

From each other

# What Are We Protecting?

- The client's private data
- The server's private data
- The integrity (sometimes also secrecy) of their transactions
- The client and server's machines
- Possibly server availability
  - For particular clients?

# Some Real Threats

- Buffer overflows and other compromises
  - *Client attacks server*
- SQL injection
  - *Client attacks server*
- Malicious downloaded code
  - *Server attacks client*

# More Threats

- Cross-site scripting
  - *Clients attack each other*
- Threats based on non-transactional nature of communication
  - *Client attacks server*
- Denial of service attacks
  - *Threats on server availability (usually)*

# Yet More Threats

- Browser security
  - Protecting interactions from one site from those with another
  - *One server attacks client's interactions with another*
- Data transport issues
  - *The network attacks everyone else*
- Certificates and trust issues
  - *Varied, but mostly server attacks client*

# Compromise Threats

- Much the same as for any other network application
- Web server might have buffer overflow
  - Or other remotely usable flaw
- Not different in character from any other application's problem
  - And similar solutions

# What Makes It Worse

- Web servers are complex
- They often also run supporting code
  - Which is often user-visible
- Large, complex code base is likely to contain such flaws
- Nature of application demands allowing remote use

# Solution Approaches

- Patching
- Use good code base
- Minimize code that the server executes
- Maybe restrict server access
  - When that makes sense
- Lots of testing and evaluation
  - Many tools for web server evaluation

# Compromising the Browser

- Essentially, the browser is an operating system
  - You can do almost anything through a browser
  - It shares resources among different "processes"
- But it does not have most OS security features
- While having some of the more dangerous OS functionality
  - Like arbitrary extensibility
  - And supporting multiple simultaneous mutually untrusting processes

# But My Browser Must Be OK . . .

- After all, I see the little lock icon at the bottom of the page
- Doesn't that mean I'm safe?
- Alas, no
- What does that icon mean, and what is the security implication?

# The Lock Icon

- This icon is displayed by your browser when a digital certificate checks out
- A web site provided a certificate attesting to its identity
- The certificate was properly signed by someone your browser trusts
- That's all it means

# What Are the Implications?

- All you know is that the web site is who it claims to be
  - Which might not be who you think it is
  - Maybe it's `amozon.com`, not `amazon.com`
  - Would you notice the difference?
- Only to the extent that a trusted signer hasn't been careless or compromised
  - Some have been, in the past

# Another Browser Security Issue

- What if you're accessing your bank account in one browser tab
- And a site showing silly videos of cats in another?
- What if one of those videos contains an attack script?
- Can the evil cat script steal your bank account number?

# Same Origin Policy

- Meant to foil such attacks
- Built into many web scripting languages
- Basically, pages from a single origin can access each other's stuff
- Pages from a different origin cannot
- Particularly relevant to cookies

# Web Cookies

- Essentially, data a web site asks your browser to store
- Sent back to that web site when you ask for another service from it
- Used to set up sessions and maintain state (e.g., authentication status)
- Lots of great information about your interactions with sites in the cookies

# Same Origin Policy and Cookies

- Script from one domain cannot get the cookies from another domain
  - Prevents the evil cat video from sending authenticated request to empty your bank account
- Domain defined by DNS domain name, application protocol
  - Sometimes also port