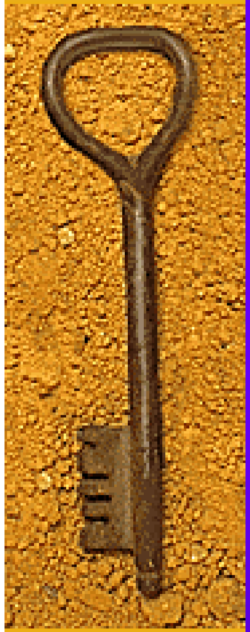
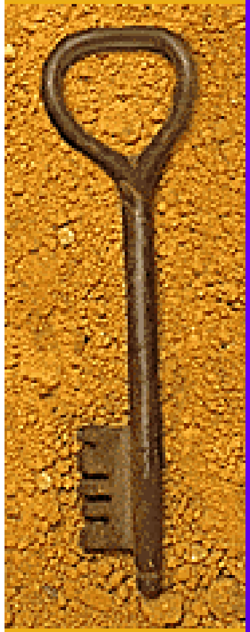
Perimeter Security - Firewall



Topics

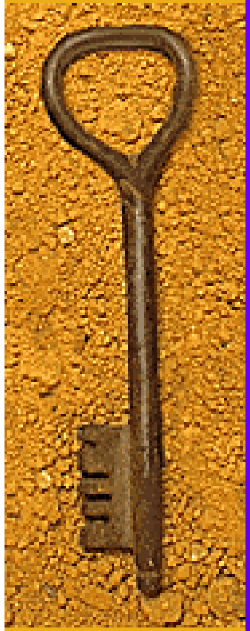
- Background of Perimeter Security
- Firewalls
  - Basic Firewall Concepts
  - Packet filter (stateless)
  - Stateful firewall
  - Application-layer gateway
- Problems with Firewalls
- Real Firewalls

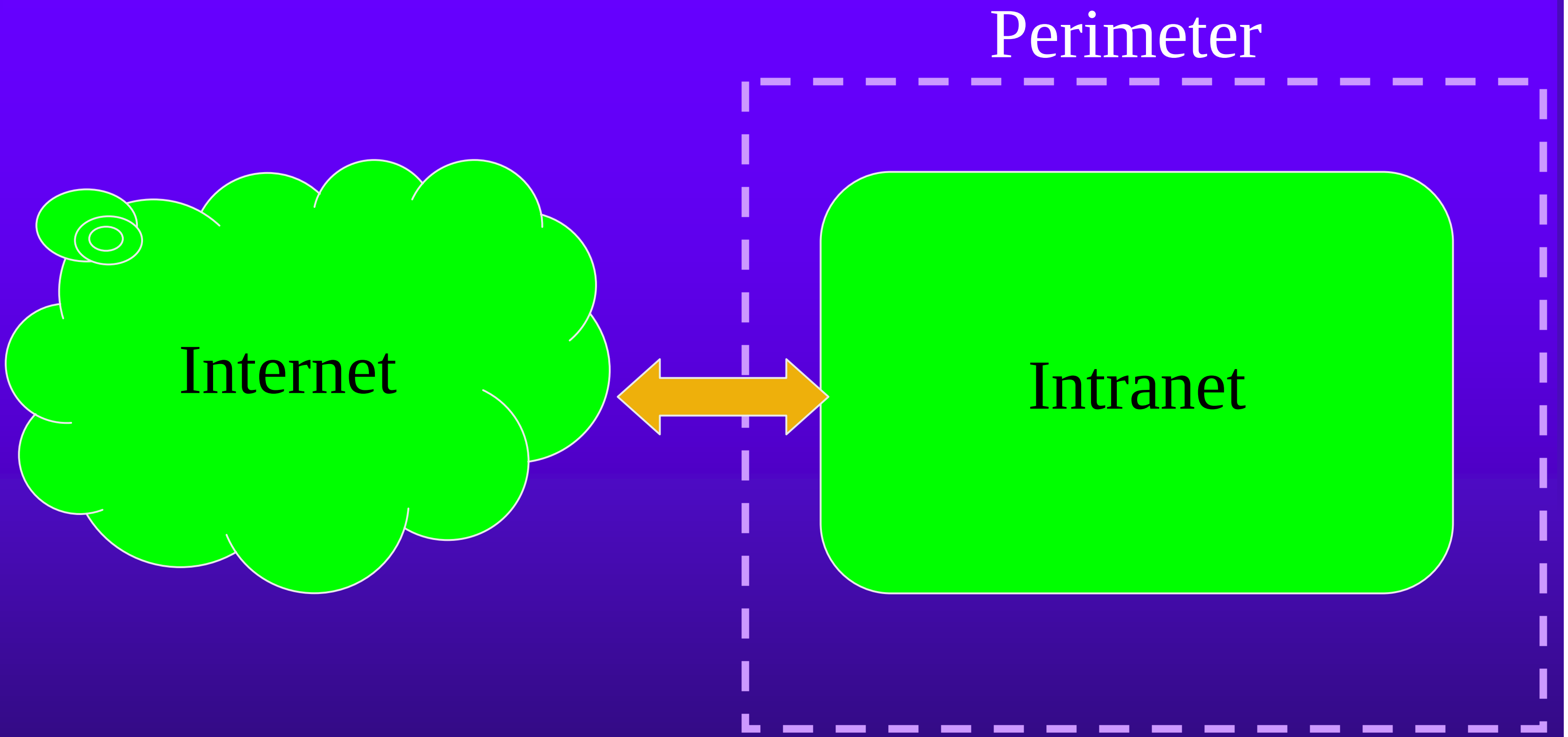


Network Security Approaches

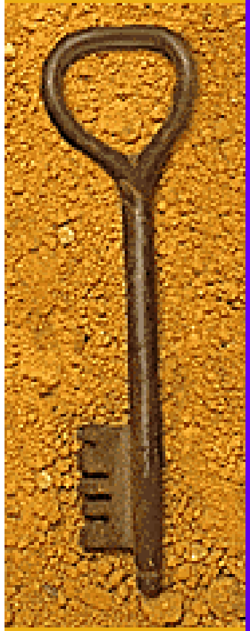
- Secure Networked Computer
- Secure Network Protocols
- Perimeter Security

Perimeter Defense

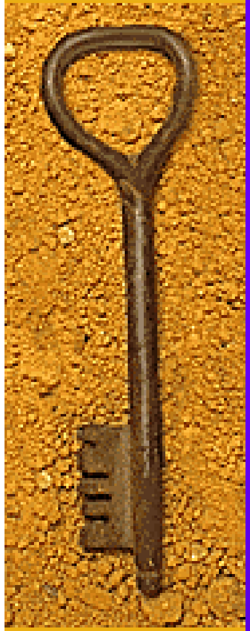




Perimeter Defense Strategy

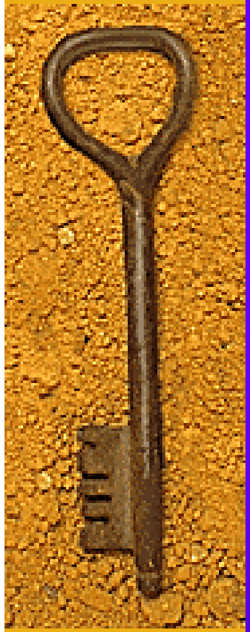


- Divide networks into *zones* of varying trust
  - Simplest division: intranet (trusted) and Internet (untrusted)
- Put security measures on boundaries between zones
  - E.g. connection to ISP



Perimeter Defense Advantages

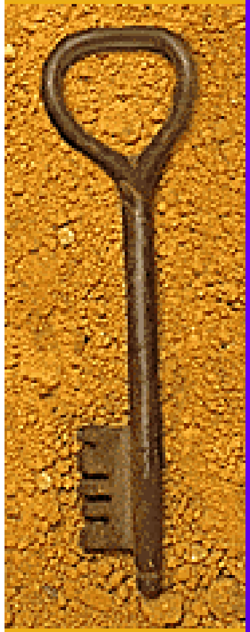
- ♦ Scale
  - – Can configure one computer to be secure, but how about 1,000?
- ♦ Threat model
  - – Most threats come from less trusted zones
- ♦ Convenience
  - – Can use less secure protocols and software inside perimeter
  - – Don’t bother users with security protections unless they talk to the outside



Major Perimeter Defense Technologies

- Firewalls
- Intrusion Detection System (IDS)
- Intrusion Prevention System (IPS)
- Anti-Virus Gateway
- Virtual Privation Network

……

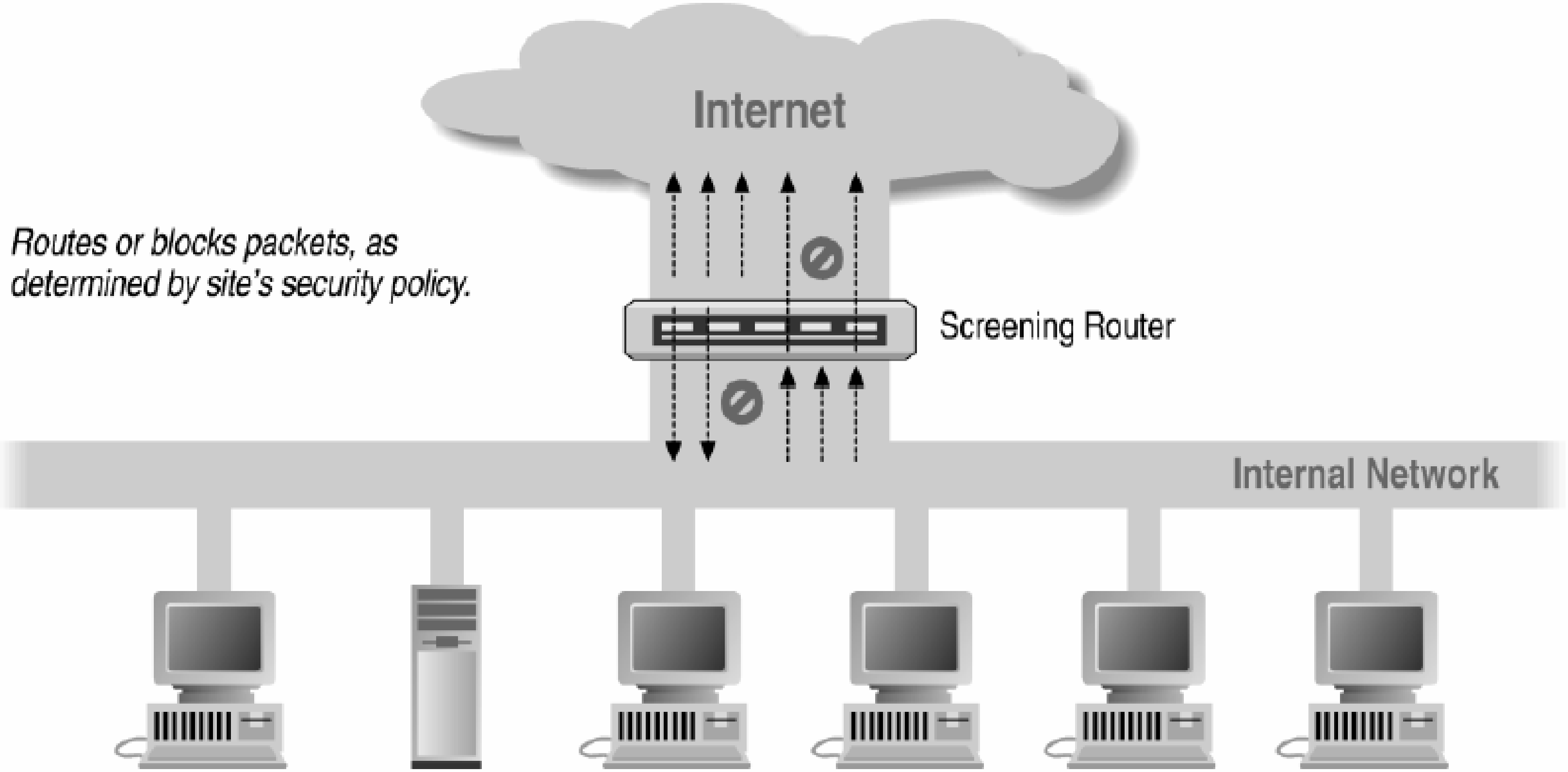


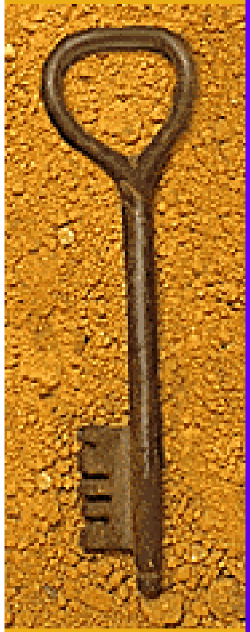
Topics

- Background of Perimeter Security
- Firewalls
  - Basic Firewall Concepts
  - Packet filter (stateless)
  - Stateful firewall
  - Application-layer gateway
- Problems with Firewalls
- Real Firewalls

Firewalls

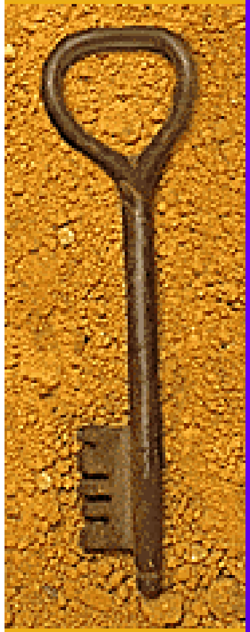
- Filter traffic going across perimeter boundary
- Various levels of sophistication (from IP to App.)


Internet
Routes or blocks packets, as determined by site's security policy.
Screening Router
Internal Network



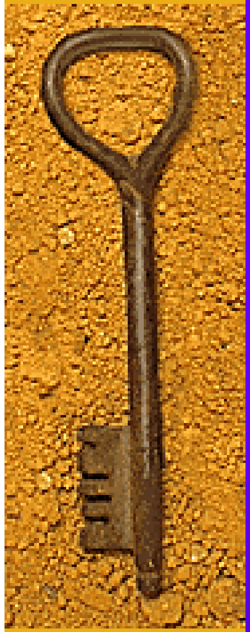
Why firewalls?

- ♦ Need to exchange information
  - – Education, business, recreation, social and political
- ♦ Bugs, everywhere, can not be eliminated
  - – All programs have bugs, Larger ones have more bugs!
  - – Network protocols contain;
    - • Design weaknesses (IP, TCP, SSH, CRC)
    - • Implementation flaws (SMTP, DNS, SSL, NTP, FTP, ...)
  - – Careful (defensive) programming & protocol design is **hard**
- ♦ Defense in depth



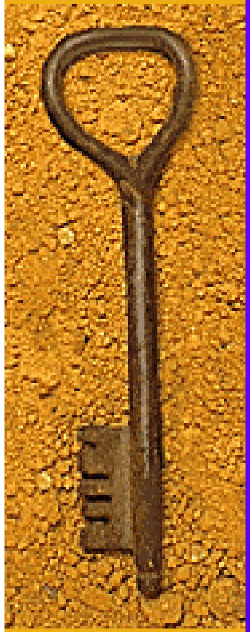
Topics

- Background of Perimeter Security
- Firewalls
  - Basic Firewall Concepts
  - Packet filter (stateless)
  - Stateful firewall
  - Application-layer gateway
- Problems with Firewalls
- Real Firewalls



Packet Filter

- ♦ Filter IP packets based on their headers
- ♦ Fields may include:
  - – IP source address, destination address
  - – Protocol Header (TCP, UDP, ICMP, etc)
  - – TCP or UDP source & destination ports
  - – TCP Flags (SYN, ACK, FIN, RST, PSH, etc)
  - – ICMP message type
- ♦ Stateless & fast
  - – Implementation is based on lookup of header bits/bytes and decisions



Example Rules

allow proto=TCP AND port=80 **(HTTP)**

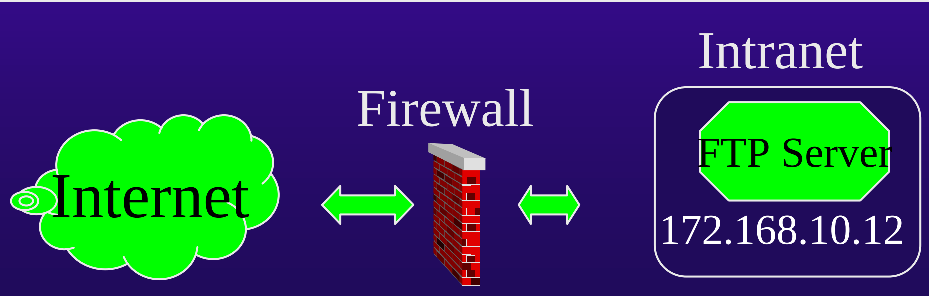
deny proto=UDP AND port=1434 **(SQL)**

allow proto=TCP AND port=21 AND sourceIP=adminConsole **(FTP)**

Example Rules: FTP Packet Filter

The following filtering rules allow a user to FTP from any IP address to the FTP server at 172.168.10.12

```
interface Ethernet 0
 access-list 100 in     ! Apply the first rule to inbound traffic
 access-list 101 out    ! Apply the second rule to outbound traffic
 ! Allows packets from any client to the FTP control and data ports
access-list 100 permit tcp any gt 1023 host 172.168.10.12 eq 21
access-list 100 permit tcp any gt 1023 host 172.168.10.12 eq 20
 ! Allows the FTP server to send packets back to any IP address with
TCP ports > 1023
access-list 101 permit tcp host 172.168.10.12 eq 21 any gt 1023
access-list 101 permit tcp host 172.168.10.12 eq 20 any gt 1023
```



Intranet

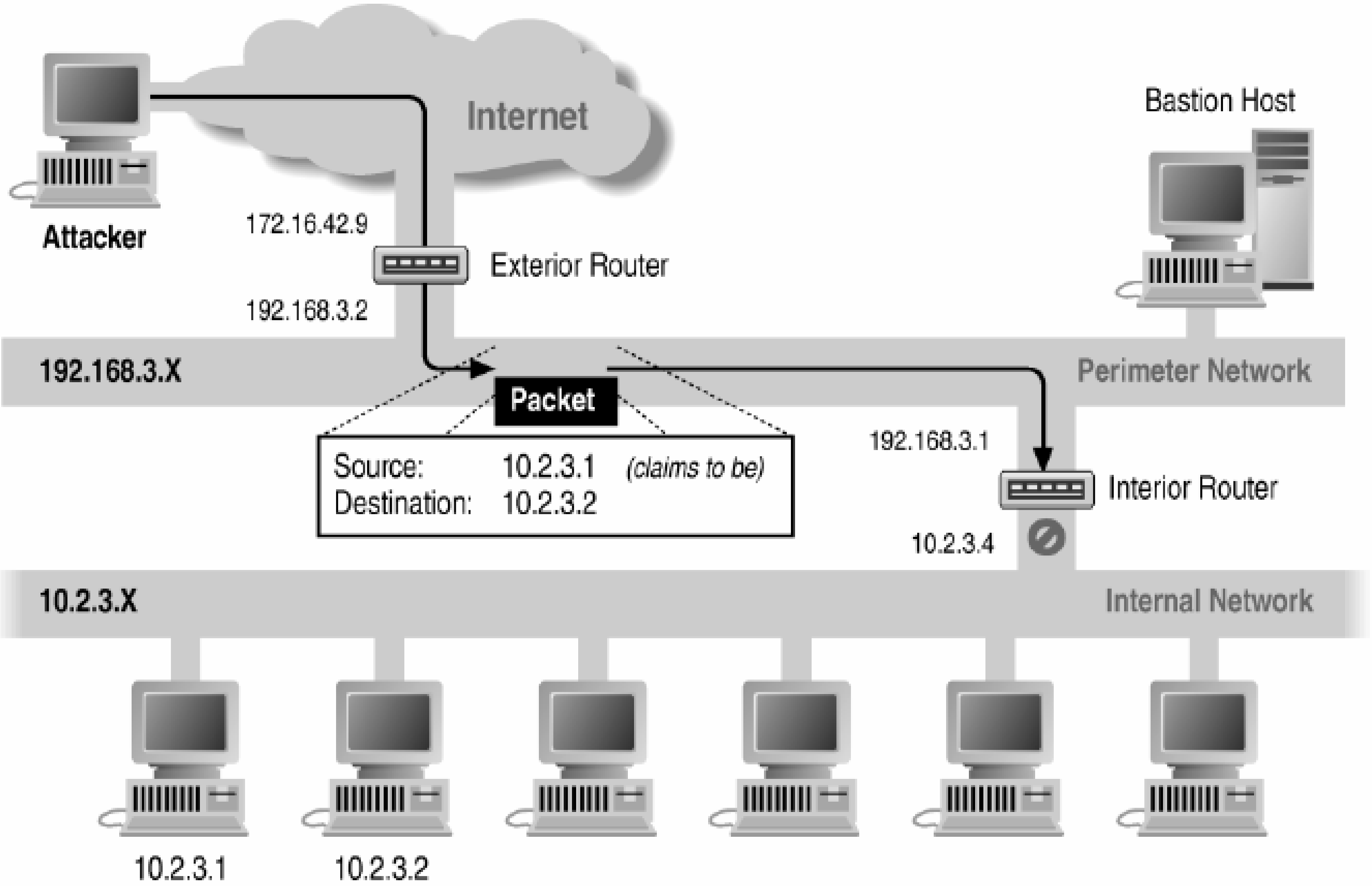
Firewall

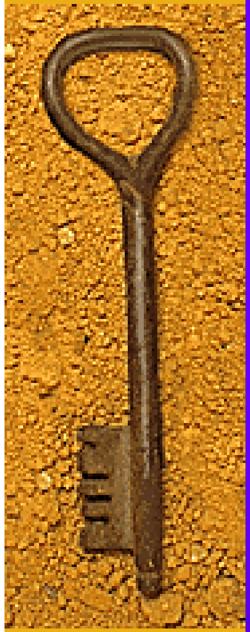
Internet

FTP Server

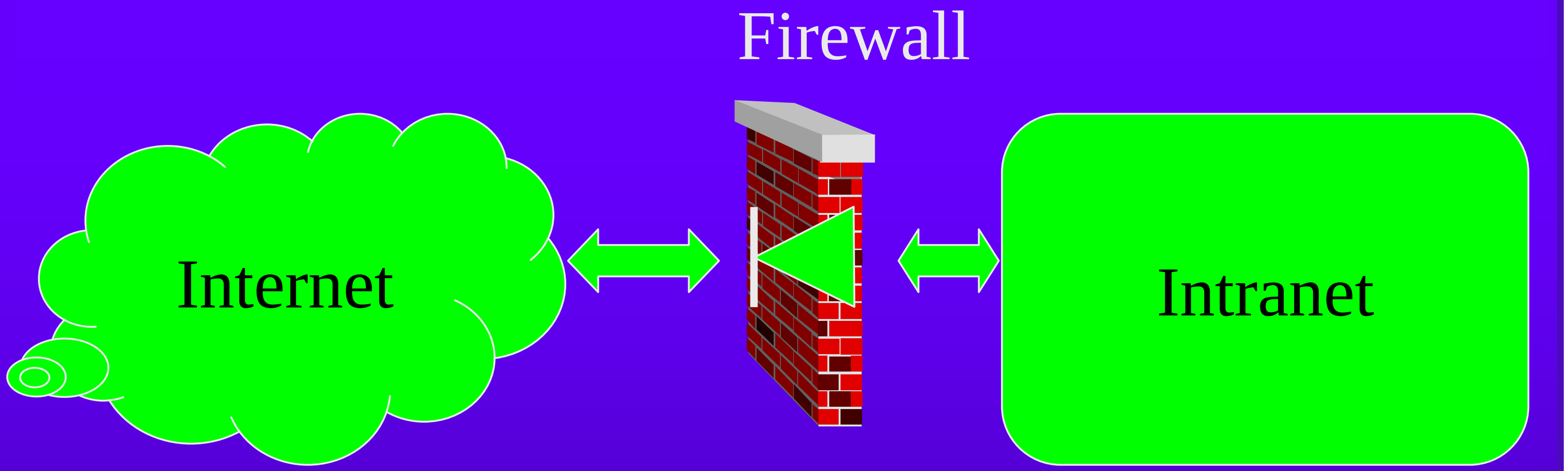
172.168.10.12

Example: Address Forgery

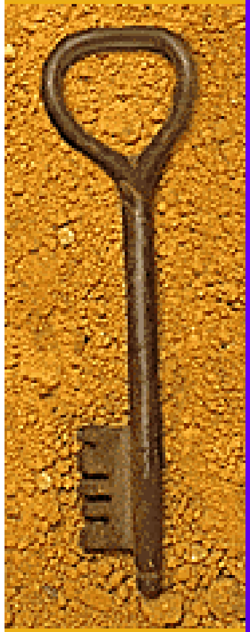




Example Policy



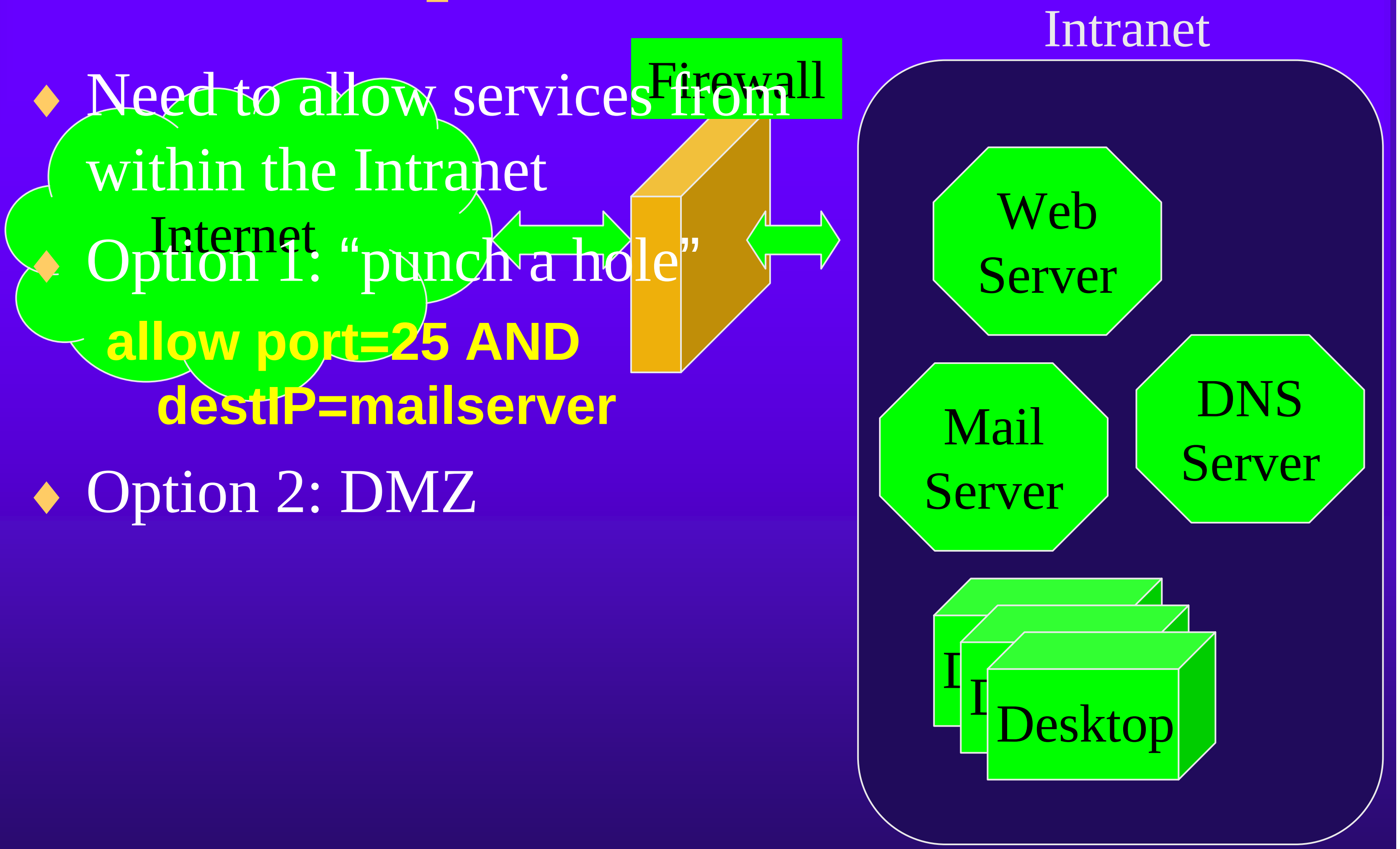
- Outbound traffic only
  - **allow proto=TCP AND (sourceIP=inside OR ACK=true)**

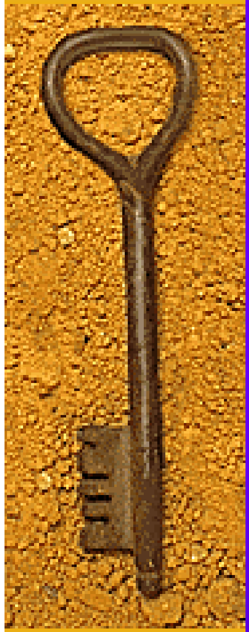


More complicated network

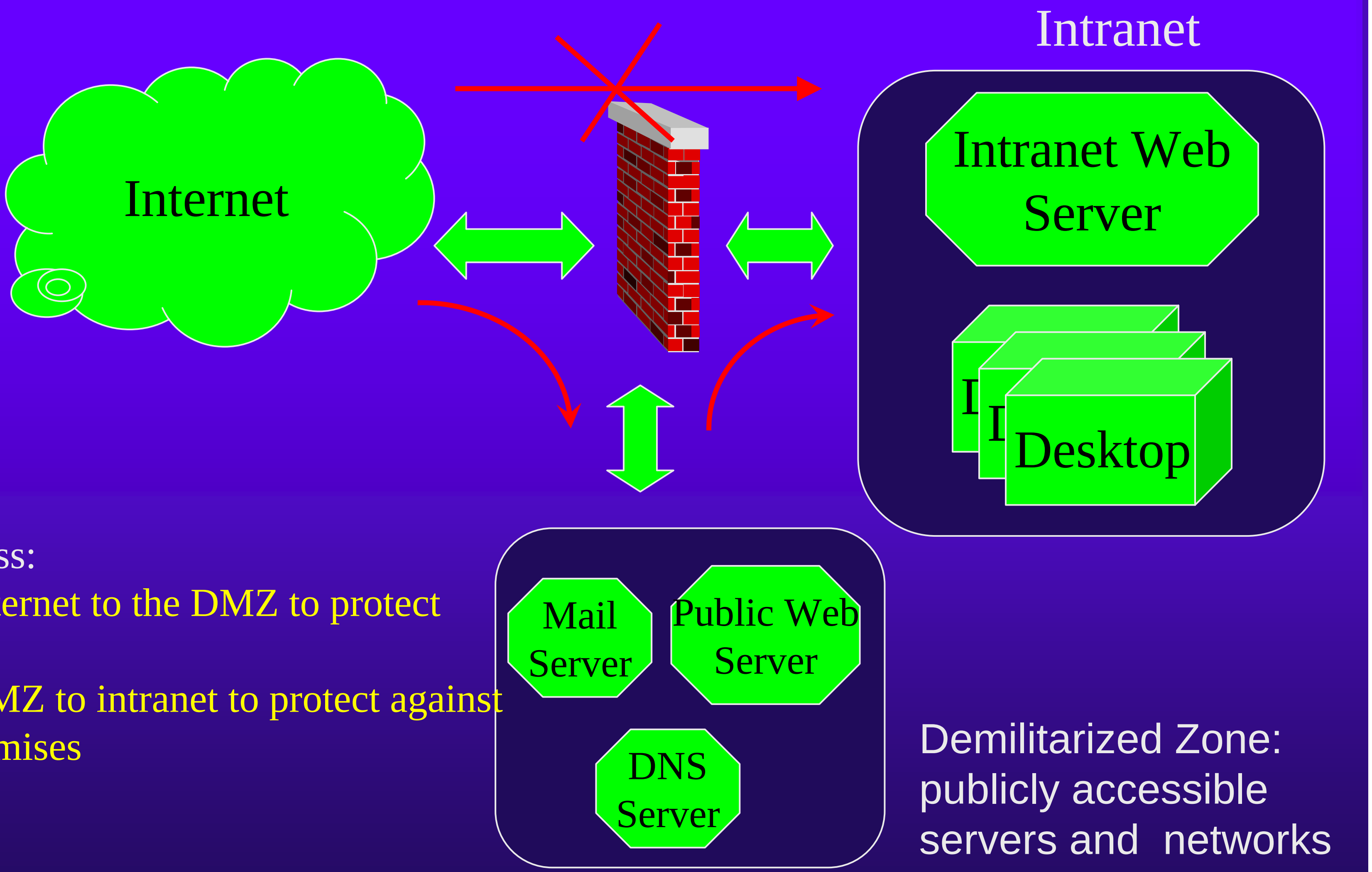
- Need to allow services from within the Intranet
- Option 1: “punch a hole”

  allow port=25 AND destIP=mailserver
- Option 2: DMZ





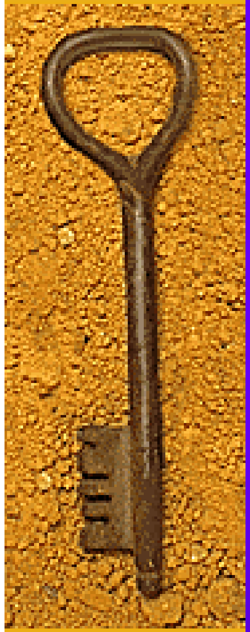
Demilitarized Zone



Restrict access:

- from Internet to the DMZ to protect servers
- from DMZ to intranet to protect against compromises

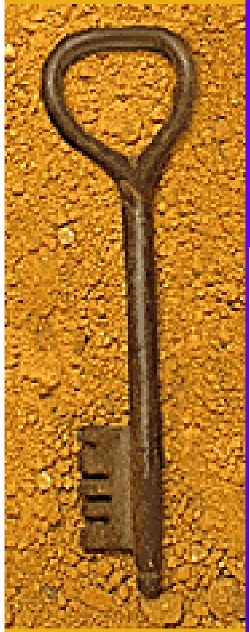
Demilitarized Zone: publicly accessible servers and networks

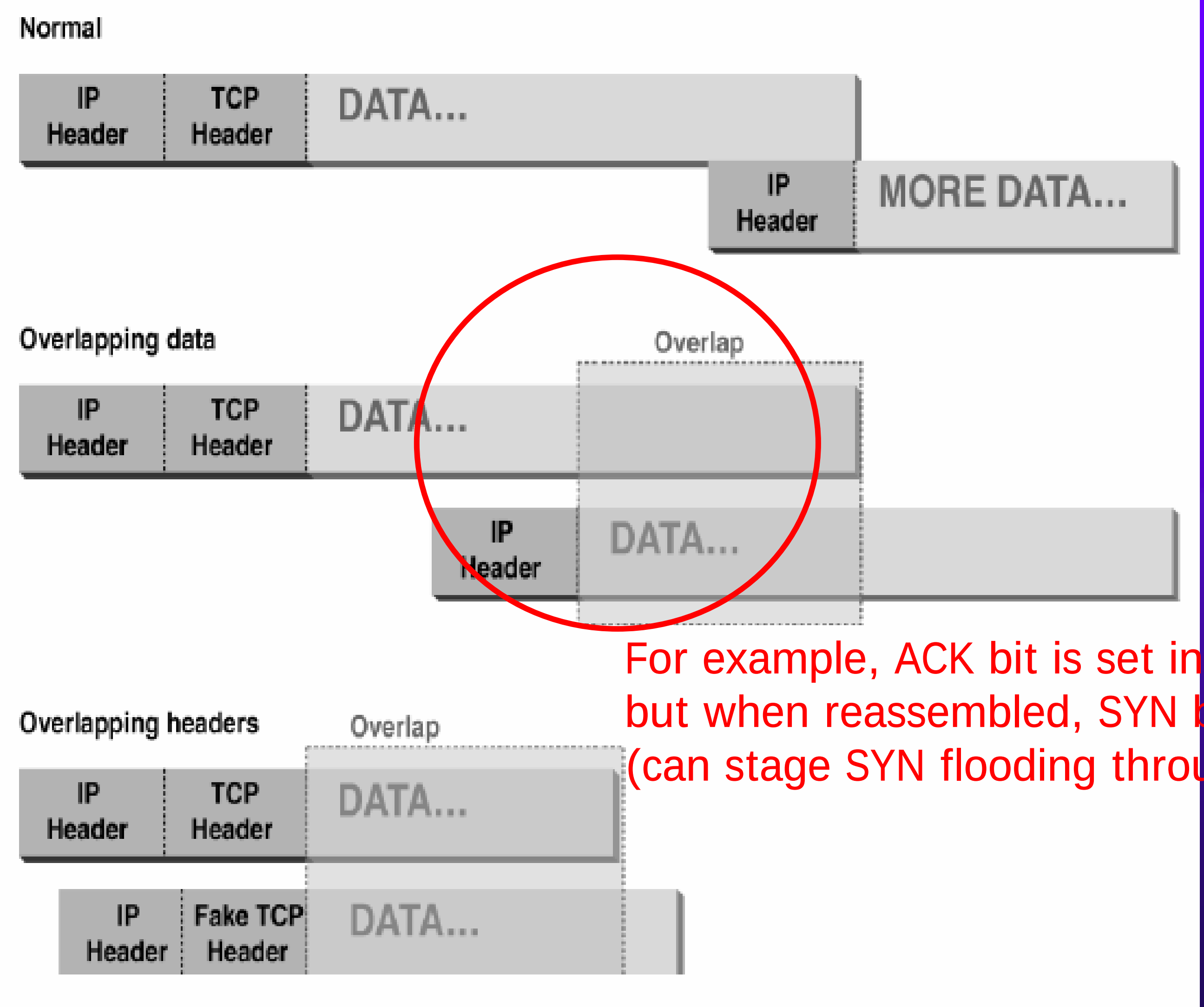


Packet Filter Limitation

- No connection semantics
  - Actions only on individual packets
- No application semantics
  - IP address/Port Number based only
- Packet fragmentation
  - IP allows packets to be split into several fragments

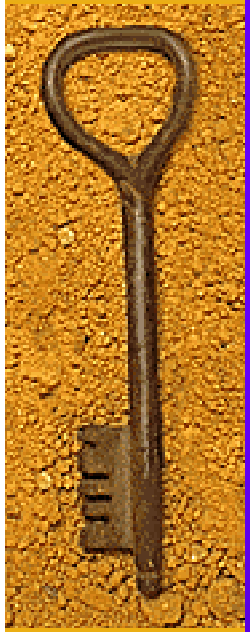
Abnormal Fragmentation





For example, ACK bit is set in both fragments, but when reassembled, SYN bit is set (can stage SYN flooding through firewall)

Fragmentation



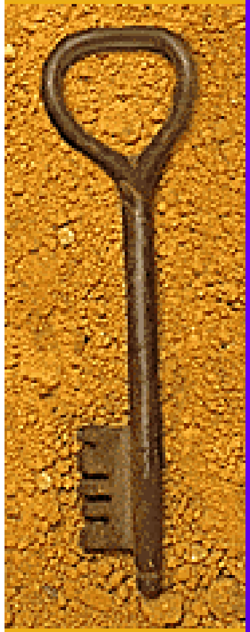
<table>
<tr><td colspan="4">Data Link Layer Header</td></tr>
<tr><td>Ver/IHL</td><td>Type of Service</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="2">Identifier</td><td>Flags</td><td><b>Fragment Offset</b></td></tr>
<tr><td>Time To Live</td><td>Protocol</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="4">Source Address</td></tr>
<tr><td colspan="4">Destination Address</td></tr>
<tr><td colspan="4">Options + Padding</td></tr>
<tr><td colspan="2">Source Port</td><td colspan="2">Destination Port</td></tr>
<tr><td colspan="4">Sequence Number</td></tr>
<tr><td colspan="4">Acknowledgement Number</td></tr>
<tr><td>Offset/Reserved</td><td>U <b>A</b> P R S F</td><td colspan="2">Window</td></tr>
<tr><td colspan="2">Checksum</td><td colspan="2">Urgent Pointer</td></tr>
<tr><td colspan="4">Options + Padding</td></tr>
<tr><td colspan="4">Data</td></tr>
<tr><td colspan="4">Data Link Layer Trailer</td></tr>
</table>

IP Datagram

IP Header

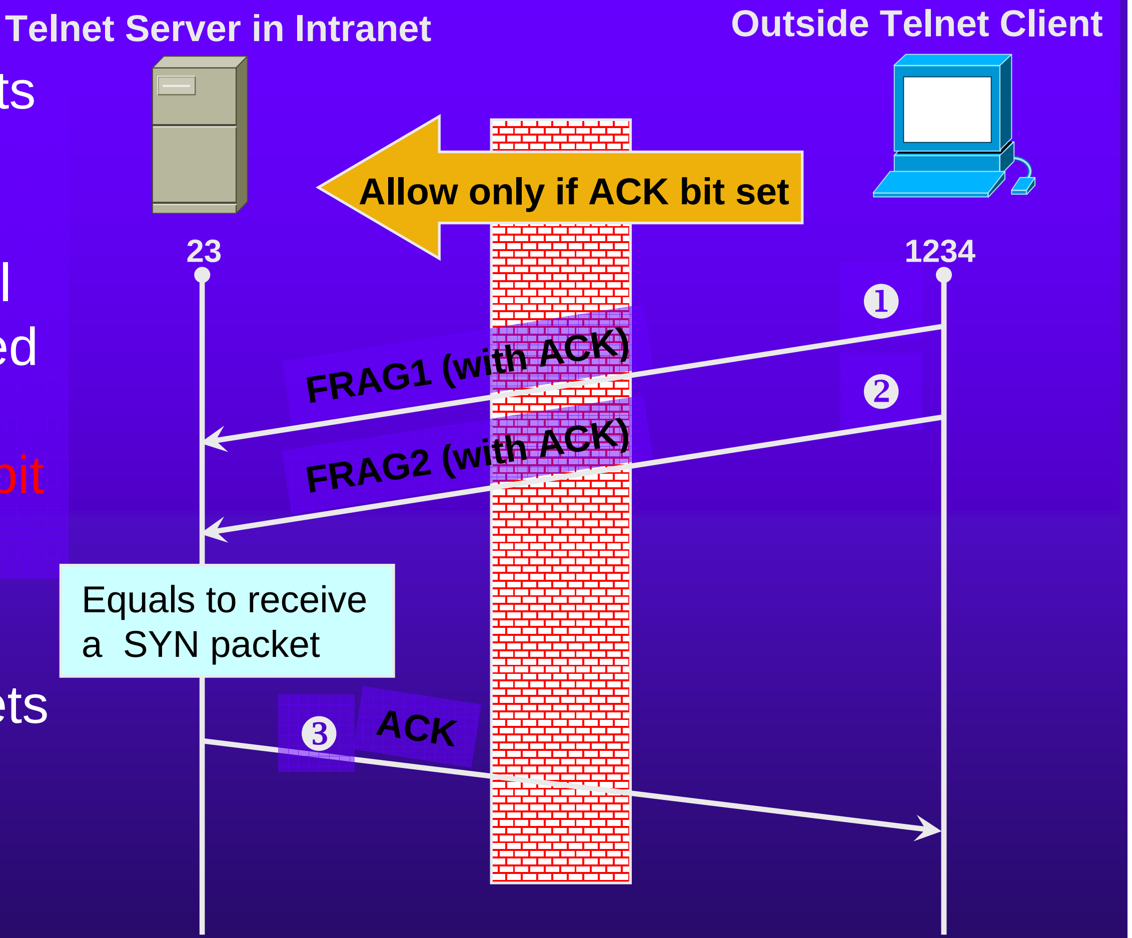
TCP Header

Fragmentation Attack



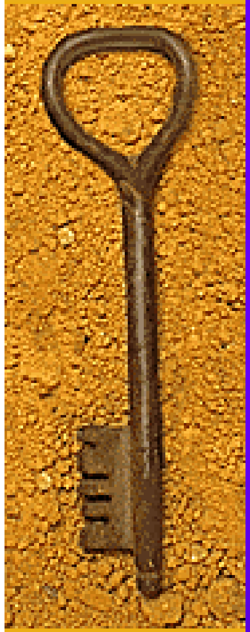
➊,➋ Send 2 fragments with the ACK bit set; fragment offsets are chosen so that the full datagram reassembled by server forms a packet with the SYN bit set

➌ All following packets will have the ACK bit set

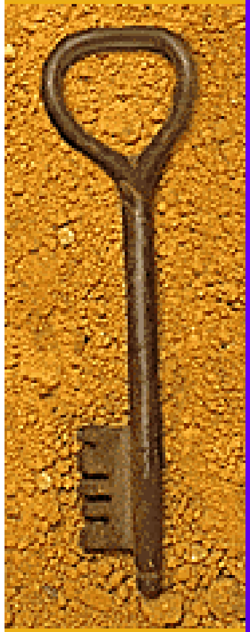

Telnet Server in Intranet
Outside Telnet Client
Allow only if ACK bit set
23
1234
①
②
FRAG1 (with ACK)
FRAG2 (with ACK)
Equals to receive a SYN packet
③
ACK

SYN Flooding attack!

More Fragmentation Attacks

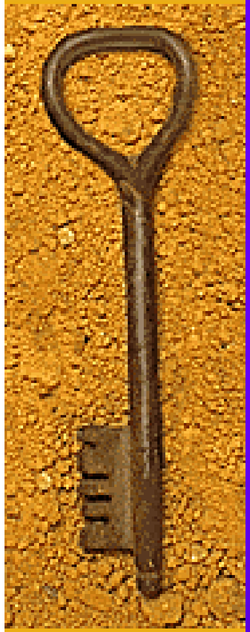


- Split ICMP message into two fragments, the assembled message is too large
  - Buffer overflow, OS crash
- Fragment a URL or FTP "put" command
  - Firewall needs to understand application-specific commands to catch this



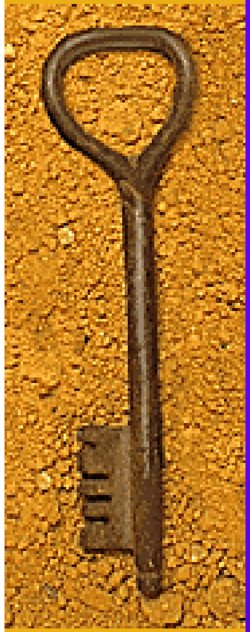
Higher-level analysis

- Packet filters cannot:
  - Forbid a particular URL
  - Detect email viruses
  - Block (malicious) ActiveX plugins
- Alternate approaches:
  - Stateful firewall: reconstruct connections
  - Application-level proxy: transform connections



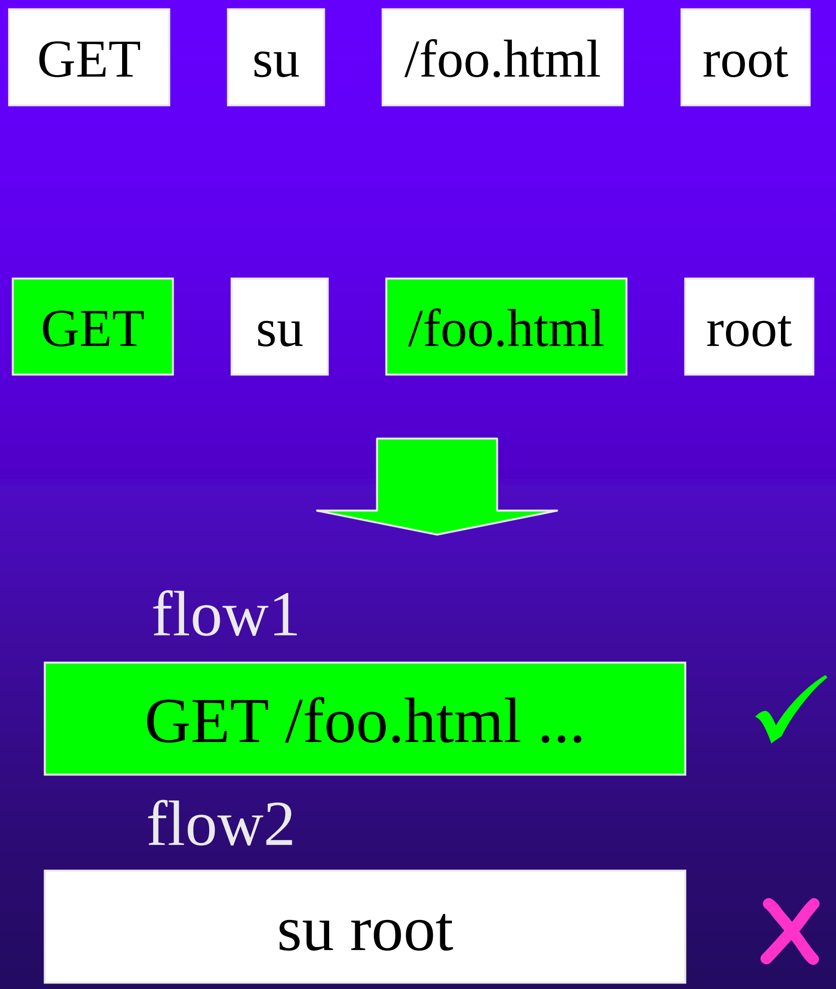
Topics

- ♦ Background of Perimeter Security
- ♦ Firewalls
  - – Basic Firewall Concepts
  - – Packet filter (stateless)
  - – Stateful firewall
  - – Application-layer gateway
- ♦ Problems with Firewalls
- ♦ Real Firewalls



Stateful Firewall

- Reconstruct connection state
- Make decisions based on *flows*, not on *packets*
- Some application protocol parsing may also be done



GET | su | /foo.html | root

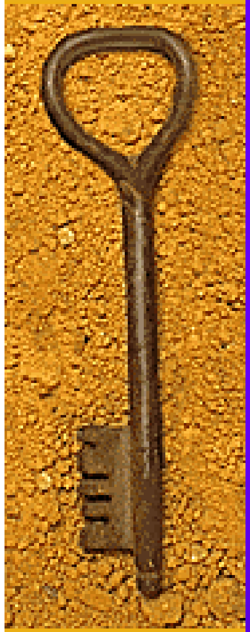
GET | su | /foo.html | root

flow1

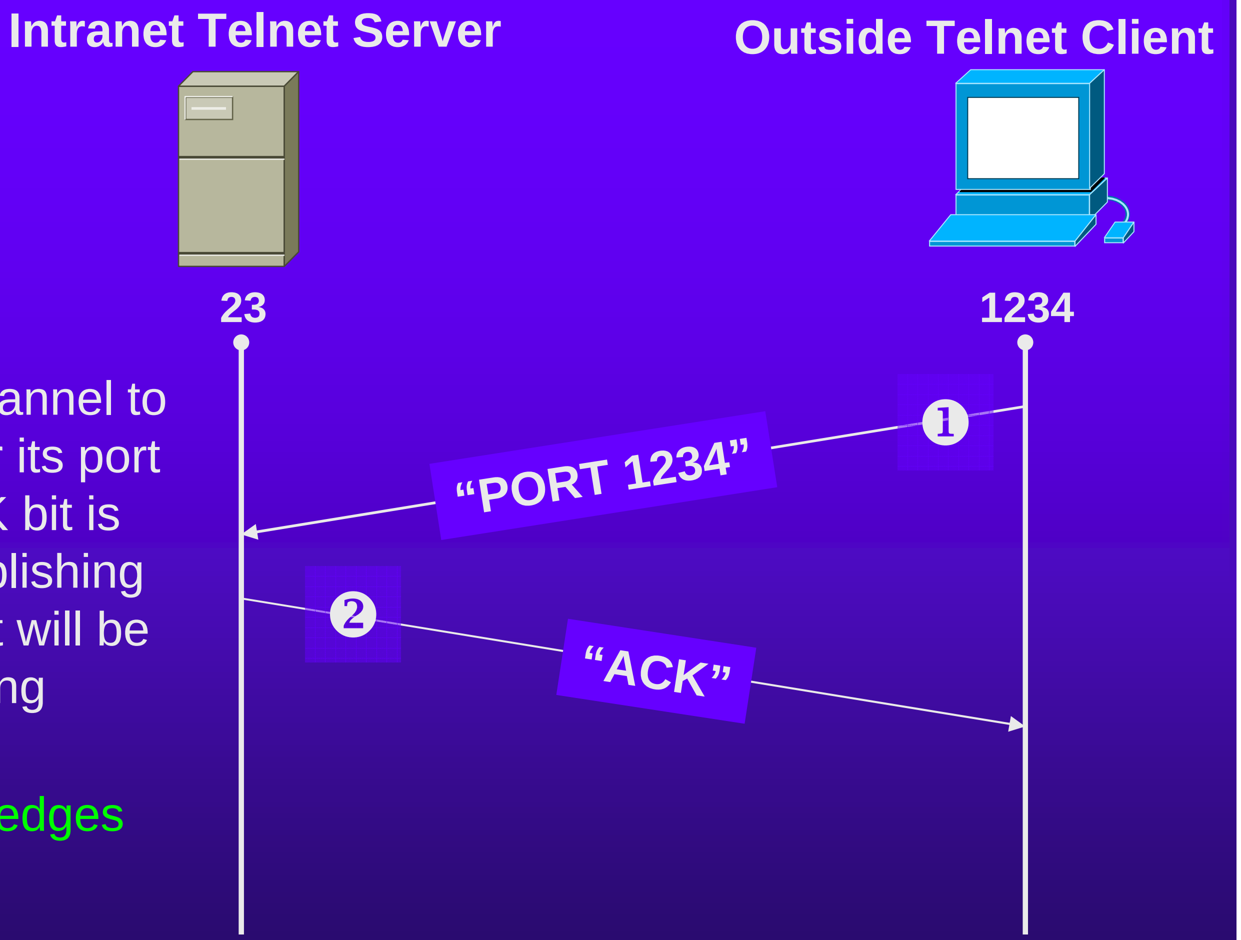
GET /foo.html ... ✓

flow2

su root ✗



Examples: Telnet

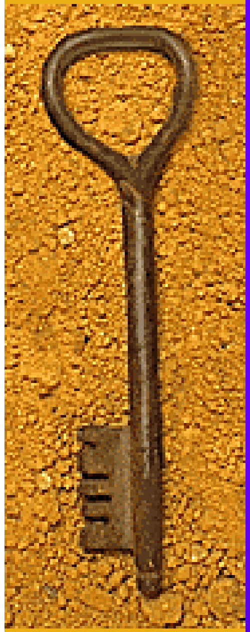


➊ Client opens channel to server; tells server its port number. The ACK bit is not set while establishing the connection but will be set on the remaining packets

➋ Server acknowledges

Stateful filtering can use this pattern to prevent SYN-Flooding Attack

Examples: FTP

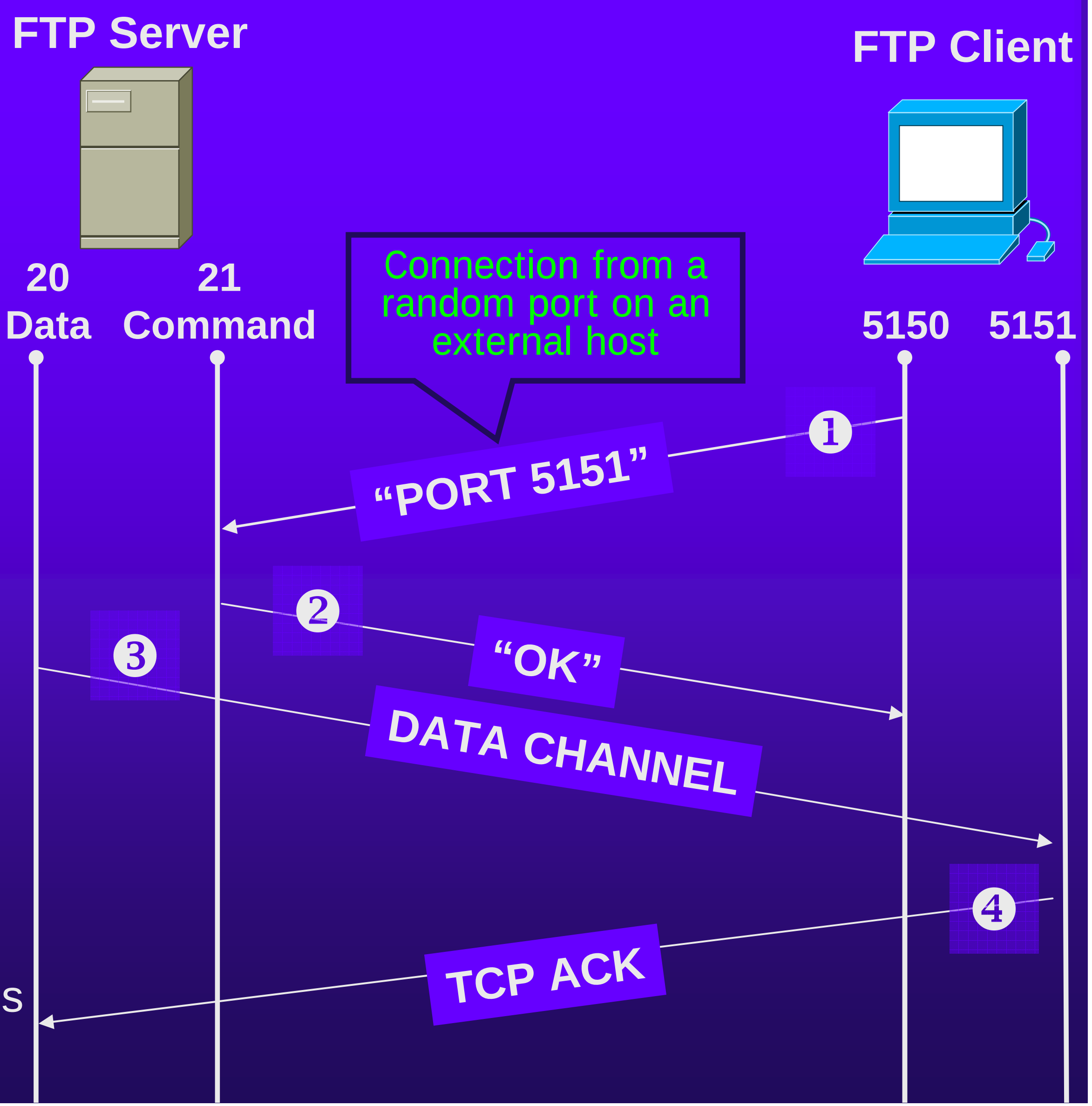


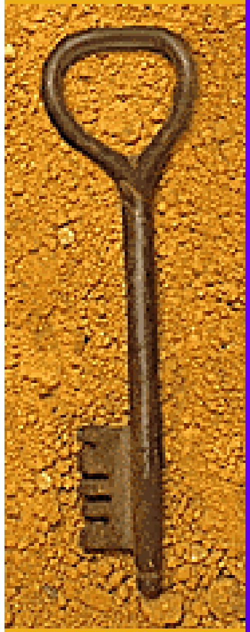
❶ Client opens command channel to server; tells server second port number

❷ Server acknowledges

❸ Server opens data channel to client's second port

❹ Client acknowledges



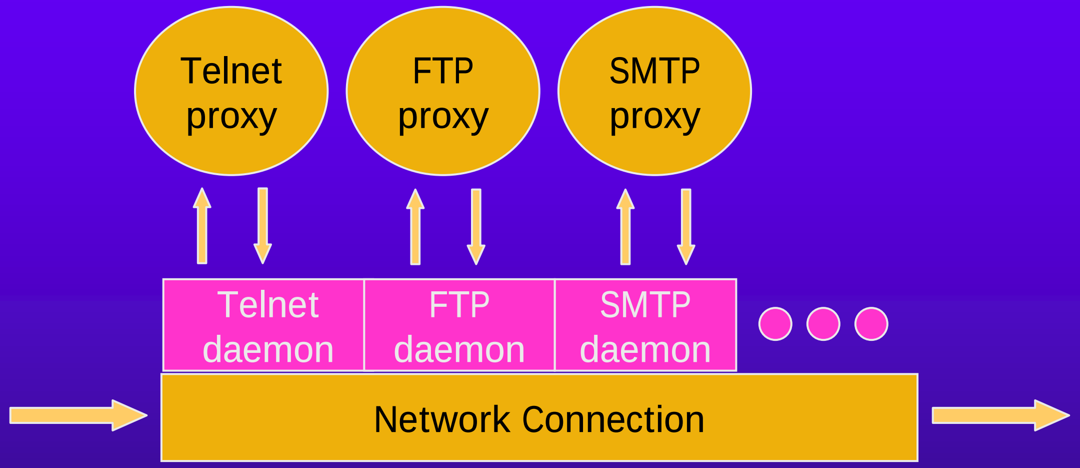


Topics

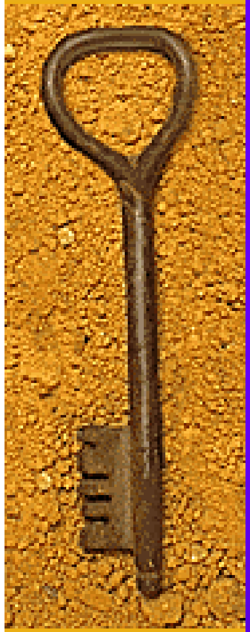
- Background of Perimeter Security
- Firewalls
  - Basic Firewall Concepts
  - Packet filter (stateless)
  - Stateful firewall
  - Application-layer gateway
- Problems with Firewalls
- Real Firewalls

Application-Level Proxy

- Process incoming packets at application layer

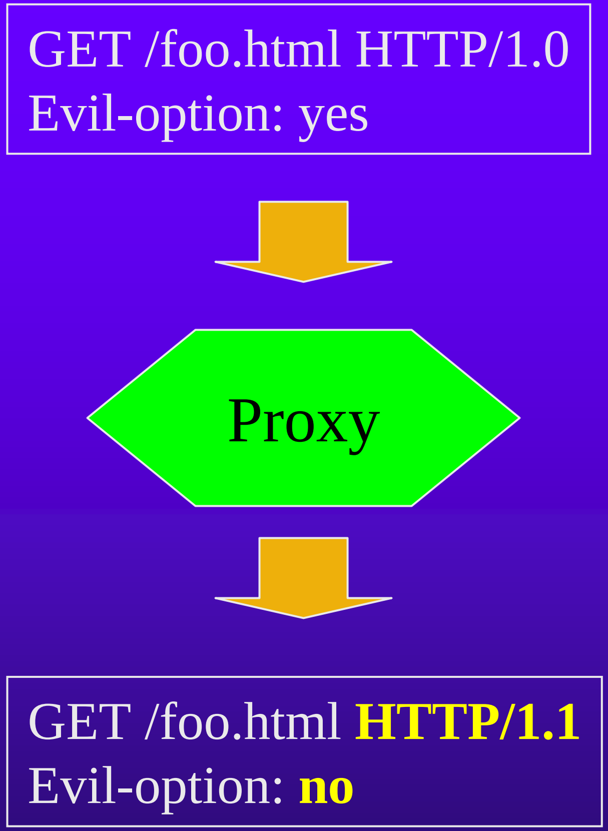


Daemon spawns proxy when communication detected



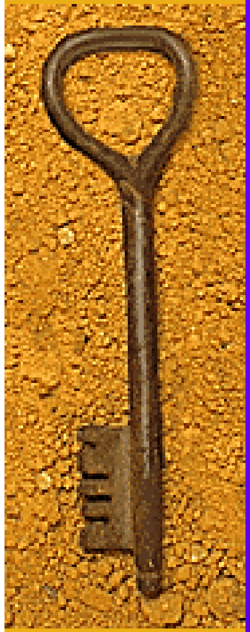
Application-Level Proxy

- Generate transformed message stream
  - Block dangerous messages
  - Normalize protocol semantics



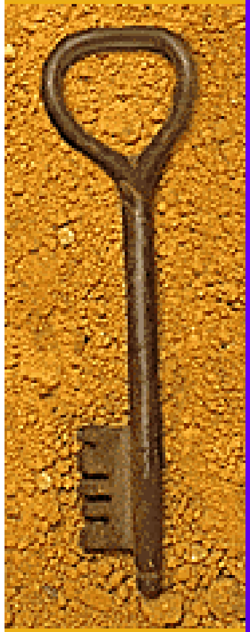
GET /foo.html HTTP/1.0
Evil-option: yes

GET /foo.html **HTTP/1.1**
Evil-option: **no**



Trade-offs

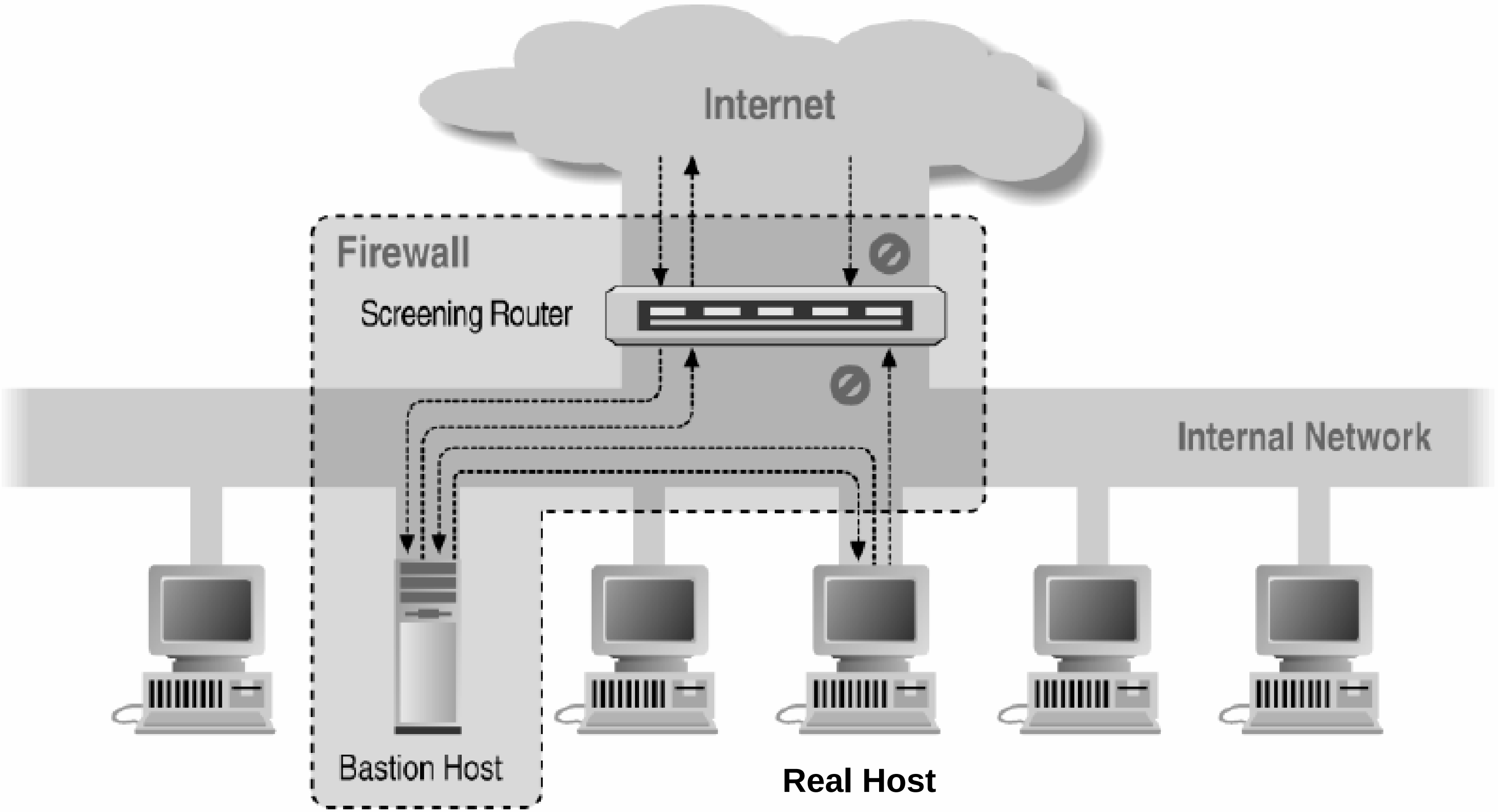
- Pro: Higher precision
- Con: Higher costs
  - Scalability: imaging that it have to keep state for all connections for 1000’s of computers!
  - Latency: proxy adds processing delays
  - Flexibility: proxy needs to understand everything you do with a protocol



Application-level proxies

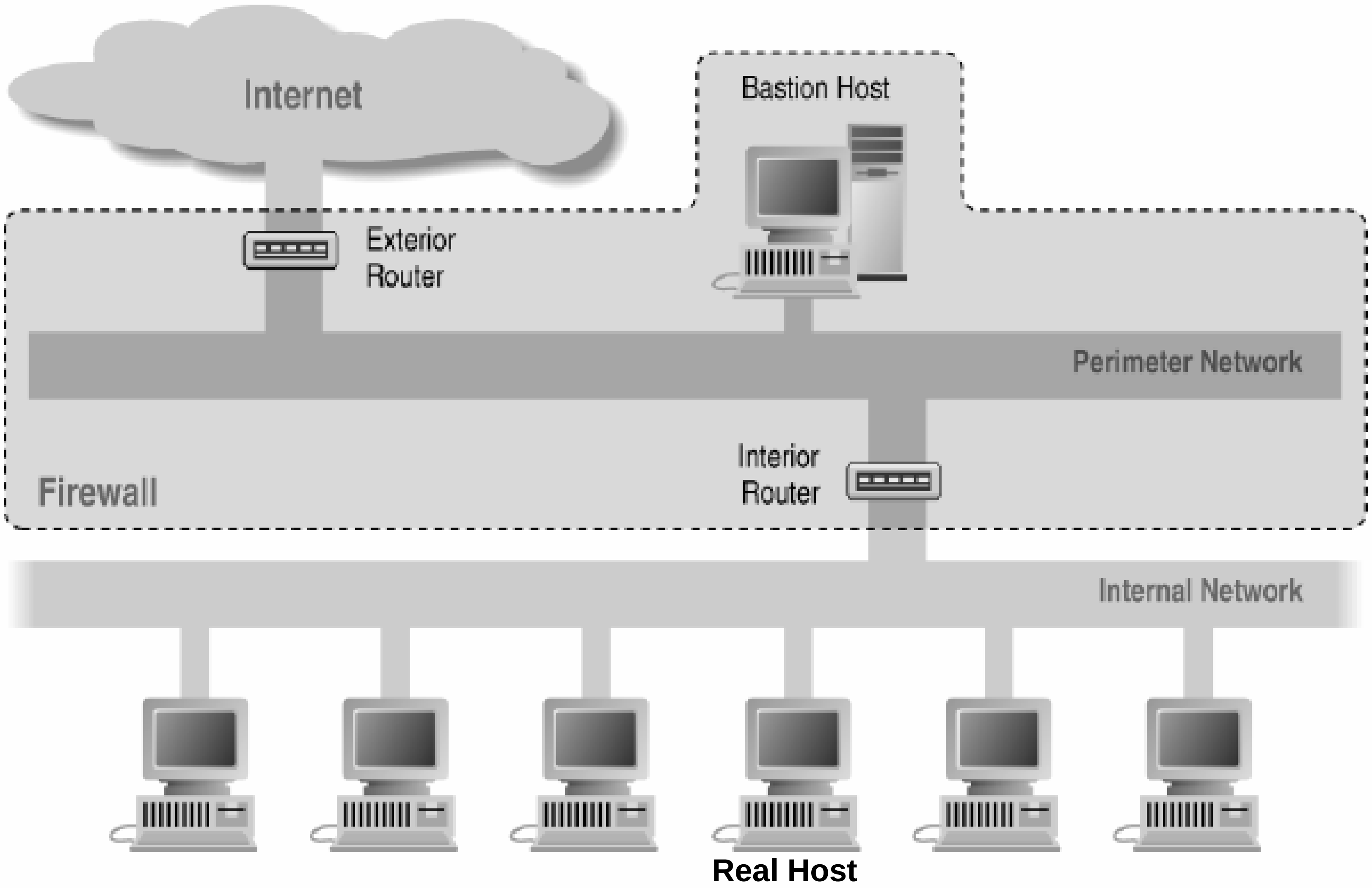
- Enforce policy for specific protocols
  - E.g., Virus scanning for SMTP
    - Need to understand MIME, encoding, Zip archives
- Use “bastion host”
  - Computer running protocol stack
  - Will interact/accepts data from the Internet
    - Install/modify services you want
    - Disable all non-required services; keep it simple
    - Run security audit to establish baseline
    - Be prepared for the system to be compromised
  - Several network locations – see next slides

Screened Host Architecture

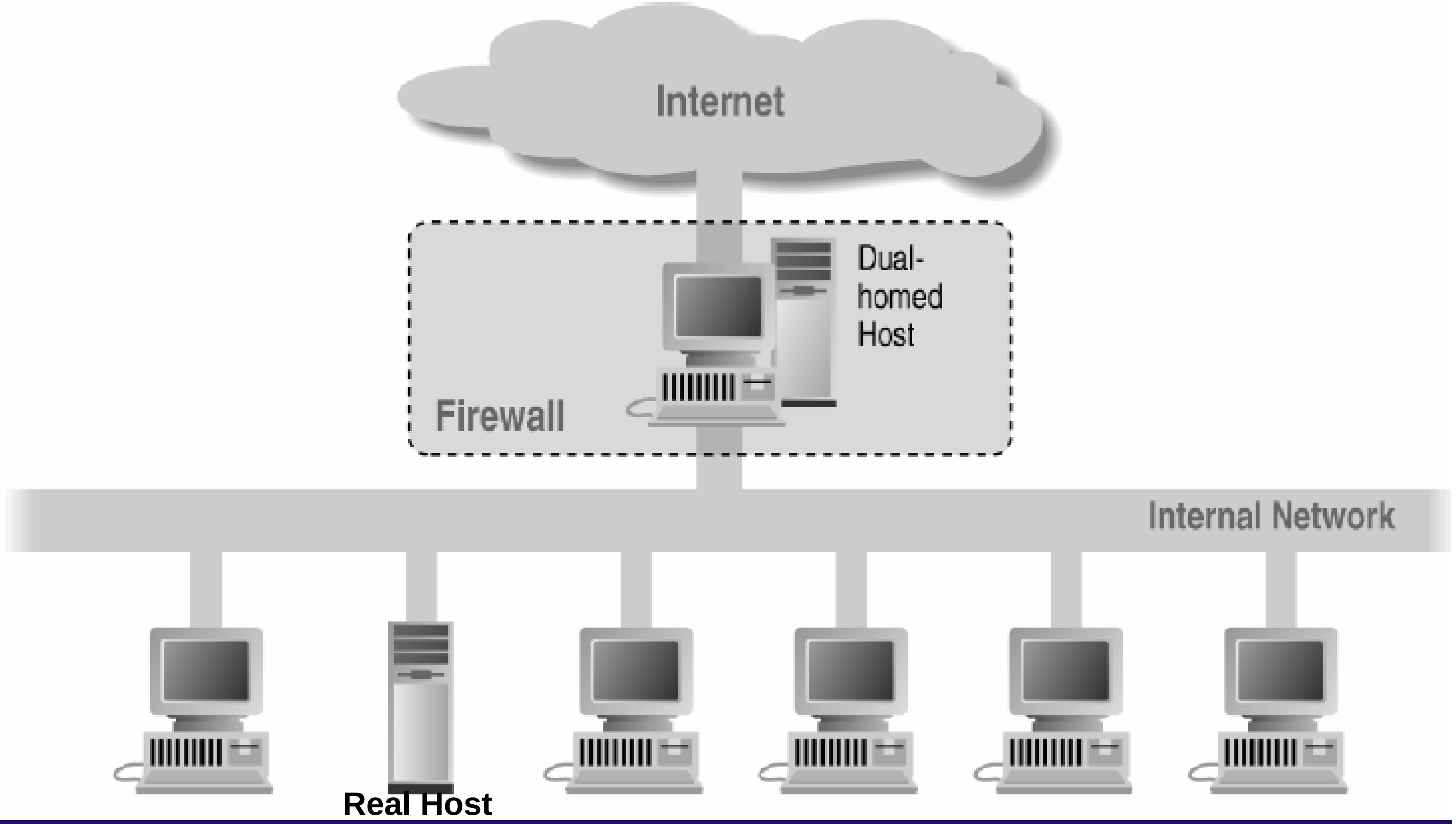


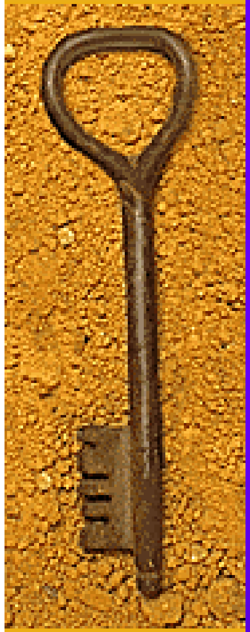
Real Host

Screened Subnet Using Two Routers



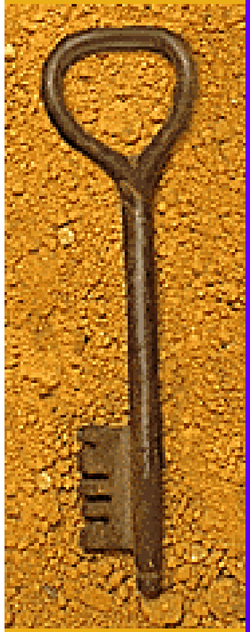
Dual Homed Host Architecture





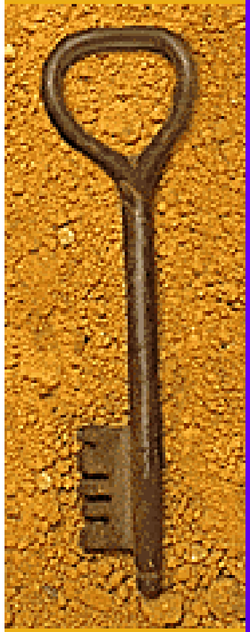
Comparison

	Security	Performance	Modify Client Applications?
Packet Filter	Low	High	No
Session Filter	Medium	Medium	No
App. GW	Hight	Low	Unless transparent, client application must be proxy-aware & configured



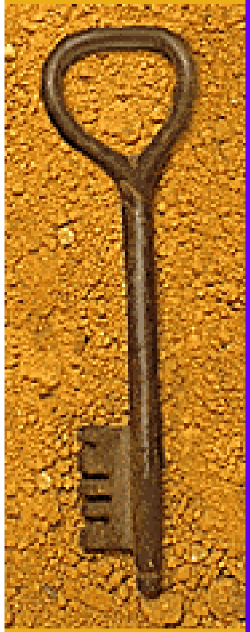
Topics

- Background of Perimeter Security
- Firewalls
  - Basic Firewall Concepts
  - Packet filter (stateless)
  - Stateful firewall
  - Application-layer gateway
- Problems with Firewalls
- Real Firewalls



Problems with Firewalls

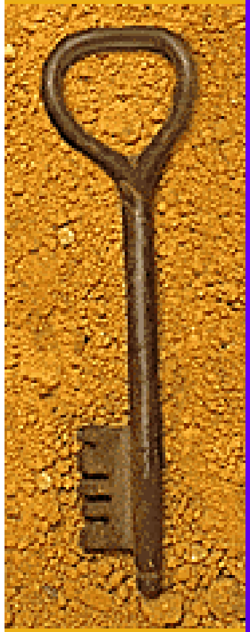
- ♦ Performance
  - – Firewalls may interfere with network use
- ♦ Limitations
  - – They don't solve the real problems
    - • Buggy software； Bad protocols
  - – Generally cannot prevent Denial of Service
  - – Do not prevent insider attacks
- ♦ Administration
  - – Many commercial firewalls permit very complex configurations



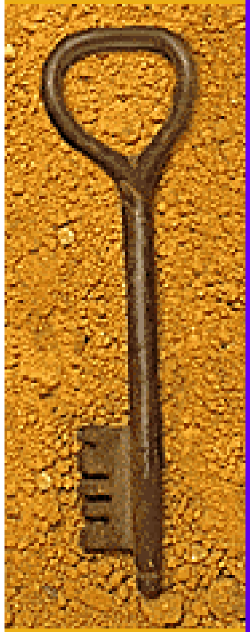
Topics

- Background of Perimeter Security
- Firewalls
  - Basic Firewall Concepts
  - Packet filter (stateless)
  - Stateful firewall
  - Application-layer gateway
- Problems with Firewalls
- Real Firewalls

Turtle Firewall

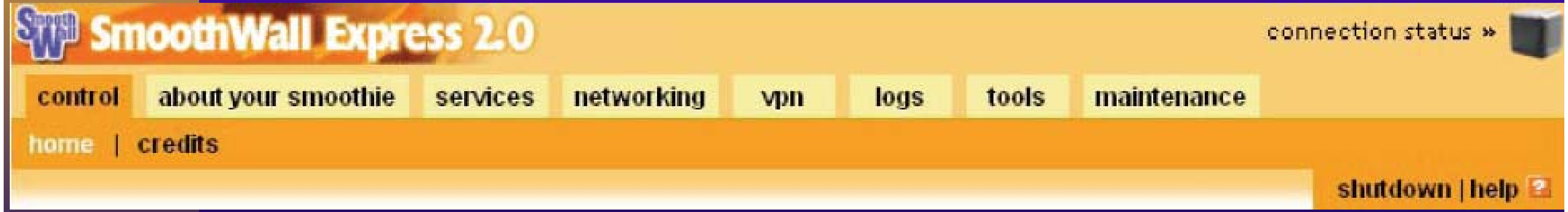


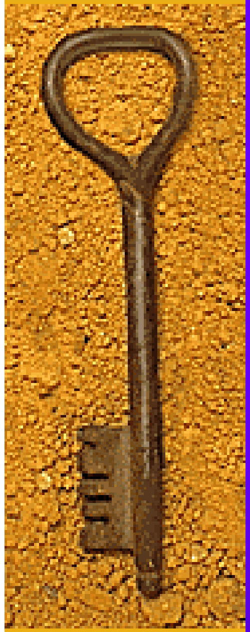
- A software which allows you to realize a Linux firewall in a simply and fast way.
- Based on Kernel 2.4.x and Iptables.
- Policies can be written by a XML file or using the comfortable web interface Webmin.
- Open Source project written using the perl language and realeased under GPL version 2.0



SmoothWall

- SmoothWall Express is an open source firewall distribution based on the GNU/Linux operating system.
- “SmoothWall is configured via a web-based GUI, and requires absolutely no knowledge of Linux to install or use” (scary statement!)
- It integrates with firewall, DHCP, VPN, IDS, Web proxy, SSH, Dynamic DNS.





Sonicwall Pro 300 Firewall

- A firewall device with 3 ports: Internet, DMZ, Intranet.
- You can use one-to-one NAT for systems in Intranet.
- Support VPN. IPSec VPN, compatible with other IPSec-compliant VPN gateways
- 3 DES (168-Bit) Performance: 45 Mbps
- ICSA Certified, Stateful Packet Inspection firewall
- Concurrent connections: 128,000
- Firewall performance: 190 Mbps (bi-directional)