

# Firewalls

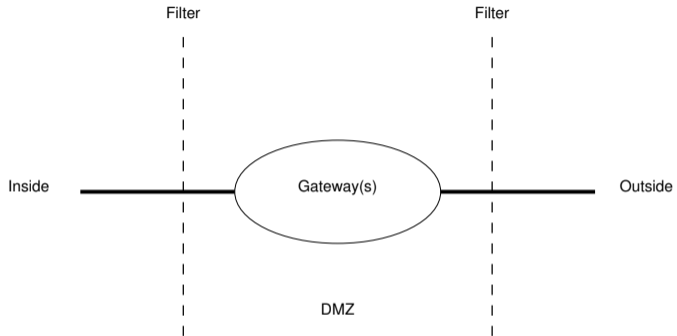


# Introduction

# What's a Firewall

- Barrier between *us* and *them*.
- Limits communication to the outside world.
- 👉 The outside world can be another part of the same company.
- Only a very few machines exposed to attack.

# Schematic of a Firewall



# Why Use Firewalls?

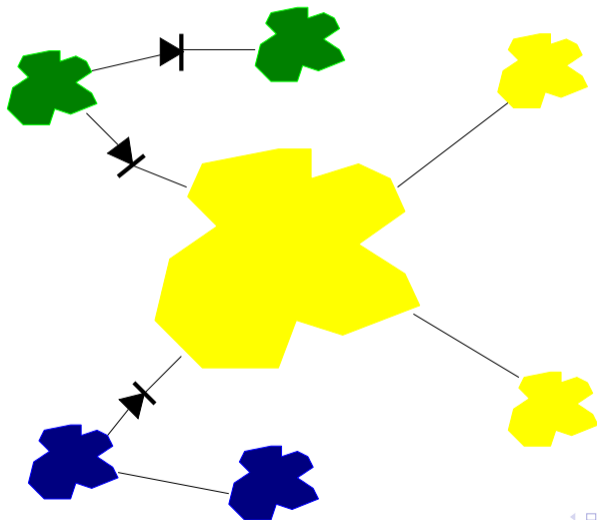
- Most hosts have security holes.  
Proof: Most software is buggy. Therefore, most security software has security bugs.
- Firewalls run much less code, and hence have few bugs (and holes).
- Firewalls can be professionally (and hence better) administered.
- Firewalls run less software, with more logging and monitoring.
- They enforce the partition of a network into separate security domains.
- *Without such a partition, a network acts as a giant virtual machine, with an unknown set of privileged and ordinary users.*

# Should We Fix the Network Protocols?

- Network security is not the problem.
- Firewalls are *not* a solution to network problems. They are a network response to a host security problem.
- More precisely, they are a response to the dismal state of software engineering; taken as a whole, the profession does not know how to produce software that is secure, correct, and easy to administer.
- Consequently, better network protocols will not obviate the need for firewalls. The best cryptography in the world will not guard against buggy code.
- That said, we need to engineer—and deploy—better security protocols.

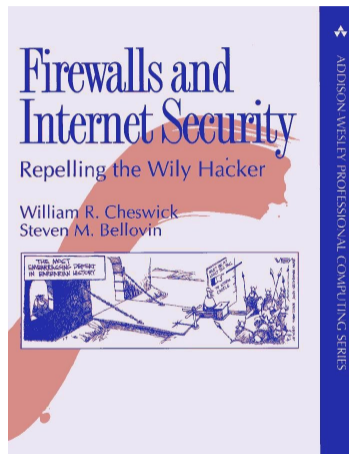
# Positioning Firewalls

Firewalls protect *administrative* divisions.



# However...

- Firewalls are not new
- The preceding slides were taken verbatim from a talk I gave in **1994**—all I did was a bit of reformatting
- What is the modern perspective?
- What are firewalls good for? What don't they do? Do we still need them?



Cheswick and Bellovin, 1994



# 25 Years Ago

- Laptops were rare
- WiFi and hotel broadband were non-existent
- You connected via a dial-up modem, and logged in to a remote shell account to read your email
- There were few, if any, corporate links to other companies over the Internet
- Even the Web was very new

- A large company has hundreds or thousands of links that bypass the firewall
- There's far more telecommuting, and many more services than email are needed
- Laptops, tablets, and smartphones are ubiquitous
- Often, employees use their own equipment (“Bring Your Own Device”)
- Do firewalls still help?
- Yes, but not as much, and even then only if used properly

# How They Work

- Using metadata, identify the protocol or application
- Identify the outside and inside hosts
- Identify the state of the connection, if any
- Make a yes/no decision, depending on policy
- Optional: filter content

# Types of Firewalls

- Packet filters
- Application gateways
- Circuit gateways
- Stateful packet filters

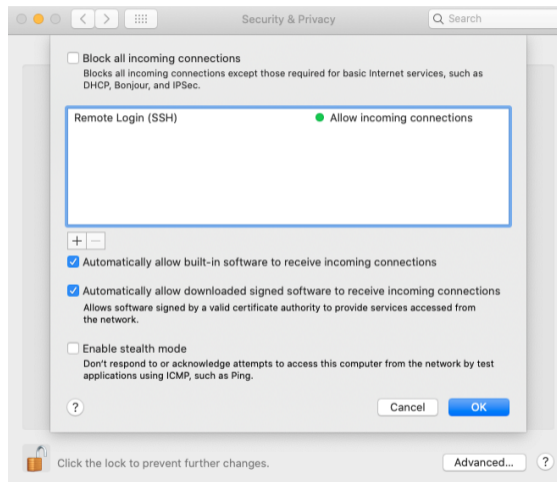
- Look only at the IP and TCP header fields
- Example: permit any host to connect to port 25 on the mail gateway
- Example: permit all outbound connections by looking for the ACK bit in the TCP header, since the only packet that doesn't have it set is the original SYN from the originating client
- Ability to do this built into most routers since the late 1980s, i.e., since the dawn of dedicated routers

# Application Gateways

- Understands and terminates application protocols
- Much better for content-scanning—easier to look at more than a single packet at a time
- Example: email gateway
  - Use a DNS MX record to declare a particular host to be the inbound email gateway for your domain
- Example: some outbound web gateways, to scan incoming HTML for nasty stuff
- Plus: many complex applications, e.g., VoIP
- Also: host-resident “personal firewalls”

# “Personal” Firewalls

- Application running on a computer
- Looks at the *program* involved in the connection, rather than the network metadata






# Circuit Gateways

- Rare today for general-purpose firewalls
- SSH can forward TCP connections or act as a socks proxy

# “Stateful” Packet Filters

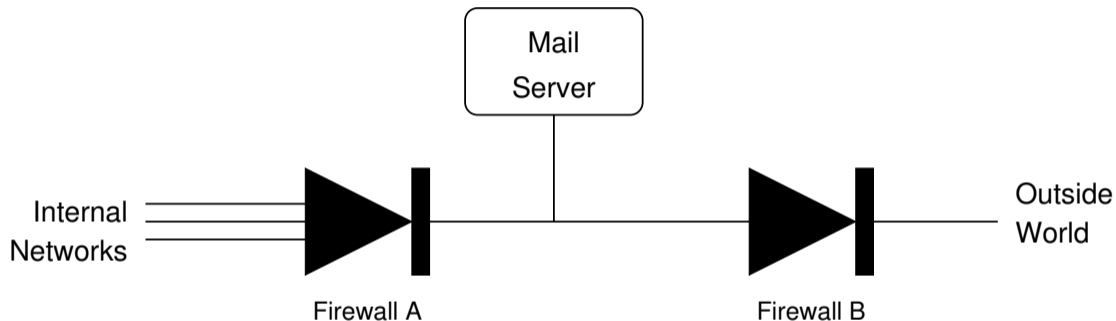
- Similar to ordinary packet filters but keeps state
  - That is, it knows which connections exist, and can't be fooled by forged packets with an ACK bit set
  - Works for UDP, too where there are no connections—set a timeout to allow for return packets
  - (Also handles certain error packets better)
-  Most enterprise firewalls are a combination stateful packet filter/application firewall

# Complex Applications and Application Gateways

- Sophisticated applications have complex semantics; simple packet filters or stateful packet filters cannot provide sufficient security
- Example: a “transfer call” request for a VoIP session
  - The firewall application has to validate the request, perhaps do authentication, and ensure that the endpoints are valid
  - For that matter, a VoIP call setup message describes other connections that have to be opened for voice and/or video
- But: if the firewall is running so much complex code, does that make it insecure?

- Firewalls aren't absolute—some things need to pass through
- Example: inbound email and web
- But—you don't want your entire site hacked if one of these services is hacked
- Solution: a *demilitarized zone* (DMZ)

# A Mail Server in the DMZ



Firewall B protects the mail server from the outside. Firewall A protects the inside from the mail server.

# Do We Need Firewall B?

- What services run on the mail server?
- If it's only inbound SMTP, what does the firewall buy us?
- It may not be necessary.

# What About the IMAP Server?

- Inbound email must be transferred to an IMAP server for retrieval by mail clients
- Where does the IMAP server live? It holds valuable past emails.
- On the inside net? Should the exposed SMTP server be allowed to call in to it?
- On the outside net? It allows employees to check work email from their phones—but that exposes it to outside attack
- No perfect answer—see the next section

# Airgaps

- Some networks need to be *strongly* isolated from the outside world
- Solution: airgaps
- But—airgaps aren't perfect
- How does new code—OS patches, antivirus, updated applications—get in?
- How do you get results out?
- Flash drives? Those have been vectors for attacks on airgapped systems, i.e., the Iranian centrifuge plant at Natanz
- Airgaps can help, but they demand a great deal of process and employee discipline





# Firewall Policies

- Firewalls are *policy enforcement tools*
- Their benefit is at most the benefit from the policies the firewall is enforcing (but buggy code and firewall bypasses mean it could be less)
- What are good policies?
- Where do they come from?
- How are they distributed?

- Is your attitude default-deny or default-allow?
- (Who made that decision?)
- How are proposed exceptions evaluated?
- By whom?

- Do you assume that services are safe until shown otherwise?
- Or do you assume that anything can be dangerous?
- The latter is safer—but the essential issue is the differing cost-benefit ratio for different organizations

- Barring known security issues, the cost of allowing a service is the *risk* of possible flaws in the code or in administration of the system
- In some sense, this is unknowable, but there are often good proxies: the complexity of the service, the reputation of the product, and the reputation of the developers
- Example: Adobe Flash—now thankfully about extinct—has *always* been a horrible security risk
- But: it's possible for companies to turn things around— Windows security used to be a joke, but it's quite good now
- Also: does the service itself allow bypassing of other enterprise policies? (Many sites bar Dropbox for that reason)

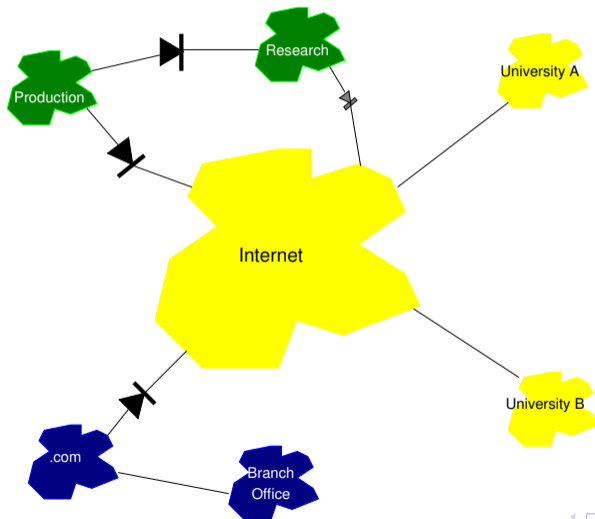
- The service itself may be a benefit
- In a tech company, it's always good for staff to see other tech products
- Employee morale can matter, too

# Organizational Needs and Culture

- Organizations have their own cultures—a university will and should have looser security than a corporation
- A freewheeling research lab needs more freedom to explore than the group developing your next product
- *You can't enforce stronger security than the culture will accept*
- Internal firewalls are very frequently necessary

# Firewall Policies

Note that Research has a “smaller”—less stringent—firewall than Production.





# Why is the Research Firewall Separate?

- A single corporate firewall would need a more complex rule set
- The firewall protecting Production from Research would also need more complex rules
- Also: who administers those firewalls? Is the administrative and coordination load too much?
- But a smaller company could rationally opt for a single link

# Approving Policies

- Requests for changes to the firewall should be accompanied by a rationale and a risk analysis
- Evaluation should be done by a combination of security people, system administrators, and *line management*
- Again: it is not a purely security decision—the business needs matter, too

# Tracking Changes

- It's important to track changes—some should have a finite lifespan
- Record: who approved it, when, why, and when the question should be revisited
- Use `git` to track firewall ruleset changes
- Reminder: treat firewall rules like code changes: test the changes not only to make sure you allow what you now want to allow, but that you block what you want to block. . .

- In an enterprise setting, policies need to be set centrally
- They then have to be distributed to all firewalls



This includes personal firewalls

- N.B.: That implies that individual users should not be able to change such policies on their computers
- Distribution needs to be tracked

# Sample Simple Policies

- Allow inbound mail and web
- Allow most outbound web—but not to porn sites
- Block a known spammer
- Permit DNS queries
- A joint venture partner can access a shared internal web server

# Sample Ruleset

<i>Action</i>	<i>Proto</i>	<i>Src Host</i>	<i>Src Port</i>	<i>Dest Host</i>	<i>Dest Port</i>	
block	TCP	spammer	*	mail-gw	25	Block a spammer
pass	TCP	*	*	mail-gw	25	Allow inbound mail to gw
pass	*	*	53	dns	53	Allow DNS queries
pass	TCP	*	*	www	80	Anyone can talk to
pass	TCP	*	*	www	443	our web server
block	TCP	*	*	porn	80	No porn at work
block	TCP	*	*	porn	443	No porn at work
pass	TCP	*	*	*	80	Allow outbound HTTP
pass	TCP	*	*	*	443	Allow outbound HTTPS
pass	TCP	partner	*	joint-web	443	Joint venture access
block	*	*	*	*	*	Block everything else

Note: rules, like ACLs, are order-sensitive

# Using Firewalls

# A Theory of Firewalls

There are three properties necessary for a firewall to be effective:

- 1 The firewall must be placed at a topological chokepoint
- 2 The nodes on the “inside” must generally share the same security policy
- 3 All of the nodes on the inside must be “good”; all nodes on the outside are “bad” or perhaps merely untrusted



# None of these Properties Hold Today

- 1 There are too many links through or around the firewall; there are also mobile devices that wander back and forth
- 2 There are too many different kinds of computers inside; their security needs are very different
- 3 There are so many internal nodes that some are very likely to be infected at any given time; in addition, there are mobile devices

But—firewalls can work in special cases where these three properties hold

# A Single Machine

- Imagine a firewall between the network stack and the applications (many personal firewalls work that way)—clearly, all three properties hold
- Where does the policy come from?
- If a central administrator can ship out (good enough) policies, we can (mostly) replace the traditional firewall
- One problem: how does this machine know which other machines are trusted?

# Departmental Firewalls

- For small-enough organizations—perhaps a small company, or a department of a large one—a traditional firewall can provide some useful protection
- Departments rarely have rich connectivity, so Property 1 is satisfied (except for mobile devices—to where do they connect?)
- Within a department, policy is usually pretty straightforward, thus satisfying Property 2
- By definition, the number of machines inside a departmental firewall is comparatively small, satisfying Property 3
- Good for protecting low-value resources or as supplemental protection for the departmental fileserver

# Point Firewalls

- We can generalize this: put a small, simple firewall (e.g., a packet filter) in front of a resource that can't protect itself, or that needs extra protection
- Example: a networked printer
- (Policy is simple: prevent outsiders from using up the paper and toner.)
- Example: protect everything *but* ports 80 and 443 on a Web server

# What Firewalls Can't Do

- Firewalls provide no protection at other layers of the stack than the ones at which they operate
- 👉 Packet filters are lousy at virus-scanning; mail proxies can't detect ARP-spoofing
- Firewalls rarely do a good job filtering permitted protocols
- 👉 Why should a firewall be better at spotting nasty HTTP than the Web server itself is? Once, firewall code was small and simple, but that's no longer true, especially for enterprise-grade firewalls
- Comprehensive firewalls need components at all layers of the stack, but this adds to their complexity

# Modern Firewall Philosophy

- Classic firewalls have limited, but non-zero, utility
- The need for enforcement of central policy has not changed
- The need to keep the bad guys away from buggy code has not changed
- We need new styles of firewall, new ways to manage them, and new ways to protect mobile devices

# Questions?



(Scarlet tanager, Riverside Park, May 8, 2019)