Computer & Information Security (3-721-460-1)

Firewalls

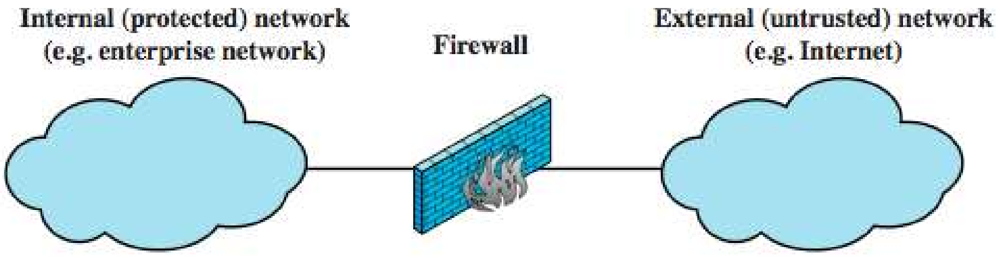
Dept. of Software and Information Systems Engineering, Ben-Gurion University

Prof. Yuval Elovici, Dr. Asaf Shabtai
{elovici, shabtaia}@bgu.ac.il

Spring, 2018

Firewall Goals

- separate between two zones/networks
  - private / public
  - sub-networks
- inspect all traffic from inside to outside and vice versa
  - based on applied rule set
- prevent unwanted/unknown traffic from entering the network; only authorized traffic
- immune to penetration


Internal (protected) network
(e.g. enterprise network)
Firewall
External (untrusted) network
(e.g. Internet)

(a) General model

- service control
- direction control
- user control (local users)
- behavior control (filter spam)

Firewall Guidelines

- least privilege
- defines a single choke point
- fail-safe (define how will it react in case of failure)
- block all unless allowed
- provides a location for monitoring security events
- convenient platform for some Internet functions such as NAT, usage monitoring, IPSEC VPNs
- avoid connection from outside to the internal network

Firewall Limits

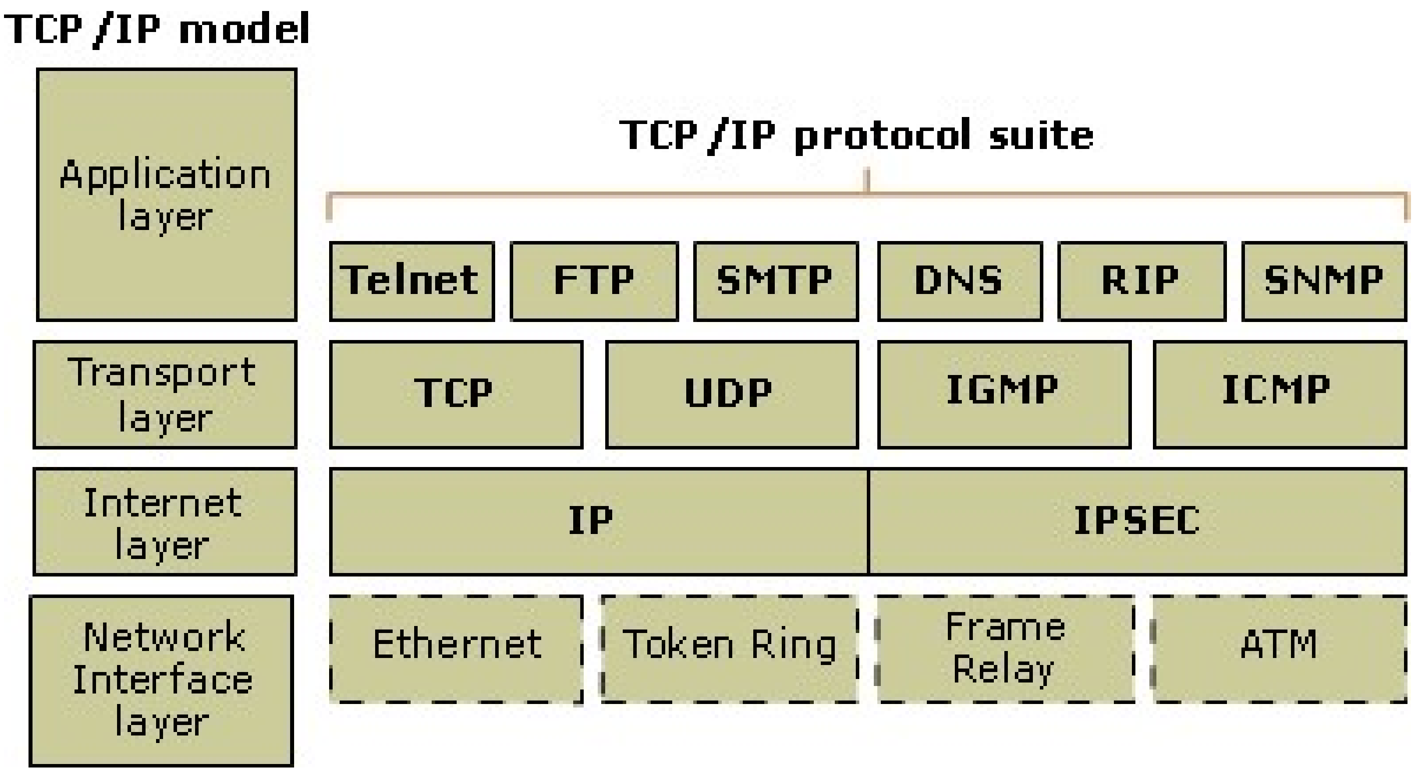
- cannot protect against attacks bypassing firewall (e.g., dial-out capability to an ISP)
- may not protect fully against internal threats
- improperly secured wireless LAN
- laptop, PDA, portable storage device infected outside then used inside

Types of Firewalls

- (Stateless) packet filtering firewall
- Stateful inspection firewall
- Application proxy firewall
- Circuit-level proxy firewall

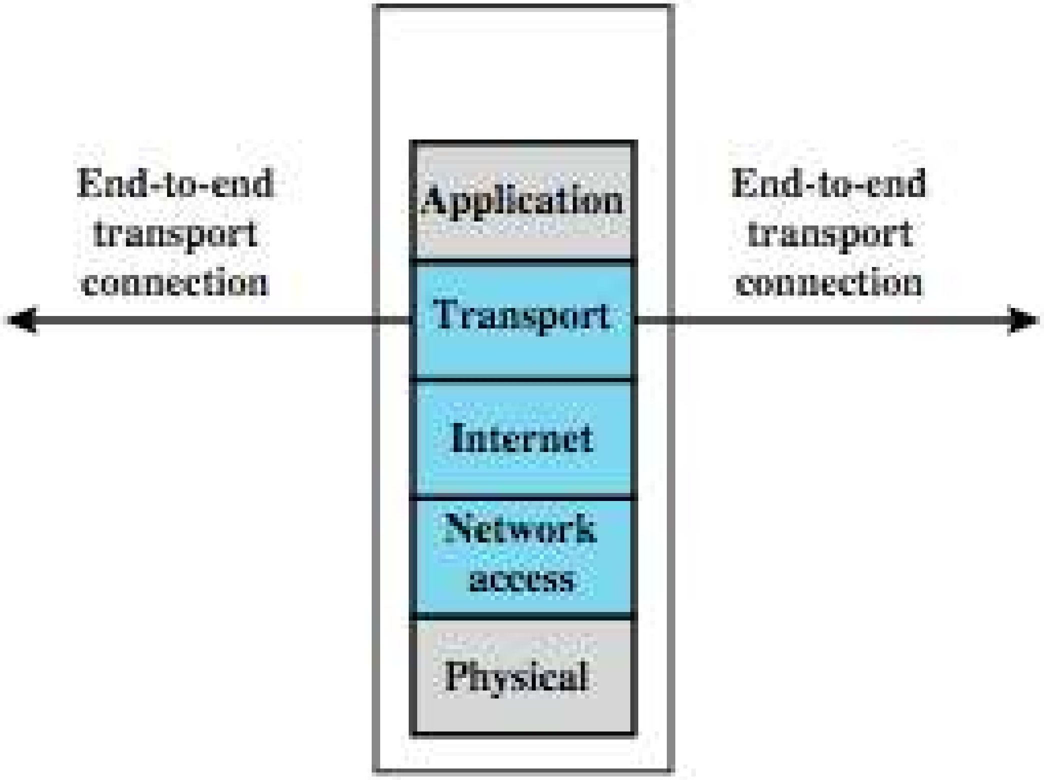
- Different in
  - analyzed info
  - analysis time
  - decision level

Types of Firewalls



TCP /IP model	TCP /IP protocol suite
Application layer	Telnet FTP SMTP DNS RIP SNMP
Transport layer	TCP UDP IGMP ICMP
Internet layer	IP IPSEC
Network Interface layer	Ethernet Token Ring Frame Relay ATM

Packet Filtering Firewall



Packet Filtering Firewall

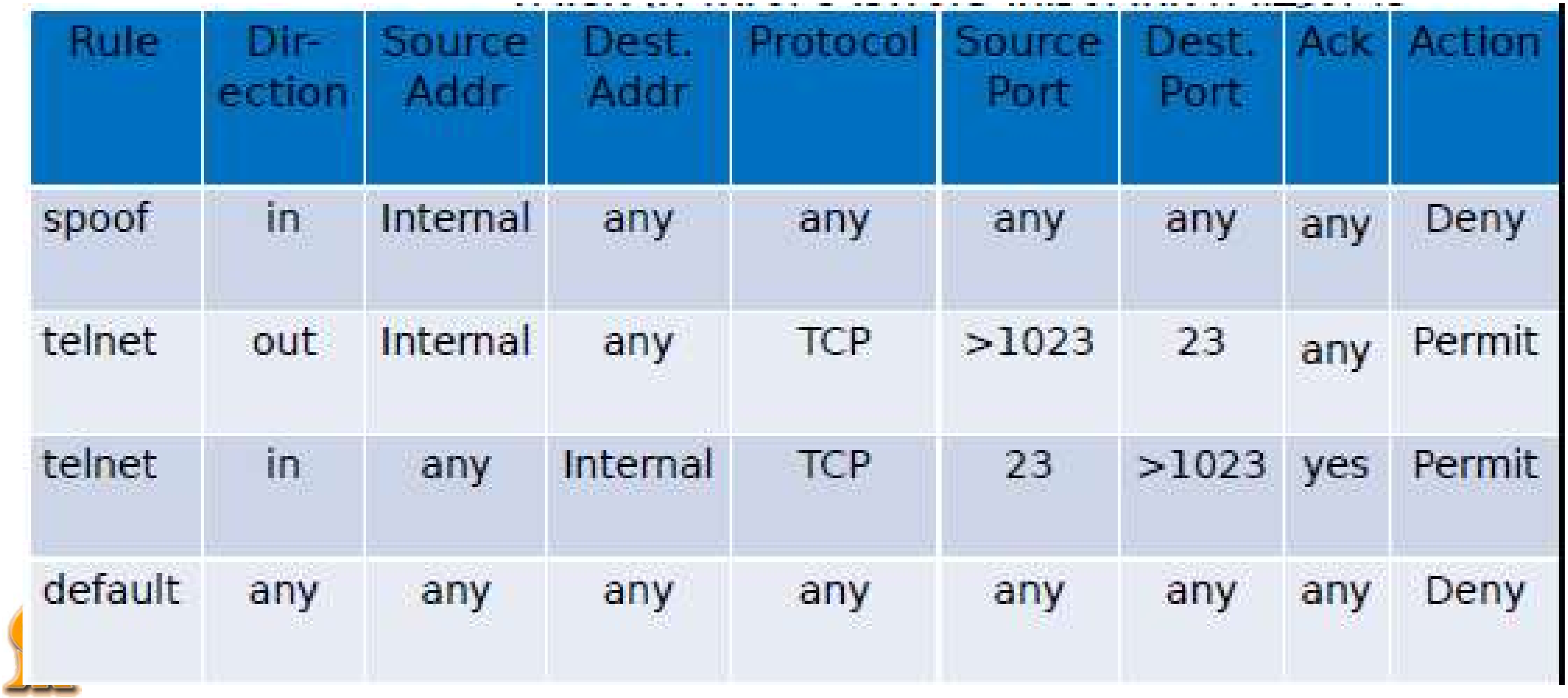
- scans and applies rules to packets in/out of firewall
- based on information in packet header
  - src/dest IP addr & port, IP protocol, interface
- typically a list of rules of matches on fields
  - if match rule says if forward or discard packet
- two default policies:
  - discard - prohibit unless expressly permitted
    - more conservative, controlled, visible to users
  - forward - permit unless expressly prohibited
    - easier to manage/use but less secure

Packet Filtering Firewall

- Rule table is not updated dynamically
- Checks rules one-by-one
- If none of the rules is matched, discard
- Ack bit
  - applies to TCP traffic only
  - first TCP packet ack=0; the rest of the packets in the same session ack=1
  - Therefore, ack=0 means new session attempt
  - use rules on the ack bit to prevent initiating sessions from outside

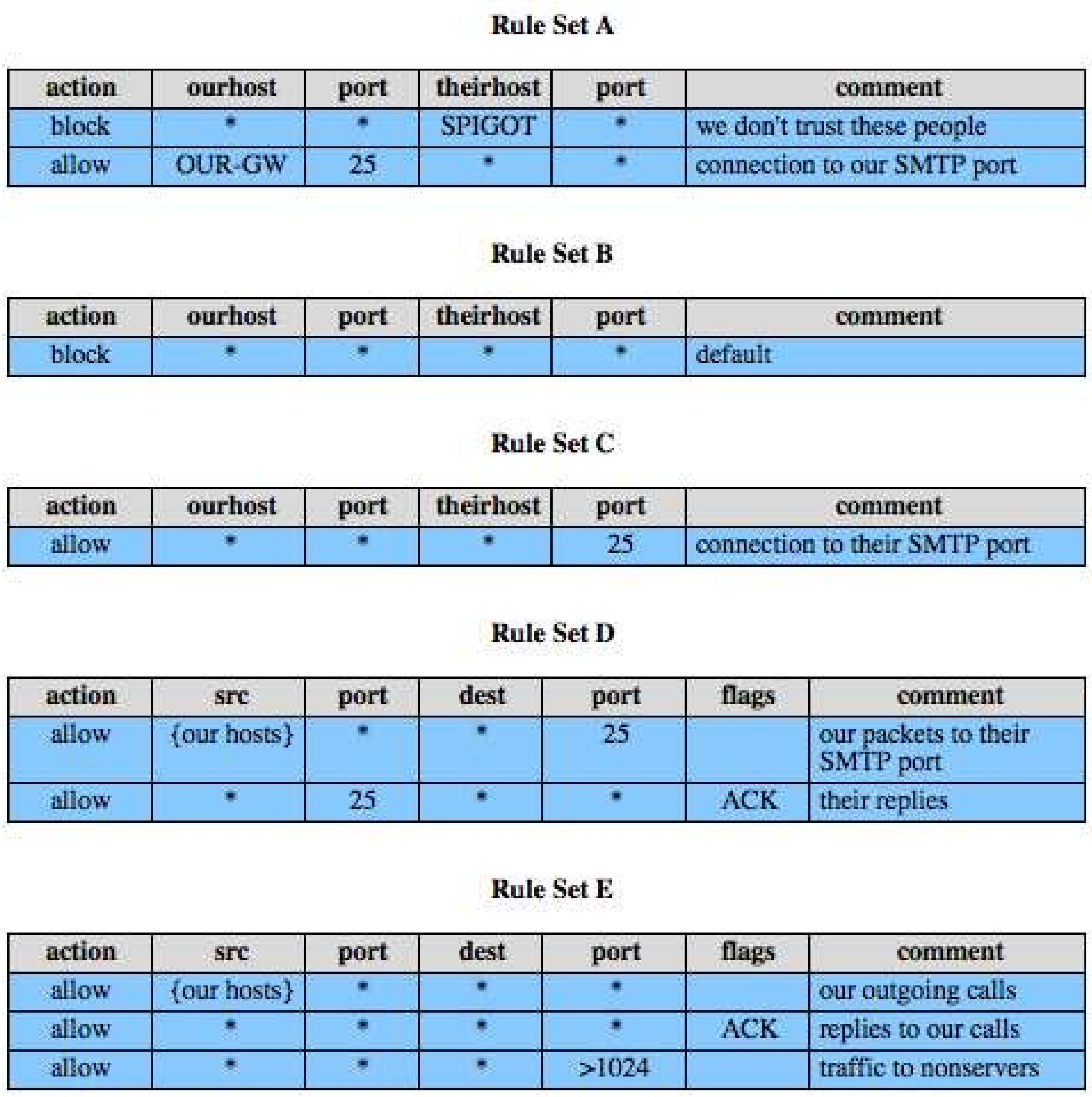
Packet Filter Rules

- Allow telnet from private network to servers in public networks
- Any other traffic is not allowed



Rule	Dir-ection	Source Addr	Dest. Addr	Protocol	Source Port	Dest. Port	Ack	Action
spoof	in	Internal	any	any	any	any	any	Deny
telnet	out	Internal	any	TCP	>1023	23	any	Permit
telnet	in	any	Internal	TCP	23	>1023	yes	Permit
default	any	any	any	any	any	any	any	Deny

Packet Filter Rules



Rule Set A

action	ourhost	port	theirhost	port	comment
block	*	*	SPIGOT	*	we don't trust these people
allow	OUR-GW	25	*	*	connection to our SMTP port

Rule Set B

action	ourhost	port	theirhost	port	comment
block	*	*	*	*	default

Rule Set C

action	ourhost	port	theirhost	port	comment
allow	*	*	*	25	connection to their SMTP port

Rule Set D

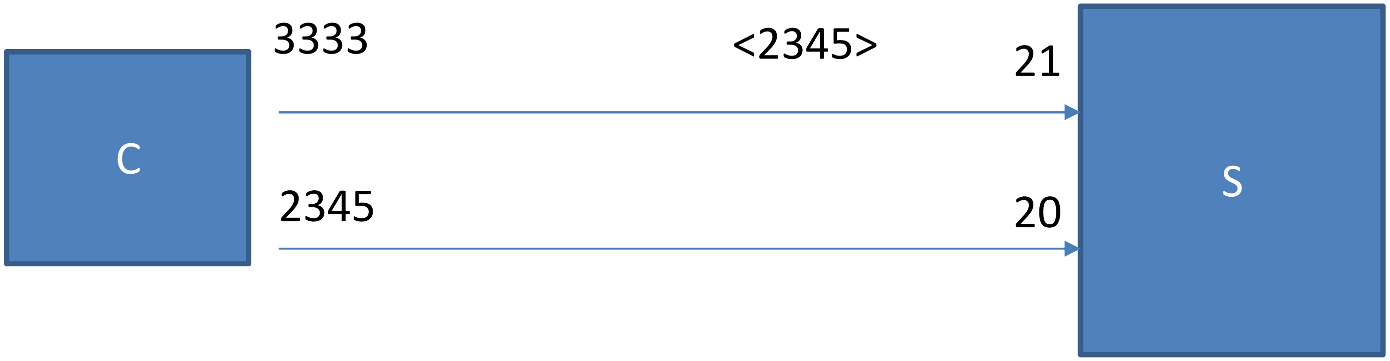
action	src	port	dest	port	flags	comment
allow	{our hosts}	*	*	25		our packets to their SMTP port
allow	*	25	*	*	ACK	their replies

Rule Set E

action	src	port	dest	port	flags	comment
allow	{our hosts}	*	*	*		our outgoing calls
allow	*	*	*	*	ACK	replies to our calls
allow	*	*	*	>1024		traffic to nonservers

Packet Filter - FTP Protocol

- Uses two static ports: 21 (command), 20 (data transmission)
- Active mode:
  - Client sends in the command session (port 21) the port that will be used in the data session (higher ports, selected randomly)
  - Server opens a data session from port 20 to the port sent by the client


3333
<2345>
21
C
S
2345
20

Packet Filter - FTP Protocol

- Solution: use passive mode
- Client sends pasv command in the command session
- Server sends random port (>1023)
- Client opens a session from a random port to the port sent by the server

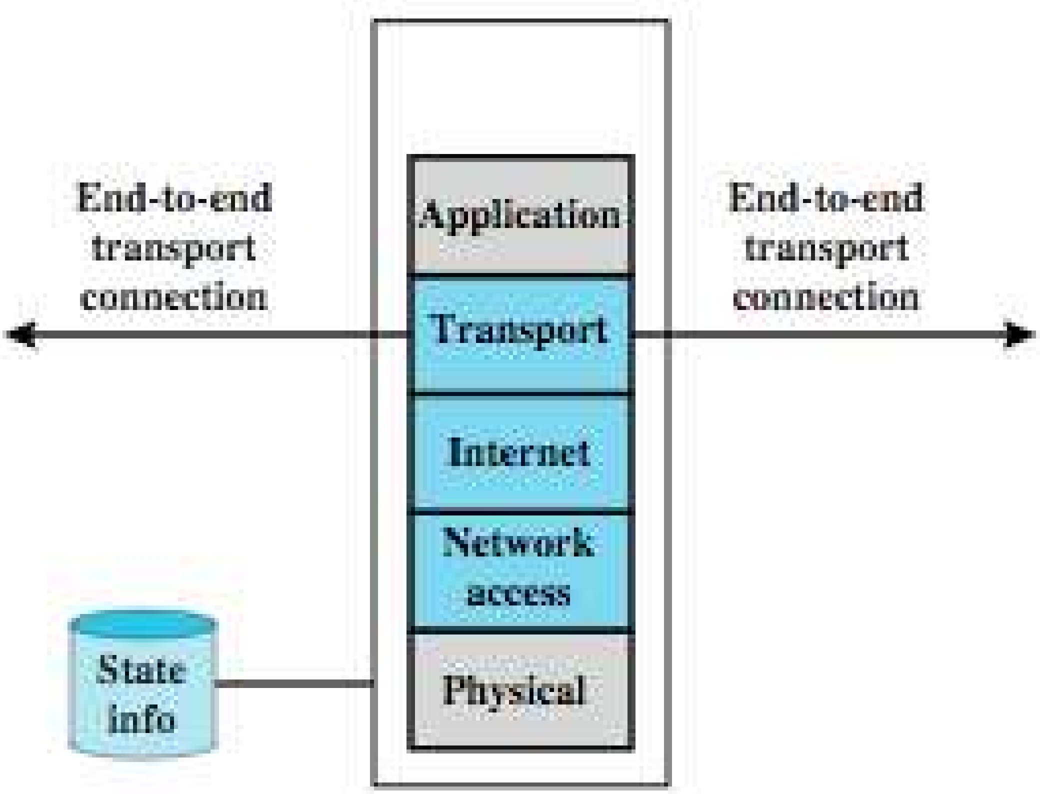
Packet Filter Weaknesses

- weaknesses
  - cannot prevent attack on application bugs (content is not examined)
  - limited logging functionality
  - do no support advanced user authentication
  - vulnerable to attacks on TCP/IP protocol bugs (e.g., network layer IP spoofing)
  - improper configuration can lead to breaches
  - Dynamic multi ports protocols (dynamic FTP)

Packet Filter Weaknesses

- attacks
  - IP address spoofing - send crafter packets with internal IP address
  - source route attacks - bypass security measures
  - tiny fragment attacks - fragmentation of TCP header information

Stateful Firewall


End-to-end transport connection
Application
Transport
Internet
Network access
Physical
End-to-end transport connection
State info

Stateful Inspection Firewall

- Keeps the context of a session

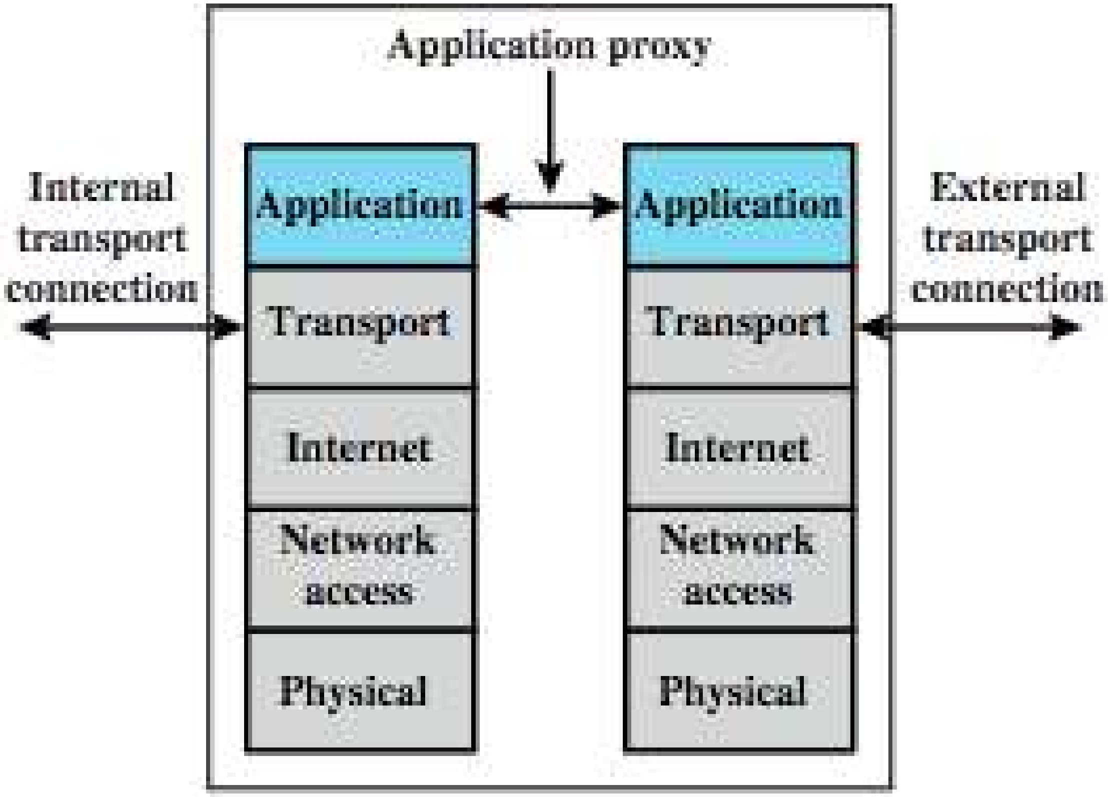
<protocol, src address, src port, dst address, dst port>

- Apply static rules on the first packet of the session
- Store all tuples of the session (drop others)
- Example, Simple Mail Transfer Protocol (SMTP)
  - TCP connection from client to mail server (port 25)
  - Local (client) port between 1024 - 65535

Stateful Inspection Firewall

- Reviews packet header information but also keeps info on TCP connections
  - typically have low, “known” port number for server
  - and high, dynamically assigned client port number
  - simple packet filter must allow all return high port numbered packets back in
  - stateful inspection packet firewall tightens rules for TCP traffic using a directory of TCP connections
  - only allow incoming traffic to high-numbered ports for packets matching an entry in this directory
  - may also track TCP seq numbers as well

Application Proxy Firewalls



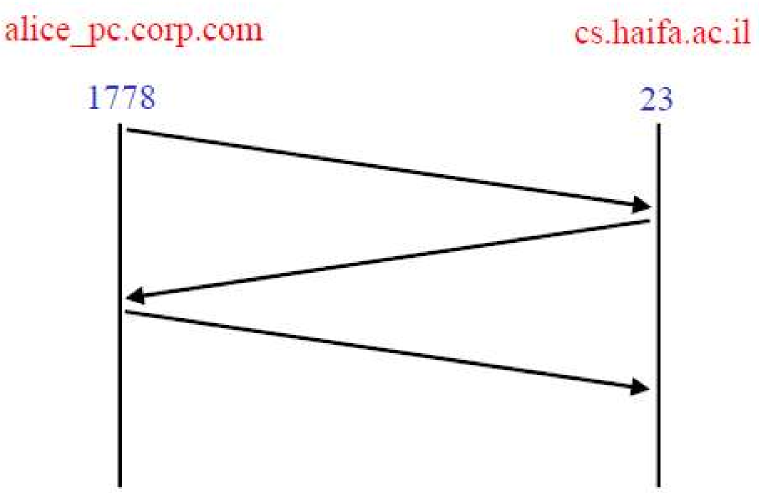
Application-Level Gateway

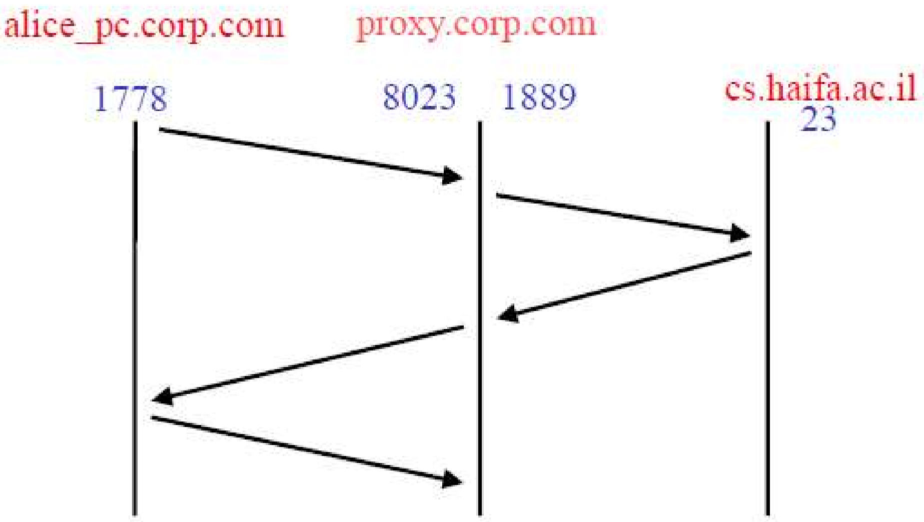
- Acts as a relay of application-level traffic (e.g., Browser, Mail) / legitimate Man in the Middle
  - user contacts gateway with remote host name
  - authenticates themselves
  - gateway contacts application on remote host and relays TCP segments between server and user
- Must have proxy code for each application
  - may restrict application features supported
- More secure than packet filters (can apply anti-malware scanning for example) but have higher overheads

Application-Level Gateway

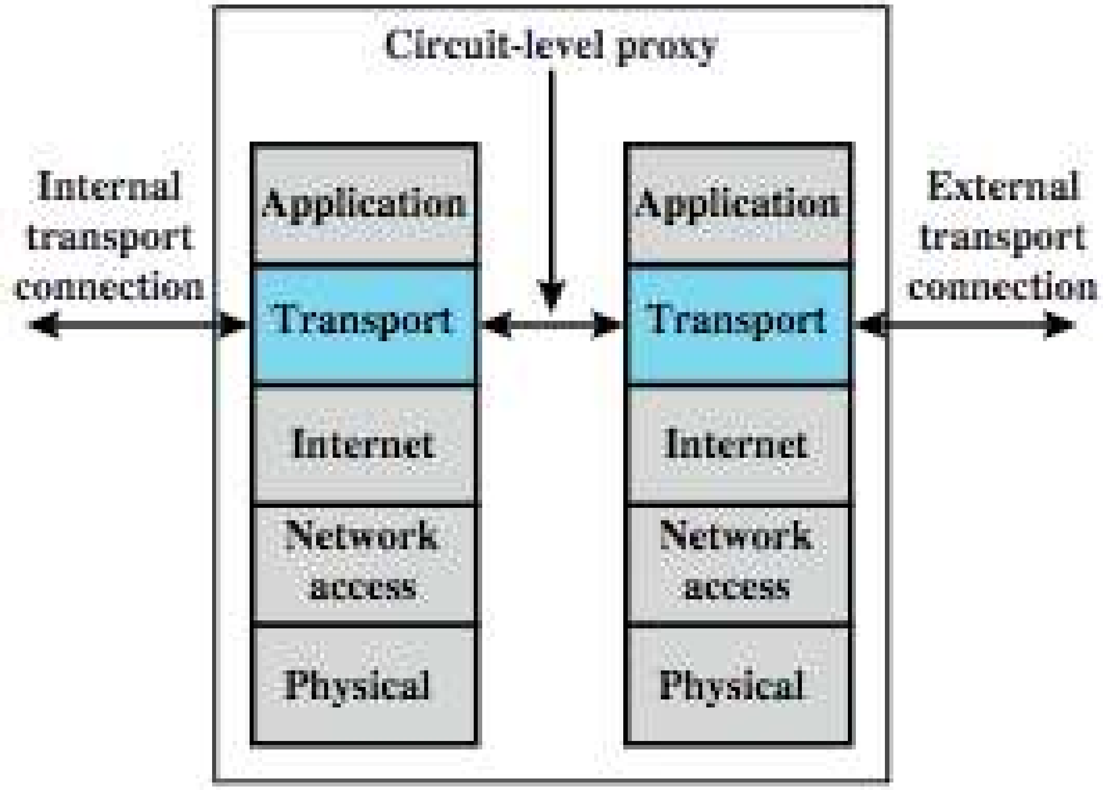
- Traffic should pass through the proxy (enforced by packet filters)
- No direct TCP communication between client and server
- Transparent to the user
- Difficult to configure

Telnet with/out proxy





Circuit Level Firewall



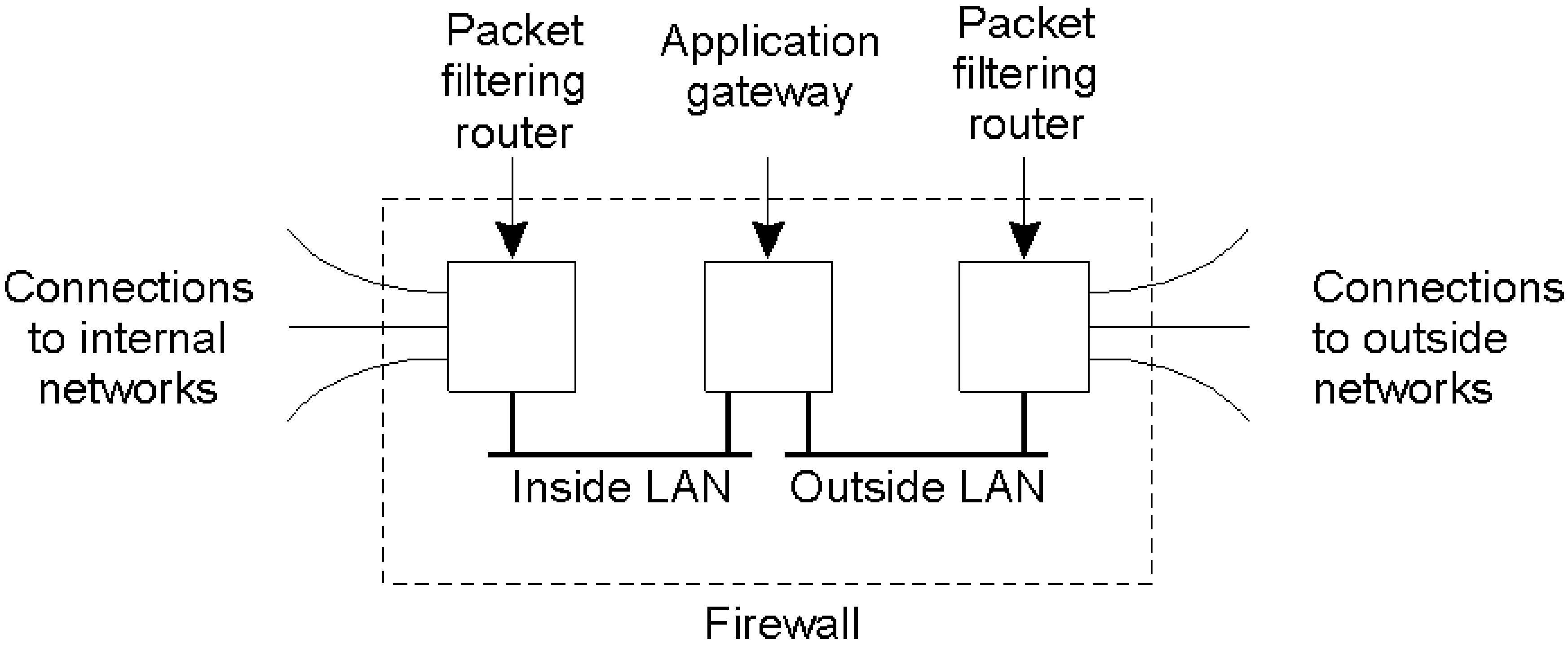
Circuit-Level Gateway

- Sets up two TCP connections, to an inside user and to an outside host
- Relays TCP segments from one connection to the other without examining contents
  - hence independent of application logic
  - just determines whether relay is permitted
- Typically used when inside users trusted
  - may use application-level gateway inbound and circuit-level gateway outbound
  - hence lower overheads

Connecting Mobile Users

- Use proxy server to implement access control and application level filtering
- Stateful – authenticated to the firewall and then keep the assigned IP
- Stateless – not possible

Common Implementation



Host-Based Firewalls

- Used to secure individual host
- Available in/add-on for many OS
- Filter packet flows
- Often used on servers
- Advantages:
  - tailored filter rules for specific host needs
  - protection from both internal / external attacks
  - additional layer of protection to org firewall

Personal Firewall

- Controls traffic flow to/from PC/workstation
- For both home or corporate use
- May be software module on PC or in home cable/DSL router/gateway
- Typically much less complex
- Primary role to deny unauthorized access
- May also monitor outgoing traffic to detect/block worm/malware activity

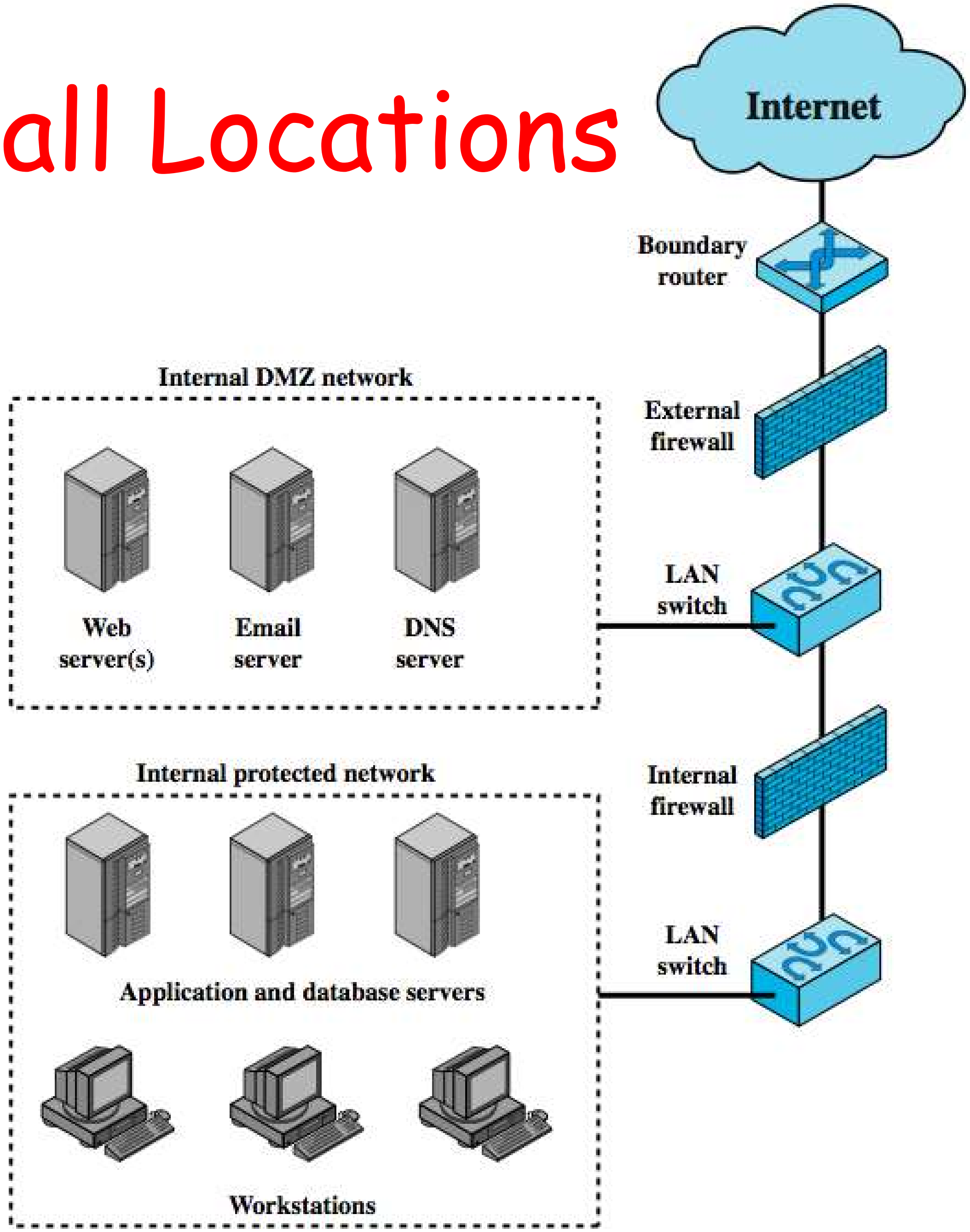
Bastion Host

- Can be accessed from the public network
- Can be accessed sometimes from the internal network
- Should not hold sensitive data
- Make sure that attackers cannot attack the internal/private network
  - serve the two networks, all Internet users, internal

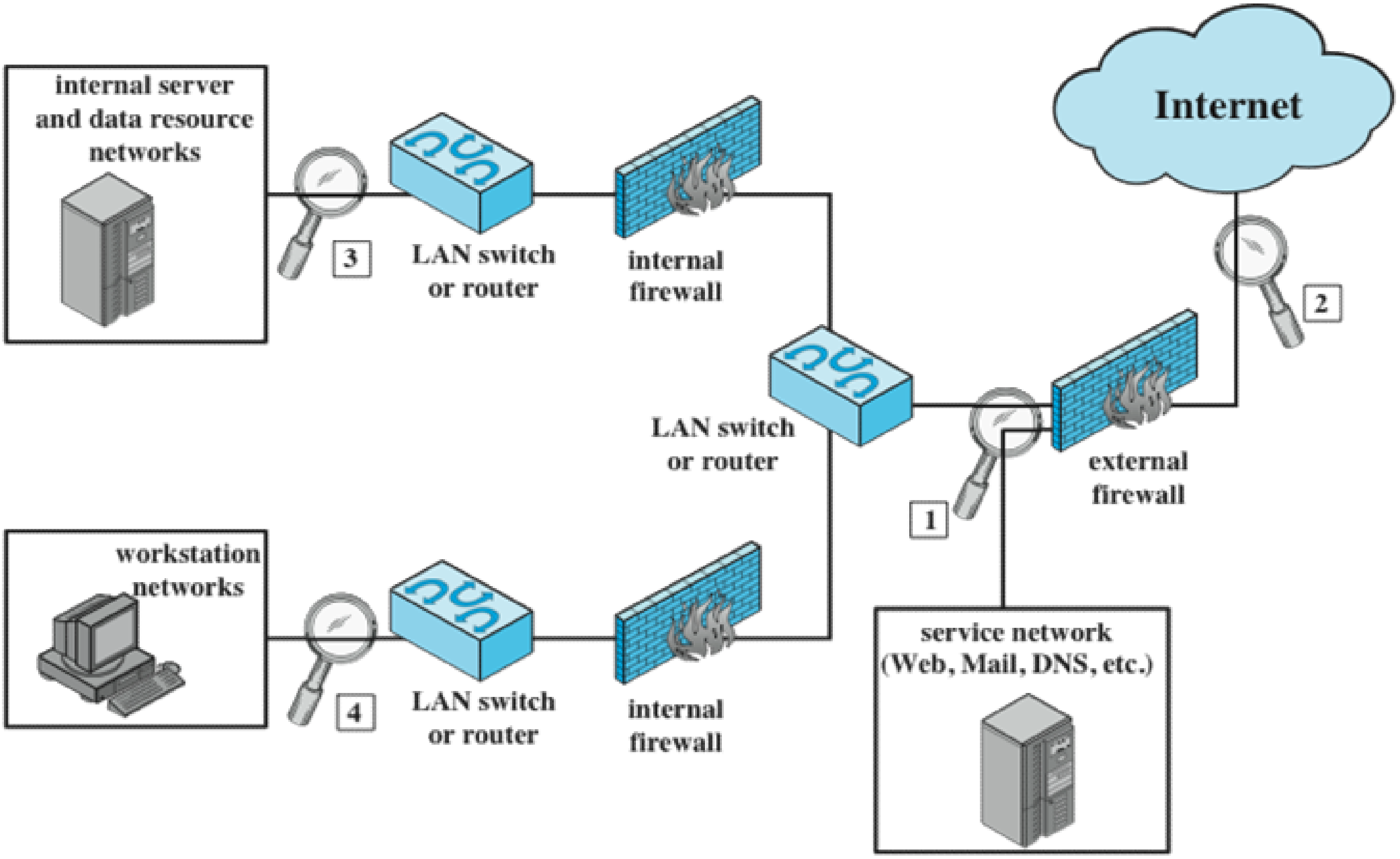
Demilitarized Zone

- Intermediate network separating the internal / private network and public network
- Usually hosts the bastion machines
- Additional security layer (e.g., can deploy proxies)

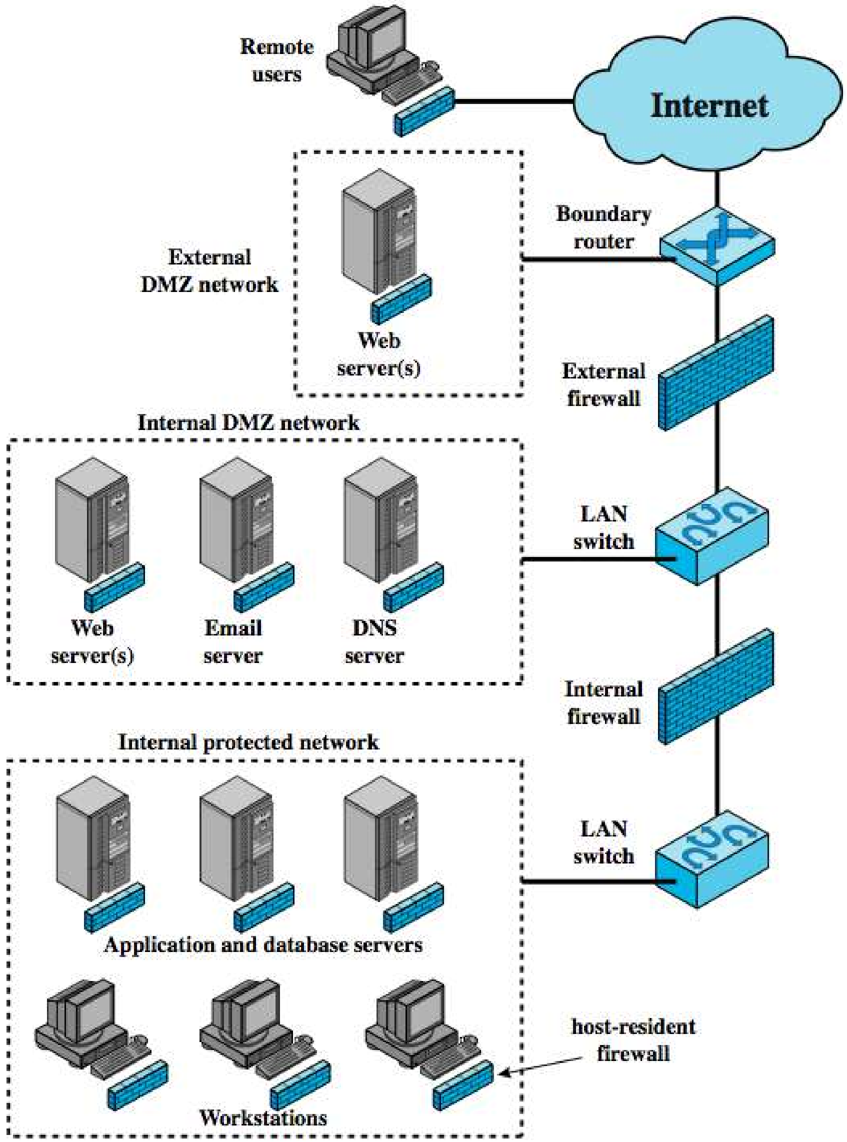
Firewall Locations



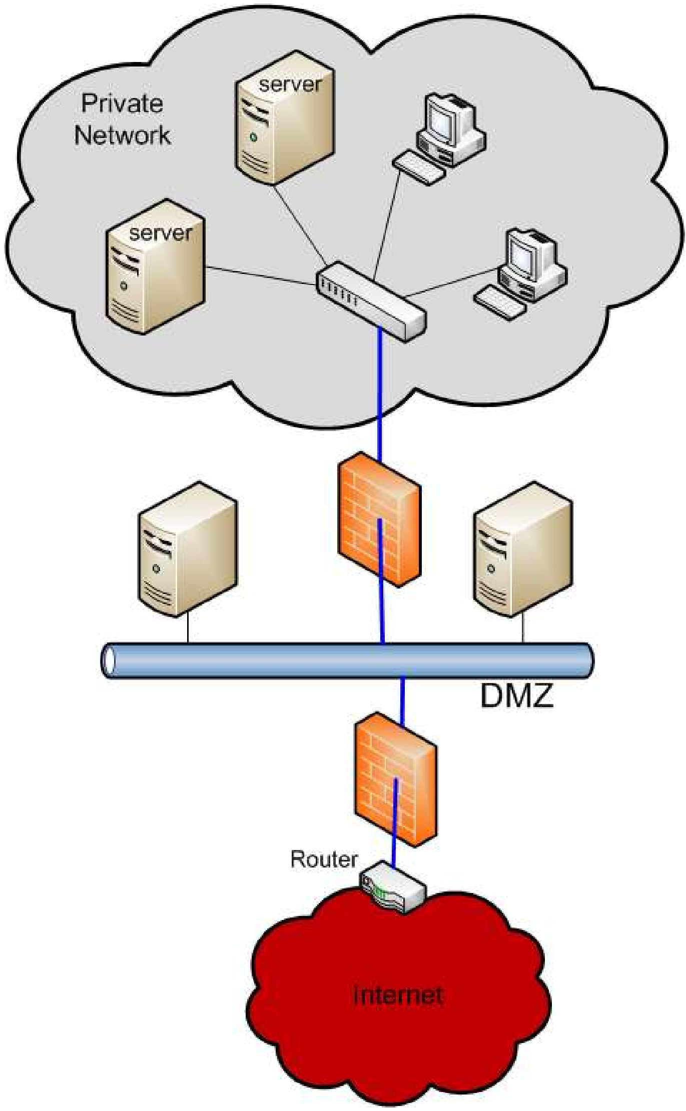
Firewall Locations

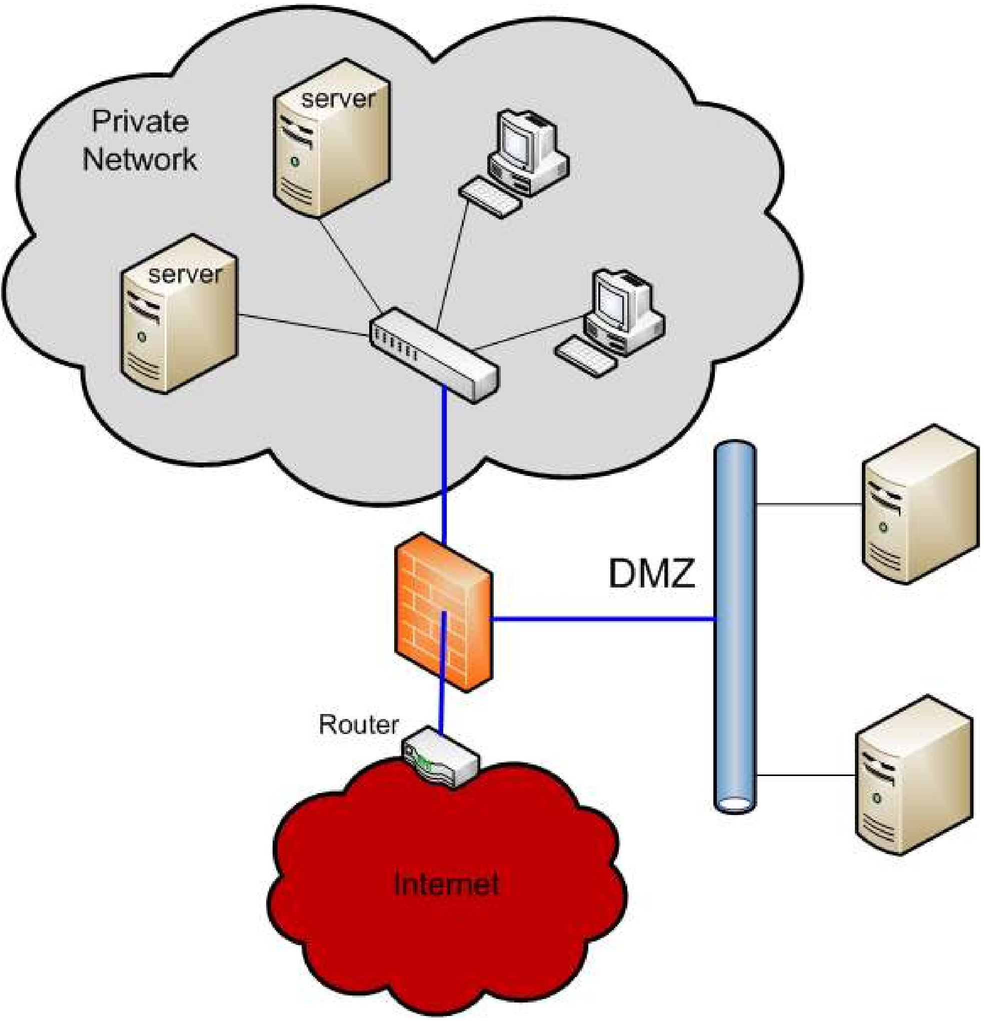


Demilitarized Zone

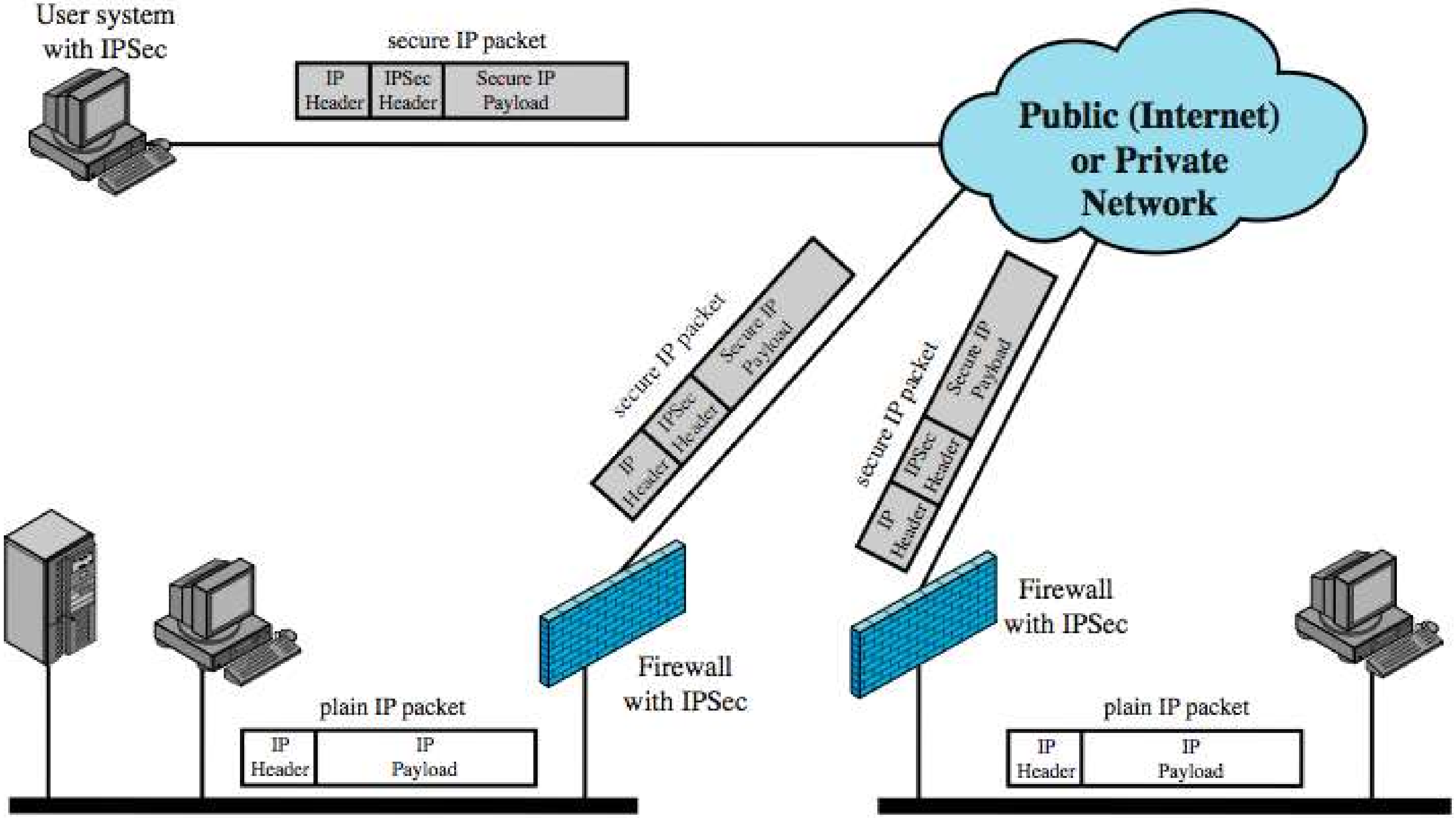


Demilitarized Zone





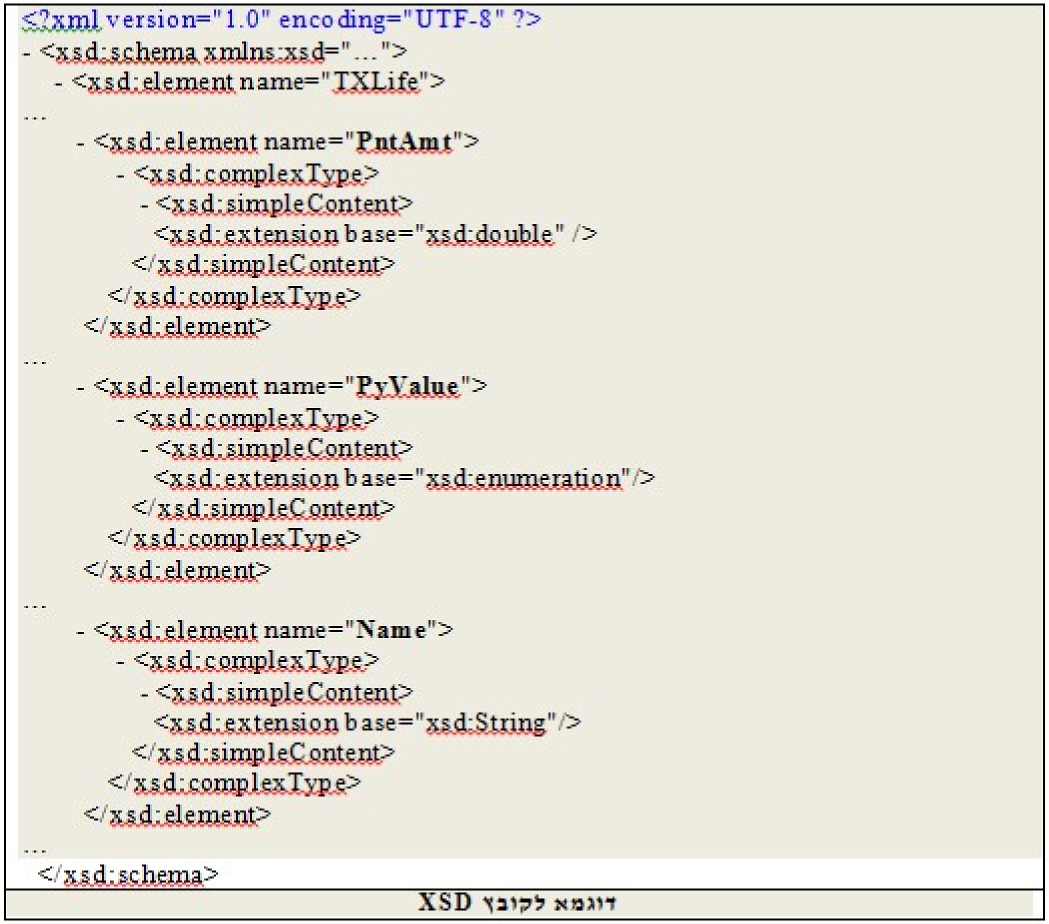
Virtual Private Networks



Firewalls

- Traffic not passing through the firewall is not protected
- Trust internal users
- Bypass using legitimate applications (FTP active mode, HTTP)
- Use additional solutions such as IDS

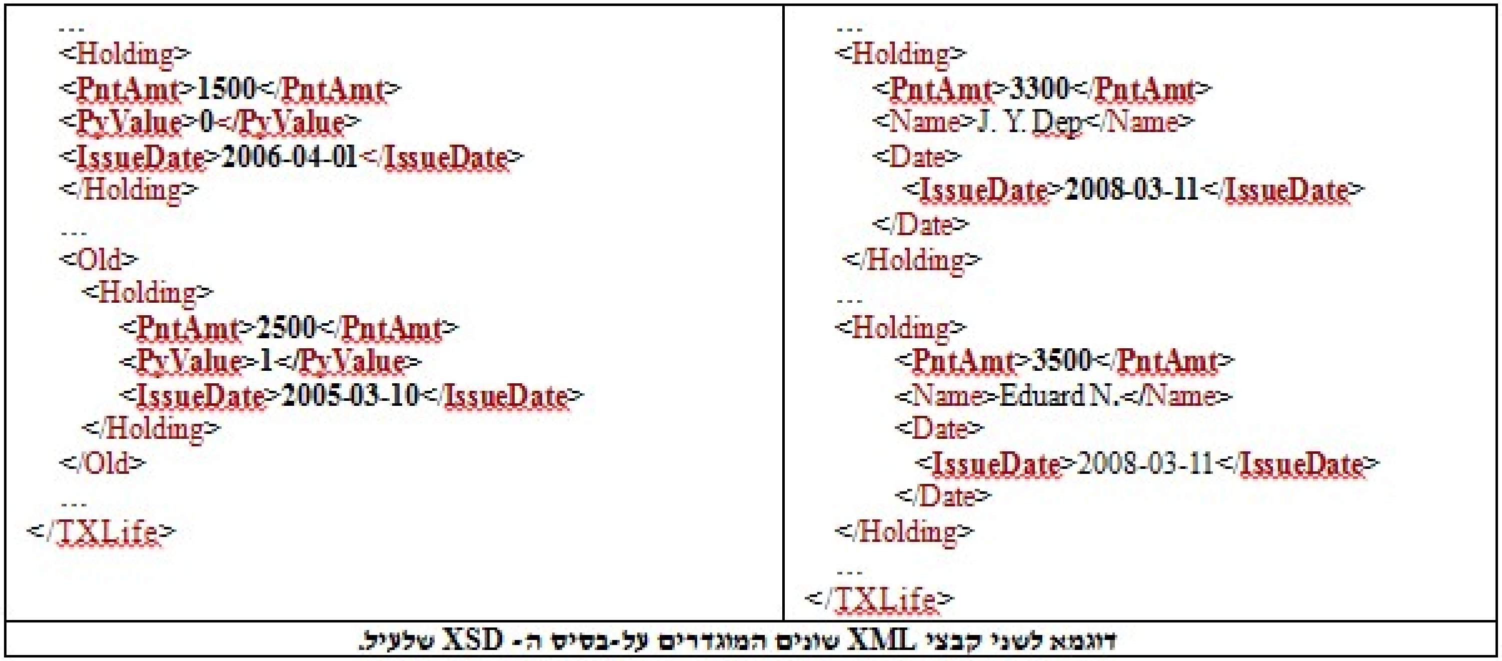
Example - XML firewall



```
<?xml version="1.0" encoding="UTF-8" ?>
- <xsd:schema xmlns:xsd="...">
   - <xsd:element name="TXLife">
...
     - <xsd:element name="PntAmt">
         - <xsd:complexType>
           - <xsd:simpleContent>
              <xsd:extension base="xsd:double" />
             </xsd:simpleContent>
           </xsd:complexType>
        </xsd:element>
...
     - <xsd:element name="PyValue">
         - <xsd:complexType>
           - <xsd:simpleContent>
              <xsd:extension base="xsd:enumeration"/>
             </xsd:simpleContent>
           </xsd:complexType>
        </xsd:element>
...
     - <xsd:element name="Name">
         - <xsd:complexType>
           - <xsd:simpleContent>
              <xsd:extension base="xsd:String"/>
             </xsd:simpleContent>
           </xsd:complexType>
        </xsd:element>
...
  </xsd:schema>
```

דוגמא לקובץ XSD

Example - XML firewall

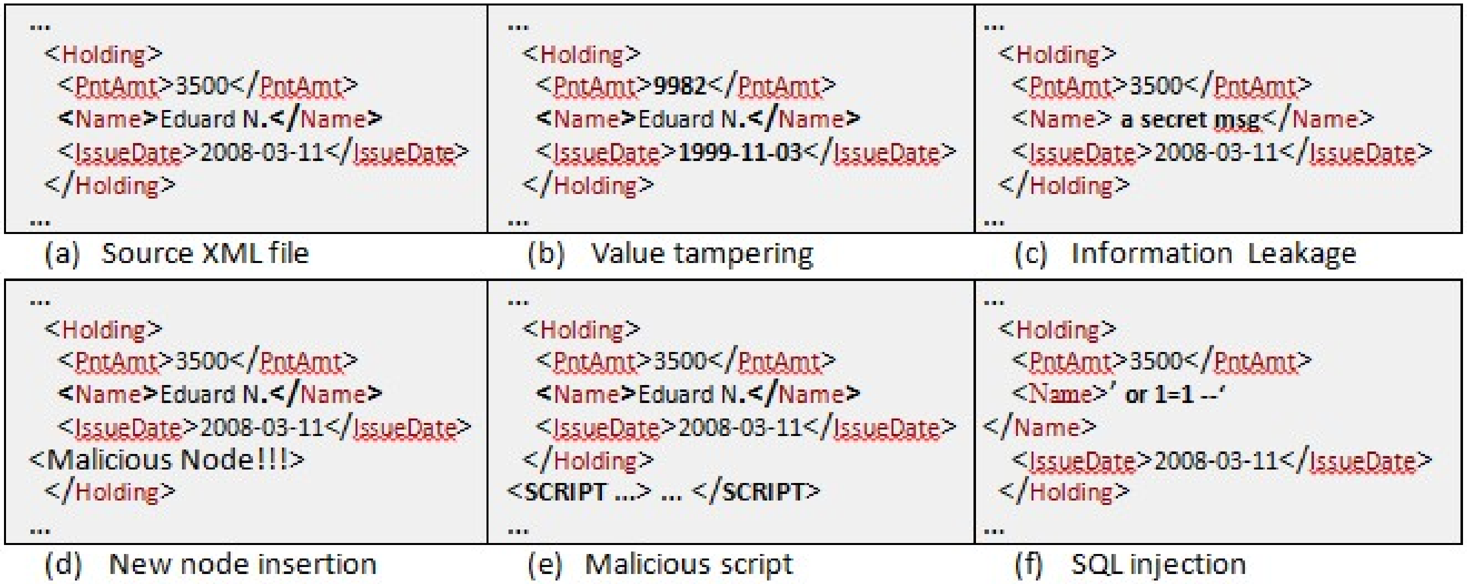


```
...
<Holding>
<PntAmt>1500</PntAmt>
<PyValue>0</PyValue>
<IssueDate>2006-04-01</IssueDate>
</Holding>
...
<Old>
  <Holding>
      <PntAmt>2500</PntAmt>
      <PyValue>1</PyValue>
      <IssueDate>2005-03-10</IssueDate>
  </Holding>
</Old>
...
</TXLife>
```

```
...
<Holding>
    <PntAmt>3300</PntAmt>
    <Name>J. Y. Dep</Name>
    <Date>
       <IssueDate>2008-03-11</IssueDate>
    </Date>
 </Holding>
...
<Holding>
      <PntAmt>3500</PntAmt>
      <Name>Eduard N.</Name>
      <Date>
        <IssueDate>2008-03-11</IssueDate>
      </Date>
</Holding>
...
</TXLife>
```

דוגמא לשני קבצי XML שונים המוגדרים על-בסיס ה- XSD שלעיל.

Example - XML firewall



```
...
 <Holding>
  <PntAmt>3500</PntAmt>
  <Name>Eduard N.</Name>
  <IssueDate>2008-03-11</IssueDate>
 </Holding>
...
```

(a) Source XML file

```
...
 <Holding>
  <PntAmt>9982</PntAmt>
  <Name>Eduard N.</Name>
  <IssueDate>1999-11-03</IssueDate>
 </Holding>
...
```

(b) Value tampering

```
...
 <Holding>
  <PntAmt>3500</PntAmt>
  <Name> a secret msg</Name>
  <IssueDate>2008-03-11</IssueDate>
 </Holding>
...
```

(c) Information Leakage

```
...
 <Holding>
  <PntAmt>3500</PntAmt>
  <Name>Eduard N.</Name>
  <IssueDate>2008-03-11</IssueDate>
<Malicious Node!!!>
 </Holding>
...
```

(d) New node insertion

```
...
 <Holding>
  <PntAmt>3500</PntAmt>
  <Name>Eduard N.</Name>
  <IssueDate>2008-03-11</IssueDate>
 </Holding>
<SCRIPT ...> ... </SCRIPT>
...
```

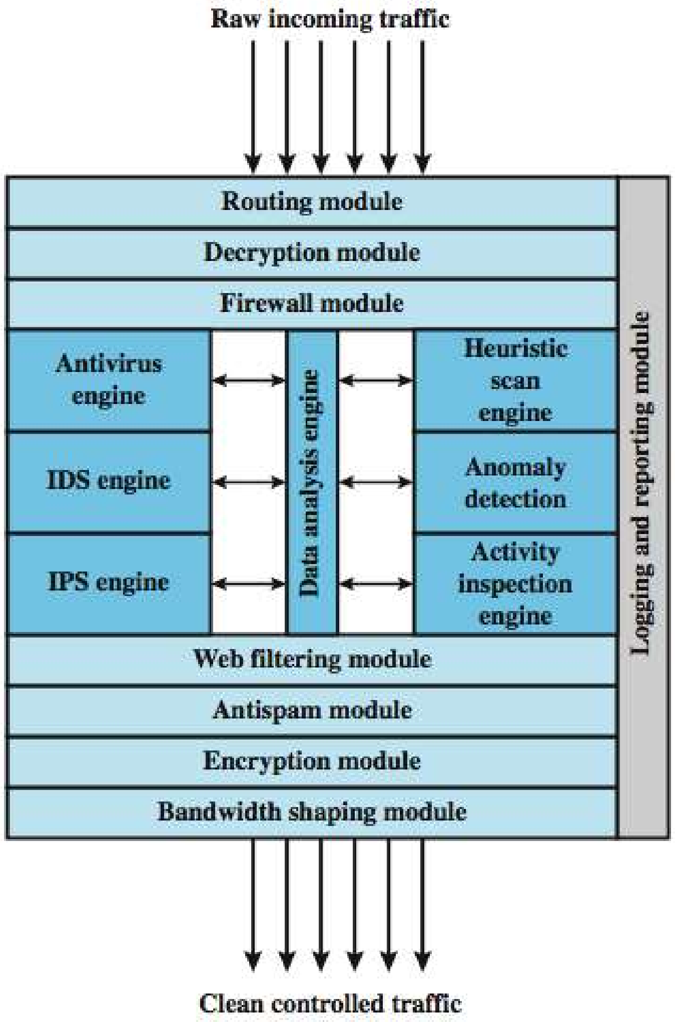
(e) Malicious script

```
...
 <Holding>
  <PntAmt>3500</PntAmt>
  <Name>' or 1=1 --'
</Name>
  <IssueDate>2008-03-11</IssueDate>
 </Holding>
...
```

(f) SQL injection

טבלה 1: חמש מניפולציות (התקפות) אפשריות על מסמכי XML

Unified Threat Management Products


Raw incoming traffic
Routing module
Decryption module
Firewall module
Antivirus engine
IDS engine
IPS engine
Data analysis engine
Heuristic scan engine
Anomaly detection
Activity inspection engine
Logging and reporting module
Web filtering module
Antispam module
Encryption module
Bandwidth shaping module
Clean controlled traffic