

1.264 Lecture 18

**Security: certificates, SSL
Banking, monitoring**

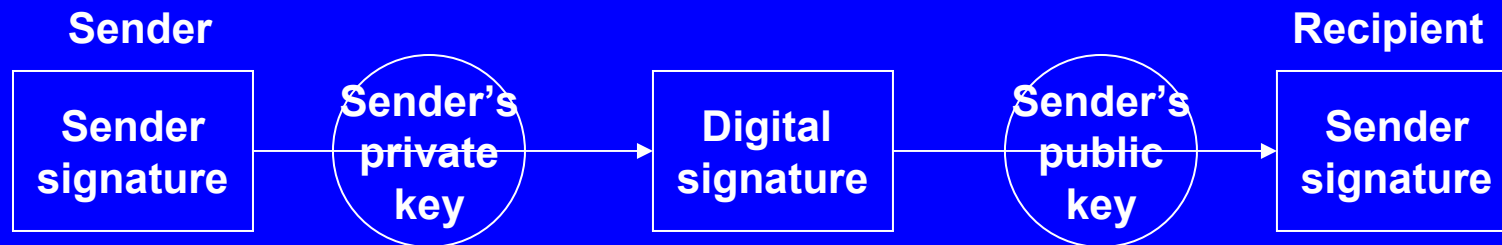
Case study 3

- **Network attack and defense**
 - List 4 strategies that firewalls and SSL don't address
 - List measures to mitigate these risks

Case study 4

- **What is the credit card fraud rate in various countries?**
- **What is the merchant discount?**
- **What fraction of merchant discount goes to**
 - **Fraud costs**
 - **Interest costs**
 - **Incentives**
- **How do credit card companies make money?**
 - **Focus on the US**
- **What implications does this have for your organization?**

Digital signatures



- **Use public/private key in opposite fashion from message encryption to provide sender authentication**
 - Sender signs document with her private key
 - Receiver decrypts with sender's public key
 - If the decryption is correct, message must have been sent by sender
- **Compare:**
 - **Encryption:**
 - Sender signs message with receiver public key and sends
 - Receiver decrypts with her private key
 - This allows any sender to send secure messages to any receiver
 - Secure Sockets Layer(SSL) distributes public keys– covered next
 - **Digital signature:**
 - Sender signs message with own private key and sends
 - Receiver decrypts with sender's public key
 - This allows any receiver to verify the sender of any message

Digital signatures, cont.

- **Problems with digital signatures**
 - **Spoofers can cut and paste encrypted signature from old message to new faked message.**
 - One solution is for receiver to send 'challenge phrase' to sender
 - Sender then encrypts with private key and sends to receiver, who can check if it's what she sent initially
 - **Spoofers can alter parts of the message**
 - **Solution is message digest functions to provide integrity check**
 - Message digest is function run on entire message that produces short digest, often 128 bits (note that 2^{128} is a very big number of combinations!)
 - Send hash and message. Receiver hashes message and checks if same hash.

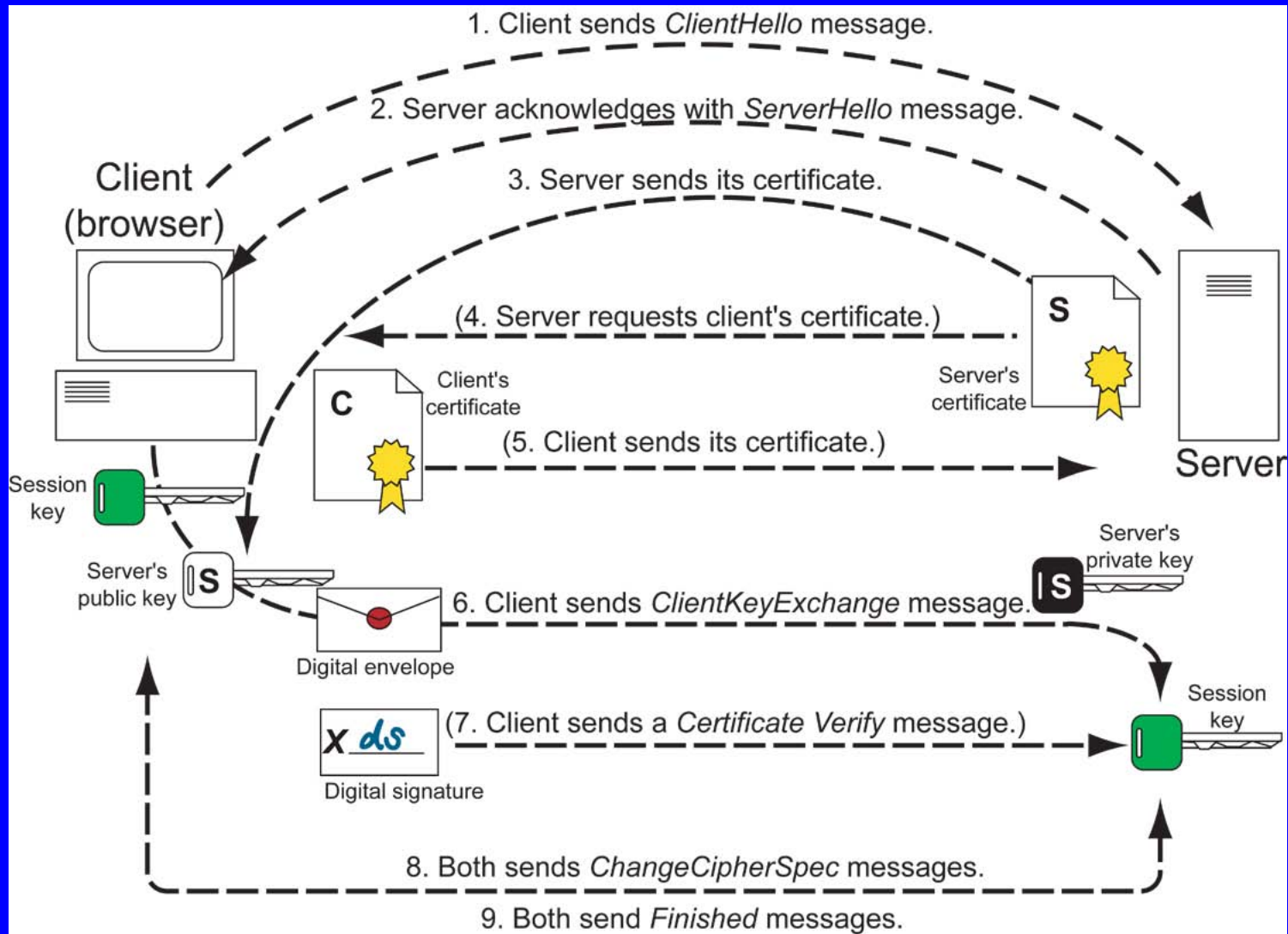
Digital envelopes

- To solve performance problems with public key encryption
 1. Client generates session key, a secret symmetric key, at random
 2. Client encrypts msg using session key and symmetric algorithm
 3. Client encrypts session key with receiver's public key: digital envelope
 4. Client sends encrypted message and digital envelope to receiver
 5. Receiver uses her private key to decrypt envelope and get session key
 6. Receiver uses session key to decrypt message
 7. When session is over, both parties discard session key
 8. Optionally, digital certificate could be used at start of session to verify client identity
- Secure Sockets Layer (SSL) essentially implements this
 - Most widely used security system on Web

Secure Sockets Layer (SSL)

- **Dominant protocol for browser-server communications**
 - Being standardized as Transport Layer Security (TLS), TLS 1.0 is essentially the same as current SSL 3.0
- **Many choices in symmetric algorithm, message digest and authentication.**
 - Uses RSA public key asymmetric algorithm.
- **Client and server negotiate strongest common protocol**
- **SSL has built-in compression**
 - Encrypted message has no patterns and can't be compressed, so compression must be done before or within SSL, or not at all
- **SSL encrypts all client-server communications**
 - Only public info available is that client is talking to this server
 - IP protocol (IPsec) must be modified to mask endpoints

SSL protocol steps



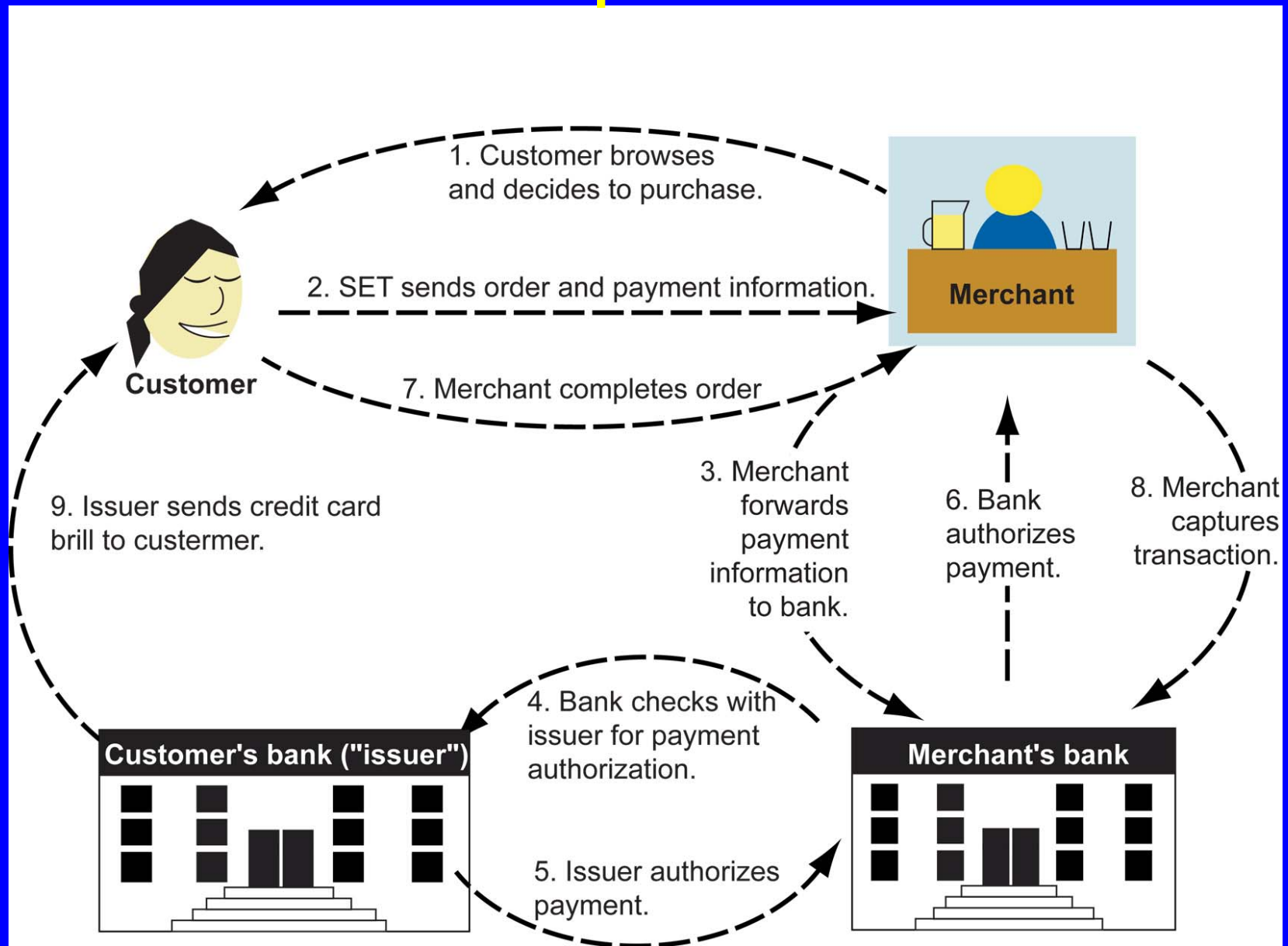
SSL protocol steps

1. **ClientHello**: list client capabilities: SSL version, algorithms
2. **ServerHello**: chosen algorithms
3. **Server sends its certificate**
 - Client can verify server certificate with root CA in browser
- 4-5. **Optional, rarely done today (but done at MIT)**
6. **ClientKeyExchange**: Client generates trial symmetric key, encrypts it with server public key and sends it to server
 - This prevents a listener (snooper) from copying past message (spoofing)
7. **CertificateVerify**: Optional, rarely done. Can authenticate client
8. **ChangeCipherSpec**: Confirm session key and cipher to be used
9. **Finished**: Client and server message digest entire conversation to ensure all messages were received intact
10. **Client and server switch to encrypted mode using symmetric session key**

Secure Electronic Transactions (SET)

- **Joint development: Visa, Mastercard, Netscape, Microsoft**
- **Used only for secure credit and debit card transactions**
 - **Has same features as current credit card system:**
 - Registration, purchase requests, authorization, funds transfer, ...
- **SSL handles encryption but not the financial functions**
 - Need online card authorization, bank transactions, etc.
 - SSL does not encrypt financial database on server; SET does
 - SET does not give merchant the customer credit card number
 - SET inhibits guessing credit card numbers (shuts off multiple guesses)
- **SET doesn't handle non-financial functions, has been abandoned**
 - We use it as an example of 'blind signatures', for which we will need a technology and standards in the future
 - One major problem: merchants like to use credit card as user ID

SET protocol



Certifying authorities and public keys

- How to obtain public keys
 - Can't keep all possible recipient keys on your system
 - Can't ask recipient to send it to you, because he/she might be spoofer
 - Don't (can't?) have large public database:
 - Costs, performance for hundreds of millions of keys a concern
- Certifying authorities (CAs)
 - Commercial entities, whose public keys are in your browser
 - Ask user to send you their digital certificate, signed by a CA, before you communicate with them
 - From certificate, you verify their identity and get their public key
 - CA encrypts the certificate and its hash with the CA's private key
 - User (you) decrypts the certificate with the CA public key and check the hash to make sure the certificate has not been altered

Obtaining a digital certificate for your server

1. Generate a public/private key pair on your system
2. Keep private key and send “certificate request” to CA:
 - Includes public key and identifying information about you
3. Pay CA fee
4. CA verifies your identity, cursorily or extensively
5. If you are ok, CA creates certificate body with your public key and ID info:
 - Server (“site”) certificate has URL
 - Browser (“personal”) certificate has name and email address
6. CA generates message digest from certificate and signs it with its private key, creating the actual certificate
7. CA sends certificate to you.

Root CAs and certificate chains

- **Root CAs have certificates on browser or Web server**
 - Must believe Microsoft, Netscape and your system vendor are ok
- **Root CAs can sign other CA's public keys**
 - Signature includes the root CA's certificate
 - Chains of certificates can be created
 - Last certificate contains certificates of every CA in the chain, so it can be traced back to the original root CA
 - You use first CA's public key to get 2nd CA's public key, which you use to get 3rd CA's public key, etc.
 - Chains typically used in intranets today to verify browsers via a chain of servers
 - SET and digital payment schemes also use chains
- **It's reasonable to generate and administer your own public and private keys when number of sites is limited**

Certificate problems

- **Events invalidating public/private key pair**
 - Theft, change of ID info, compromise of key
 - Disk corruption (private key is encrypted via password on disk)
 - Certificate revocation list (CRL) intended but often not implemented
 - Technically, should check against CRL before communicating
 - Certificates generally expire in a year, but that's a long time...
- **Privacy is impossible, because you are identified to other party**

Banking systems

- **Double entry bookkeeping**
- **Clark-Wilson security model (see text)**
 - Formal definitions of electronic bookkeeping
 - Separation of duties is most troublesome principle
 - Prevent-detect-recover model
- **What goes wrong (see text)**
 - Insider activity is the worst problem. One study:
 - 82% of fraud committed by insiders
 - Half had been at bank over 5 years
 - One third were managers
- **Telecom examples of fraud (decades worth...)**
 - Supervisor also hands-on in small office (no checks)
 - 2nd and 3rd shifts, weekends
 - High turnover offices. Supervisors, coworkers don't know everyone
 - Ultra-low turnover offices. Standard practice not followed.
- **Fraud occurred disproportionately in situations where the 'standard assumptions' didn't hold**
 - These were known beforehand

Banking, cont

- **Bank-bank transfers handled by SWIFT**
 - Protection through substantial manual supervision
- **Automatic teller machines (ATMs) and point of sale terminals (POS) and transit ticket machines**
 - Hardware security modules used for PINs, encryption
 - Dual channels to send card, PINs, generate PINs, etc.
- **What goes wrong**
 - Software PINS and encryption seen by programmers
 - Too many banks for bank-bank keys
 - Central switches vulnerable to dishonest staff
 - Data processing errors (high volume, some errors)
 - Thefts of cards from the mail system
 - Fraud by bank staff
 - Almost no sophisticated technical fraud has occurred

Monitoring: how to infiltrate a transit system

- **Intruder enters via fire exit or maintenance access that has conventional lock. No smartcard, no camera because vulnerability is too low. He gets into tunnel with low light conditions. He then...**
- **Intruder enters station and hides in storage area or unused booth or... At night, when there is no lighting he sets something in place for the next day...**
- **Intruder waits for a stormy night. He sets off an alarm at portal. Security comes and finds nothing. He waits a half hour and does it again. Security doesn't bother. He then...**
- **Intruder goes to transit station and leaves smoke grenade on timer. When fire service responds, he comes in with them during the general chaos and...**

Monitoring: how to infiltrate a transit system

- The attacker disables non-redundant communications from a sensor. Repair doesn't occur until the next day. He enters a tunnel and...
- The attacker calls claiming to be from the camera or sensor vendor and needs the serial number on a unit. It's given to him, the staffer not knowing the serial number is also the crypto key for the communications with the unit. The attacker buys or steals a similar unit, reprograms it, and attaches it to the network. Your network is redundant and uses non-transit right-of-way. The rogue device continues to report 'all is well' as...
- The attacker breaks a device or comm line, waits to see what security personnel arrive, and waits to see them leave. The device won't be fixed until the next morning. He has several hours to...

**These are variants on text's "How to Steal a Painting"
(See the section on taximeters, tachometers too)**

Network attack: top vulnerabilities

- **Windows services: COM, print, plug-and-play, etc.**
 - Buffer overflows allow attackers to get control of system
- **Internet Explorer:**
 - Malicious Web page vulnerability, memory corruption...
- **Windows libraries**
- **Office and Outlook Express**
- **Windows weak/default passwords**
- **Backup software: compromised to obtain sensitive data**
- **Antivirus software: buffer overflows and evasion software**
- **PHP software: many weaknesses, very popular framework**
- **Databases: buffer overflows, SQL injection, weak passwords**
- **P2P software**
- **DNS: cache poisoning, open recursive servers**
- **Media players**
- **Instant messaging**

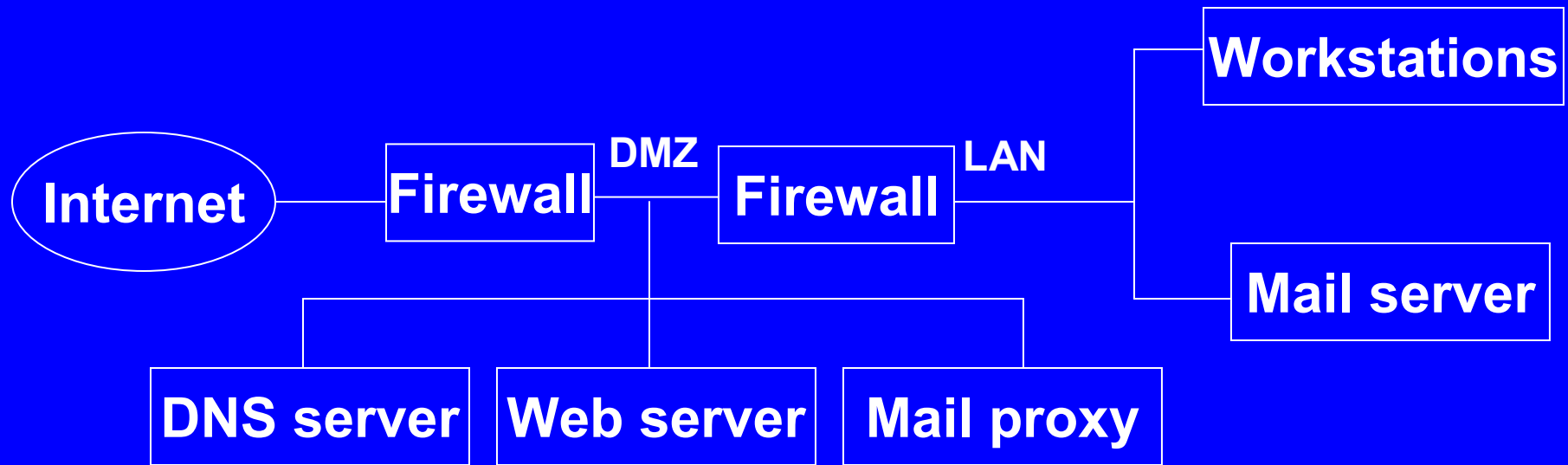
Network attack

- **Motivations for attack:**
 - **Spam (roughly 1 cent per message sent)**
 - Perhaps 30% of consumer PCs are compromised and are used as spam servers
 - Bots and zombies
 - **Blackmail and extortion**
 - Sports gambling, medical data, online services
 - Denial of server attacks
 - **Credit card fraud**
 - Phishing, database attacks, others

Network attack: firewalls

- **Without firewalls, weakest PC on internal network can be attacked and compromised, and in turn compromise others**
 - Attacker replaces some PC operating system calls with his/her own, or wreaks other havoc
 - Almost impossible to monitor all PCs for suspicious activity
- **Firewalls place special machine between internal network and Internet**
 - Direct traffic from PCs to Internet not allowed; all traffic via firewall
- **Firewall runs stripped-down version of Windows or UNIX**
 - No unnecessary services, no untrusted software
 - Logs of all activity
 - Application level proxies inspect packets (intrusion detection)
 - Layer 7: HTTP (port 80) proxy can search for and disallow binary payloads
 - Layer 3 and 4: Packet-level proxies look at headers, destinations...

Firewall configuration



Firewall:

- Filters based on TCP/IP headers (source, destination...)
- Disallows direct connections across firewall (proxy)
- Application level firewall inspects packet contents
 - E.g., scans email for viruses, HTTP for binaries
- Audit

Security Realities

- **People are the major issue**
 - Tiger team results
 - Simple passwords, on post-it notes. Etc.
- **Security in most systems is weak**
 - Even with certificates, encryption, firewalls, if a password is weak, you can get someone's information
 - Most software has a lot of bugs, and some can be exploited by attackers. Unless software is correct, security is hard to do.
- **Security is based on**
 - Prevention: client, server, network configuration
 - Detection: firewall and other analysis
 - Response: software, hardware, configuration changes
- **Security always has a large performance hit**

More information

- **“Practical Cryptography“, Schneier**
- **comp.risks newsgroup**
- **Proceedings of the Annual IEEE Symposia on Security and Privacy**
 - Applied security
 - Available online (MIT libraries)
- **www.cert.org**
 - Advisories, patches
- **www.counterpane.com (Bruce Schneier)**
 - Monthly Crypto-gram newsletter