Firewall Architecture
Firewall Technology: What, Why and How?

- What is a firewall?
  - It is a packet filtering and inspection device based on packet headers and/or contents
  - Define security boundaries between untrusted and trusted networks by permitting and denying access to internal resources

- Why a firewall?
  - Block attacks or potential attacks
  - Eliminate “noise” traffic – save bandwidth ad improves performance
  - Restrict intra- or inter- subnet traffic and external network access
  - Logging network activities
Firewall Technology: What, Why and How?

• How does a firewall work?
  • A *firewall* is inspects packets (traffic flow) based on a firewall policy
  • A *filtering policy* is a list of ordered rules that define the required actions for matching or non matching packets.
  • A rule consists of *(Matching-condition) + (action)*
    • Rule format:
      `<#> <protocol><src_IP><src_port><dst_IP><dst_port> <action>`
Firewall Rules Evaluation

- The firewall matches a received packet against the rules sequentially starting by rule of lowest order.
- The matching stops as soon as a rule is evaluated to TRUE, then action is executed (permit or deny).
- Each rule has either permit or deny action.
- Default (recommended) last rule is to “DENY ALL”
- In a stateful firewall, the accepted traffic is allowed in both inbound and outbound directions.
- Firewalls can be configured as centralized to isolate the internal network from Internet or distributed to isolate internal domains from each other.
- Recommendation: filter traffic on the first interface it enters (inbound filtering) to reduce overhead.
Overview of Network Security Perimeters

- Boarder Routers: fast static filtering
- Firewalls: through filtering and inspection
- IDSs: sensors to monitor suspicious activities
  - report, analyze and correlate events
  - Network-based vs. Host-based
- VPN: secure tunnel over untrusted network
- Software: up-to-date patches and authentication
- DMZs: insecure areas between secure areas
  - has public services
- Screen subnets: trusted (secure) network
The Role of Network Security Perimeters

- Firewall, IPSec, IDS/IPS ..etc are core in any network security infrastructure
  - Prevent attacks via Ingress and egress filtering
  - Prevent spoofing via authentication or egress filtering
  - Limit/prevent access to critical domains/server from outside or inside
  - Limit/prevent access to outside sites and servers
  - Extend trust while preserving authentication and privacy
  - Detect intruders and misbehaving users

- Network security devices operates based on security polices
  - List of ordered rules evaluated usually sequentially
  - Rule Format:
    `<#> <protocol><src_IP><src_port><dst_IP><dst_port> <action>
  - Types of actions: accept, deny, auth, encry, analyze, log, redirect, ..
The Role of Network Security Perimeters

- Network security devices are usually distributed
  - Multiple domains with different security requirements
  - Autonomous administration
  - Defense in-depth
  - Scalability
Firewalls

• Types of Firewalls
• Inspection Methods
  • Static Packet Inspection
  • Stateful Packet Inspection
  • NAT
  • Application Firewalls
• Firewall Architecture
• Firewall Policy Advisor
• Configuring, Testing, and Maintenance
Stateless Packet Filtering Firewalls

- Filters that make decisions on a packet-by-packet basis
  - Usually check headers only
  - you can not create rules that filters packet based on other packets or previous history
  - check each packet in the same stream in isolation from each other
- Provide light filtering only ➔ Fast (more packet per second)
- Screens out incoming “noise” of simple scanning attacks to make the detection of serious attacks easier
- Deployed usually at the boarder router (screening router) of the network for egress filtering—can eliminate scanning responses, even from the router
Border Firewall: Ingress/Egress Filtering

**Trusted Network**

- Client PC
  - Passed Packet (Ingress)
  - Dropped Packet (Ingress)
- Server
  - Log File

**Internet (Not Trusted)**

- Attack Packet
- Internet Border Firewall

- Passed Packet (Egress)
- Attack

- Log File

- Server

- Client PC

- Internet (Not Trusted)
Stateless Packet Filtering Firewalls

- Filters that make decisions on a packet-by-packet basis
  - Usually check headers only
  - You can not create rules that filters packet based on other packets or previous history
  - Check each packet in the same stream in isolation from each other

- Provide light filtering only ➔ Fast (more packet per second)

- Screens out incoming “noise” of simple scanning attacks to make the detection of serious attacks easier

- Deployed usually at the boarder router (screening router) of the network for egress filtering—can eliminate scanning responses, even from the router
Filtering Types and Parameters

- Filtering is mainly done on the “five-tuple” information on the protocols header
- Filtering based on the protocols
  - Deny all protocols other than TCP (eg., udp, IPX, ..etc)
  - Deny any incoming ICMP echo request but allow replies
- Filtering based on the IP source (black listing)
  - Private, 0.0.0.0, 127.0.0.1, your network address, Invalid IP, infected domains, known hacker (scanner), undesired sites
- Filtering based on the IP destination
  - Block special secret servers
  - Private/internal address
- Filtering based on the Port source
  - Deny packets coming from IRC, Napster, instant messaging
- Filtering based on the Port destination
  - Block packet coming to unused or private services
  - See www.sans.org/top20.htm for top 20 vulnerability issues
Access Control List (ACL) For Ingress Filtering at a Border Router

1. If source IP address = 10.*.*.*, DENY [private IP address range]
2. If source IP address = 172.16.*.* to 172.31.*.*, DENY [private IP address range]
3. If source IP address = 192.168.*.*, DENY [private IP address range]
4. If source IP address = 0.0.0.0, DENY [invalid IP address range]
5. If source IP address = 127.0.*.*, DENY [invalid IP address range]
6. If source IP address = 140.192.*.*, DENY [DePaul address range but outside pkt]
7. If source IP address = 1.2.3.4, DENY [black-holed address of attacker]
8. If TCP SYN=1 AND FIN=1, DENY [crafted attack packet to send RST]
9. If destination IP address = 140.192.3.9 AND TCP destination port=80 OR 443, PERMIT [connection to a public webserver]
10. If ICMP Type = 0, PERMIT [allow incoming echo reply messages]

<Last Rule> DENY ALL (might be DEFAULT)
Limitation of Static Packet Filtering

- Misses many attacks (e.g., SYN Flooding)
- No protection against *spoofing*
- No protection against fragments
  - Some FW pass fragments without checking
  - Some FW check the first fragment only (not enough. Why?)
  - Some FW create a full state by reassembling the fragments and check the packet (to avoid fragment overlap)
- Allowing retuning traffic is difficult
  - Examples: DNS, one-way Telnet initiation
- Does not distinguish between returning traffic and crafted packets (Acks) that look like replies (If a TCP SYN/ACK segment is sent, cannot tell if there was a previous SYN to open a connection)
Limitation of Static Packet Filtering

- Cannot deal with port-switching applications (FTP)
- Cisco provides dynamic packet filtering (reflexive access list) to solve some of these problems – but still it is limited
Types of Firewall Inspection

- **Packet Inspection**
  - Examines IP, TCP, UDP, and ICMP header contents

- **Static packet filtering** looks at individual packets in isolation. Misses many attacks

- **Stateful inspection** inspects packets in the context of the packet’s role in an ongoing or incipient conversation
  - Stateful inspection is the proffered packet inspection method today
Stateful Inspection Firewalls

- **Stateful Firewall Operation**
  - By default, permit connections openings from internal clients (on trusted network) to external servers (on untrusted network)
  - By default, deny connection openings from the outside to inside servers
  - These default behaviors can be changed with ACLs
  - Accept future packets between hosts and ports in open connections with little or no more inspection
Stateful Inspection Firewalls

- To improve the performance, not every single packet is matched against the rules. Only the SYN (or the initial packet of a UDP stream) packet is matched then the stream 5-tuple will be logged in the table.
- Ingress: All other packets related to already accepted stream (or open connection) are matched against the table (much faster) rather than filtering rules.
- If there is no match in the table, then the packet is matched against the filtering rules.
- FIN and Timers are used to expire table entries.
- Egress: It also logs TCP/UDP requests in the table to permit to TCP/UDP replies passing back to secure client.
Stateful Inspection Firewalls

- Types of State of Connection
  - Full state: Order of packet within a dialog
  - Open-close state: Often simply whether the packet is part of an open connection
  - Open-close and fragment state
Stateful Inspection Firewall
Operation I

1. Establish Connection
   - TCP SYN Segment
     - From: 60.55.33.12:62600
     - To: 123.80.5.34:80
   - Note: Outgoing Connections Allowed By Default

2. Stateful Firewall

3. TCP SYN Segment
   - From: 60.55.33.12:62600
   - To: 123.80.5.34:80

Connection Table

<table>
<thead>
<tr>
<th>Type</th>
<th>Internal IP</th>
<th>Internal Port</th>
<th>External IP</th>
<th>External Port</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>60.55.33.12</td>
<td>62600</td>
<td>123.80.5.34</td>
<td>80</td>
<td>OK</td>
</tr>
</tbody>
</table>
Stateful Inspection Firewall
Operation I

Internal Client PC 60.55.33.12

6. TCP SYN/ACK Segment
   From: 123.80.5.34:80
   To: 60.55.33.12:62600

Stateful Firewall

4. TCP SYN/ACK Segment
   From: 123.80.5.34:80
   To: 60.55.33.12:62600

External Webserver 123.80.5.34

5. Check Connection
   OK

Connection Table

<table>
<thead>
<tr>
<th>Type</th>
<th>Internal IP</th>
<th>Internal Port</th>
<th>External IP</th>
<th>External Port</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>60.55.33.12</td>
<td>62600</td>
<td>123.80.5.34</td>
<td>80</td>
<td>OK</td>
</tr>
</tbody>
</table>
Stateful Inspection Firewalls

- Stateful Firewall Operation
  - For UDP, also record two IP addresses in port numbers in the state table

### Connection Table

<table>
<thead>
<tr>
<th>Type</th>
<th>Internal IP</th>
<th>Internal Port</th>
<th>External IP</th>
<th>External Port</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>60.55.33.12</td>
<td>62600</td>
<td>123.80.5.34</td>
<td>80</td>
<td>OK</td>
</tr>
<tr>
<td>UDP</td>
<td>60.55.33.12</td>
<td>63206</td>
<td>1.8.33.4</td>
<td>69</td>
<td>OK</td>
</tr>
</tbody>
</table>
Application-level Filtering Firewalls

- Application Inspection and Content Filtering
  - Examines application layer messages during session initiation
  - Stops some attacks that packet inspection cannot
  - Some stateful firewall might have application-level inspection for some applications like FTP, RTSP and H.323 (you need to ask what kind of application-level inspection support this firewall provides)

- Limitation: might cause vulnerability because
  - Table entries for some applications must expire in order to be evicted because it does not track the session closing
  - It does not inspect every packet (e.g., some other traffic (worm) could be tunneled thru FTP or Web connections or thru some web sites such as www.go2mypc.com)
Also Called Proxy Firewalls

- Inspects every packet in the conversation
- Proxy server does not bypass packets but it acts as client/server between the two ends.
- Proxy server must understand the application (one proxy per application)
- Proxy server can check incoming and outgoing traffic (e.g., Web and FTP applications)
- Example: warms, blocking transferring sensitive information or command such as “deny HTTP PUT”.
- Proxy Types:
  - Classical Proxies: clients have to connect to proxy first
  - Transparent Proxies: proxies intercept client IP packets and issue the connection to remote server
- General proxy may not work well (secure) for specific or not well-known applications (TIS Firewall tool kit might be used to make a custom proxies)
Firewalls - Application Level Gateway (or Proxy)
Application-Level Filtering

- Has full access to protocol
  - user requests service from proxy
  - proxy validates request as legal
  - then actions request and returns result to user
- Need separate proxies for each service
  - E.g., SMTP (E-Mail)
  - NNTP (Net news)
  - DNS (Domain Name System)
  - NTP (Network Time Protocol)
  - custom services generally not supported
Daemon spawns proxy when communication detected …
Proxy Server Evaluation

- Advantages
  - Hiding network information
  - application/content-level filtering
  - fail over and load balancing features
  - single-point of control (easy to control access)
  - powerful logging features

- Disadvantages
  - increases the communication latency/delay
  - proxy per application and no generic one
  - client might need to be modified/reconfigured to use the proxy server
Overview of Network Security Architecture
Security Design Principles

- Least Privilege
- Fail-Safe Defaults
- Economy of Mechanism
- Complete Mediation
- Open Design
- Separation of Privilege
- Least Common Mechanism
- Psychological Acceptability
Security Design Principles

- **Least Privilege**: A subject should be given only those privileges necessary to complete its task
  - Function, not identity, controls
  - Rights added as needed, discarded after use
  - Minimal protection domain
  - E.g., different user must have different rules

- **Fail-Safe Defaults**: Default action is to deny access
  - If action fails, system as secure as when action began

- **Economy of Mechanism**: Keep it as simple as possible
  - Aggregate rules, domains
  - Simpler means less can go wrong and when errors occur, they are easier to understand and fix
  - Interfaces and interactions
Security Design Principles

- **Complete Mediation**: Check every access
  - Usually done once, on first action
  - UNIX: access checked on open, not checked thereafter
  - If permissions change afterwards, may get unauthorized access
  - Multiple devices check the same traffic in the path

- **Separation of Privilege**: Require multiple conditions to grant privilege
  - Separation of duty
  - Defense in depth

- **Least Common Mechanism**: Mechanisms (resource) should not be shared
  - Information can flow along shared channels (e.g., using Internet allows both legitimate user and attacker to be in the same boat)
  - Covert channels
  - Enforcement Examples
    - Isolation: Virtual machines, Sandboxes
    - DMZ (separate internal and external services)
Security Design Principles

- **Open Design**: Security should not depend on secrecy of design or implementation
  - Popularly misunderstood to mean that source code should be public
  - “Security through obscurity”
  - Does not apply to information such as passwords or cryptographic keys

- **Psychological Acceptability (Usability)**: Security mechanisms should not add to difficulty of accessing resource
  - Hide complexity introduced by security mechanisms
  - Ease of installation, configuration, use
  - Human factors critical here
Key Points

- Principles of secure design underline all security-related mechanisms
- Require:
  - Good understanding of goal of mechanism and environment in which it is to be used
  - Careful analysis and design
  - Careful implementation
Top 5 best practices for firewall administrators, Network World September 11, 2009

- Document all firewall rule changes.
  - Change management
  - Accountability
- Install all access rules with minimal access rights.
  - overly permissive rules (e.g., open up 65,535 attack vectors for hackers)
- Verify every firewall change against compliance policies and change requests.
  - Verify that rules meets the spirit and intent of the security policy and any compliance policies, not just the letter of the law.
- Remove unused rules from the firewall rule bases when services are decommissioned.
  - No one tells you about unused port or server
  - Can be used for scanning and attacks
- Perform a complete firewall review at least twice per year.
  - Policy and networks evolves
  - Remove redundant, irrelevant and temporally exceptions
Centralized Firewalls Architectures

Centralized Firewall

Internet

outbound interface

Firewall

inbound interface

Secured Network
Decentralized Firewall Architecture
Distributed Firewall Architecture
Firewall Architectures & Evaluation

- **Centralized**
  - (+) easy to manage
  - (-) performance Bottleneck
  - (-) a thin security layer
  - (-) single point of failure
  - (-) provides only central one-fits-all policies -- no custom policies ➔ allow for spurious traffic

- **Decentralized**
  - (+) no single point of failure or bottleneck
  - (-) no network protection ➔ more vulnerable DOS

- **Distributed**
  - (+) no single point of failure or bottleneck
  - (+) defense-in-depth
  - (+) local (domain) control and global protection
  - (+) no waste of bandwidth and less vulnerable for DOS
  - (-) complex to manage
Firewall Architectures -- Evaluation

• Centralized (may be ok)
  • (+) easy to manage
  • (-) performance Bottleneck
  • (-) a thin security layer
  • (-) single point of failure
  • (-) provides only central one-fits-all policies -- no custom policies

• Decentralized
  • (+) no single point of failure or bottleneck
  • (-) no network protection ➔ more vulnerable DOS

• Distributed
  • (+) no single point of failure or bottleneck
  • (+) defense-in-depth
  • (+) local (domain) control and global protection
  • (+) no waste of bandwidth and less vulnerable for DOS
  • (-) complex to manage
Firewall Architectures -- Evaluation

• Centralized *(may be ok)*
  • (+) easy to manage
  • (-) performance Bottleneck
  • (-) a thin security layer
  • (-) single point of failure
  • (-) provides only central one-fits-all policies -- no custom policies

• Decentralized *(Big Mistake)*
  • (+) no single point of failure or bottleneck
  • (-) no network protection ➔ more vulnerable DOS

• Distributed
  • (+) no single point of failure or bottleneck
  • (+) defense-in-depth
  • (+) local (domain) control and global protection
  • (+) no waste of bandwidth and less vulnerable for DOS
  • (-) complex to manage
Firewall Architectures -- Evaluation

- **Centralized** *(May be ok)*
  - (+) easy to manage
  - (-) performance Bottleneck
  - (-) a thin security layer
  - (-) single point of failure
  - (-) provides only central one-fits-all policies -- no custom policies

- **Decentralized** *(Big Mistake)*
  - (+) no single point of failure or bottleneck
  - (-) no network protection ➔ more vulnerable DOS

- **Distributed** *(Promising)*
  - (+) no single point of failure or bottleneck
  - (+) defense-in-depth
  - (+) local (domain) control and global protection
  - (+) no waste of bandwidth and less vulnerable for DOS
  - (-) complex to manage
Defense in Depth

- Organizing security protection in layers like an onion
  - no security layer can guarantee to withstand every attack (see http://www.sans.org/top20/)
  - distributing responsibilities and load
  - depends on the organization size, risk level, how critical, application requirements

- Using multiple crypto systems in different layers
  - Like SSL and IPSec

- Divining the domain based on the security importance/security sensitivity, risks and requirements
Defense-in-Depth Using Firewalls

- Static Ingress and Egress stateless filtering in boarder routers: ICMP, Multicast
- Distributed Stateful firewalls to segregate subnets of different security risk and requirements
- Using DMZ (external and internal DMZ)
- Distributed Firewalls to segregate resources and domains
- Proxy firewalls: viruses, attachments, contents
- Host/Personal firewall: customized protection, protection from internal threats and dialup breach/attack
- Using integrated and circuit firewall at hot-spot points
- Using redundant firewalls
Single-Site Firewall Architecture

1. Screening Router
   60.47.1.1 Last
   Rule=Permit All

Internet

172.18.9.x Subnet

Marketing Client on 172.18.5.x Subnet

Accounting Server on 172.18.7.x Subnet

Public Webserver 60.47.3.9
SMTP Relay Proxy 60.47.3.10
HTTP Proxy Server 60.47.3.1

External DNS Server 60.47.3.4
2. Main Firewall
Last Rule=Deny All

172.18.9.x Subnet

Marketing Client on 172.18.5.x Subnet
Accounting Server on 172.18.7.x Subnet

Internet

Public Webserver 60.47.3.9
SMTP Relay Proxy 60.47.3.10
HTTP Proxy Server 60.47.3.1

External DNS Server 60.47.3.4
3. **Internal Firewall Traffic Between Subnets**

- **Internet**
- **Public Webserver (60.47.3.9)**
- **External DNS Server (60.47.3.4)**
- **SMTP Relay Proxy (60.47.3.10)**
- **HTTP Proxy Server (60.47.3.1)**

**Network Layout**
- **Marketing Client on 172.18.5.x Subnet**
- **Accounting Server on 172.18.7.x Subnet**

**Firewall Configuration**
- **172.18.9.x Subnet**
Single-Site Firewall Architecture

- Internet
- 172.18.9.x Subnet
- Public Webserver
- SMTP Relay
- Proxy
- HTTP Proxy
- Server
- External DNS Server
- Marketing Client on 172.18.5.x Subnet
- Accounting Server on 172.18.7.x Subnet
- 6. DMZ

Firewall

- Public Webserver
- SMTP Relay
- HTTP Proxy Server
- External DNS Server
DMZ

- Demilitarized Zone-- it allows you to create a public access segment of your network for use by external clients without breaching the security of your network. This area will not be fully secured.

- For Servers That Must be Accessed From the Outside
  - Public webservers
  - Application (proxy) firewalls
  - DNS server that only knows the IP addresses of hosts in the firewall
  - Hosts must be specially hardened because they certainly will be attacked.
Defense in Depth Concept

- Organizing security protection in layers like an onion
- no security layer can guarantee to withstand every attack (see [http://www.sans.org/top20/](http://www.sans.org/top20/))
- distributing the responsibilities and the load
- depends on the organization size, risk level, how critical, application requirements
- Using multiple crypto systems in different layers
  - SSL and IPSec
Defense in Depth Concept

- Using multiple security perimeters
  - Static Ingress and Egress filtering in boarder routers: ICMP, Multicast
  - Stateful firewalls: UDP, TCP, established session, returning replies
  - Distributed Firewalls to segregate resources and domains
  - Proxy firewalls: viruses, attachments, contents
  - Personal firewall: customized protection, protection from internal threats and dialup breach/attack
- IDS: uncaught malicious activities
  - worms propagation, DOS, DDOS
  - logs and traffic statistical analysis
Defense in Depth Concept

- Using multiple security perimeters
  - VPN devices: isolation but not necessary secure
    - attacker gain access to your VPN-computer (thru worm ..etc), will have a trusted ticket to others!!
  - Antivirus protection: signatures
  - Operating system hardening: patches, permission, open (unused) services
  - Configuration Management: automatic updates
  - Audits: testing and verification (is my policy/expectation implemented and enforced?)
DMZ (cont.)

- External DMZ—between boarder router and main FW (e.g., risky trusted servers like dial-up and DNS sever, and heavy loaded servers/gateways
  - Provides limited security
- Internal DMZ- public servers
- DMZ Pockets- to isolate protected servers without overloading main FW
  + Fine-grain security & defense-in-depth
  + scalability
  - complex to manage (conflicts)
  - more delay in the data path
- Parallel Firewall
  - Traffic isolation
  - Faster path for corporate network
- Mix of DMZ pockets and parallel firewalls is ultimate solution
DMZ Regulations

- No initiated traffic from DMZ to the secure subnet should be allowed
- Split services used by local and external users
  - Public DNS I in the DMZ
  - Internal DNS in the trusted network
- Place in external DMZ high risky and busy servers
- You might put ext-web and ext-DNS in ext-DMZ but email-relay must be in internal-DMZ
- Isolate wireless networks completely in separate firewall zone—fw allows only limited services for wireless user and also can use IDS specially in this area
- Deeper DMZ have less restrictions (defense-in-depth)
IPSec Security Policy: Example

TCP 1.1.*.* : any 2.2.*.* : any protect
TCP 1.1.1.1 : any 2.2.2.2 : any AH Transport {MD5}
TCP 1.1.*.* : any 2.2.*.* : any protect
TCP 1.1.1.* : any 2.2.2.* : any ESP Tunnel 6.6.6.6 {3DES}
TCP 2.2.*.* : any 1.1.*.* : any protect
TCP 2.2.2.* : any 1.1.1.* : any ESP Tunnel 5.5.5.5 {3DES}
TCP 2.2.*.* : any 1.1.*.* : any protect
TCP 2.2.2.2 : any 1.1.1.1 : any AH Transport {MD5}