Chapter 23: Network Security

• Introduction to the Drib
• Policy Development
• Network Organization
• Availability
• Anticipating Attacks
Introduction

• Goal: apply concepts, principles, mechanisms discussed earlier to a particular situation
  – Focus here is on securing network
  – Begin with description of company
  – Proceed to define policy
  – Show how policy drives organization
The Drib

• Builds and sells dribbles
• Developing network infrastructure allowing it to connect to Internet to provide mail, web presence for consumers, suppliers, other partners
Specific Problems

• Internet presence required
  – E-commerce, suppliers, partners
  – Drib developers need access
  – External users cannot access development sites

• Hostile takeover by competitor in progress
  – Lawyers, corporate officers need access to development data
  – Developers cannot have access to some corporate data
Goals of Security Policy

• Data related to company plans to be kept secret
  – Corporate data such as what new products are being developed is known on a need-to-know basis only

• When customer supplies data to buy a dribble, only folks who fill the order can access that information
  – Company analysts may obtain statistics for planning

• Lawyers, company officials must approve release of any sensitive data
Policy Development

• Policy: minimize threat of data being leaked to unauthorized entities

• Environment: 3 internal organizations
  – Customer Service Group (CSG)
    • Maintains customer data
    • Interface between clients, other internal organizations
  – Development Group (DG)
    • Develops, modifies, maintains products
    • Relies on CSG for customer feedback
  – Corporate Group (CG)
    • Handles patents, lawsuits, etc.
Nature of Information Flow

• Public
  – Specs of current products, marketing literature
• CG, DG share info for planning purposes
  – Problems, patent applications, budgets, etc.
• Private
  – CSG: customer info like credit card numbers
  – CG: corporate info protected by attorney privilege
  – DG: plans, prototypes for new products to determine if production is feasible before proposing them to CG
Data Classes

• Public data (PD): available to all
• Development data for existing products (DDEP): available to CG, DG only
• Development data for future products (DDFP): available to DG only
• Corporate data (CpD): available to CG only
• Customer data (CuD): available to CSG only
Data Class Changes

• **DDFP → DDEP**: as products implemented
• **DDEP → PD**: when deemed advantageous to publicize some development details
  – For marketing purposes, for example
• **CpD → PD**: as privileged info becomes public through mergers, lawsuit filings, etc.
• **Note**: no provision for revealing CuD directly
  – This protects privacy of Drib’s customers
User Classes

- Outsiders (O): members of public
  - Access to public data
  - Can also order, download drivers, send email to company
- Developers (D): access to DDEP, DDFP
  - Cannot alter development data for existing products
- Corporate executives (C): access to CD
  - Can read DDEP, DDFP, CuD but not alter them
  - Sometimes can make sensitive data public
- Employees (E): access to CuD only
# Access Control Matrix for Policy

<table>
<thead>
<tr>
<th></th>
<th>O</th>
<th>D</th>
<th>C</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD</td>
<td>$r$</td>
<td>$r$</td>
<td>$r$</td>
<td>$r$</td>
</tr>
<tr>
<td>DDEP</td>
<td></td>
<td>$r$</td>
<td>$r$</td>
<td></td>
</tr>
<tr>
<td>DDFP</td>
<td></td>
<td>$r$, $w$</td>
<td>$r$</td>
<td></td>
</tr>
<tr>
<td>CpD</td>
<td></td>
<td></td>
<td>$r$, $w$</td>
<td></td>
</tr>
<tr>
<td>CuD</td>
<td>$w$</td>
<td></td>
<td>$r$</td>
<td>$r$, $w$</td>
</tr>
</tbody>
</table>

$r$ is read right, $w$ is write right
Type of Policy

• Mandatory policy
  – Members of O, D, C, E cannot change permissions to allow members of another user class to access data

• Discretionary component
  – Within each class, individuals may have control over access to files they own
  – View this as an issue internal to each group and not of concern at corporate policy level
    • At corporate level, discretionary component is “allow always”
Reclassification of Data

• Who must agree for each?
  – C, D must agree for DDFP $\rightarrow$ DDEP
  – C, E must agree for DDEP $\rightarrow$ PD
  – C can do CpD $\rightarrow$ PD
    • But two members of C must agree to this

• Separation of privilege met
  – At least two different people must agree to the reclassification
  – When appropriate, the two must come from different user classes
Availability

• Drib world-wide multinational corp
  – Does business on all continents
• Imperative anyone be able to contact Drib at any time
  – Drib places very high emphasis on customer service
  – Requirement: Drib’s systems be available 99% of the time
    • 1% allowed for planned maintenance, unexpected downtime
Consistency Check: Goal 1

- Goal 1: keep sensitive info confidential
  - Developers
    - Need to read DDEP, DDFP, and to alter DDFP
    - No need to access CpD, CuD as don’t deal with customers or decide which products to market
  - Corporate executives
    - Need to read, alter CpD, and read DDEP
- This matches access permissions
Consistency Check: Goal 2

- Goal 2: only employees who handle purchases can access customer data, and only they and customer can alter it
  - Outsiders
    - Need to alter CuD, do not need to read it
  - Customer support
    - Need to read, alter CuD
  - This matches access permissions
Consistency Check: Goal 3

- Goal 3: releasing sensitive info requires corporate approval
  - Corporate executives
    - Must approve any reclassification
    - No-one can write to PD, except through reclassification
- This matches reclassification constraints
### Consistency Check: Transitive Closure

<table>
<thead>
<tr>
<th></th>
<th>O</th>
<th>D</th>
<th>C</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD</td>
<td>$r$</td>
<td>$r$</td>
<td>$r$</td>
<td>$r$</td>
</tr>
<tr>
<td>DDEP</td>
<td>$r$</td>
<td></td>
<td>$r$</td>
<td></td>
</tr>
<tr>
<td>DDFP</td>
<td>$r$, $w$</td>
<td></td>
<td>$r$</td>
<td></td>
</tr>
<tr>
<td>CpD</td>
<td>$w$</td>
<td></td>
<td>$r$, $w$</td>
<td>$w$</td>
</tr>
<tr>
<td>CuD</td>
<td>$w$</td>
<td></td>
<td>$r$</td>
<td>$r$, $w$</td>
</tr>
</tbody>
</table>

$r$ is read right, $w$ is write right
Interpretation

• From transitive closure:
  – *Only* way for data to flow into PD is by reclassification
  – Key point of trust: members of C
  – By rules for moving data out of DDEP, DDFP, someone other than member of C must also approve
    • Satisfies separation of privilege

• Conclusion: policy is consistent
Network Organization

• Partition network into several subnets
  – Guards between them prevent leaks
DMZ

• Portion of network separating purely internal network from external network
  – Allows control of accesses to some trusted systems inside the corporate perimeter
  – If DMZ systems breached, internal systems still safe
  – Can perform different types of checks at boundary of internal, DMZ networks and DMZ, Internet network
Firewalls

• Host that mediates access to a network
  – Allows, disallows accesses based on configuration and type of access

• Example: block Back Orifice
  – BO allows external users to control systems
    • Requires commands to be sent to a particular port (say, 25345)
  – Firewall can block all traffic to or from that port
    • So even if BO installed, outsiders can’t use it
Filtering Firewalls

• Access control based on attributes of packets and packet headers
  – Such as destination address, port numbers, options, etc.
  – Also called a packet filtering firewall
  – Does not control access based on content
  – Examples: routers, other infrastructure systems
Proxy

• Intermediate agent or server acting on behalf of endpoint without allowing a direct connection between the two endpoints
  – So each endpoint talks to proxy, thinking it is talking to other endpoint
  – Proxy decides whether to forward messages, and whether to alter them
Proxy Firewall

• Access control done with proxies
  – Usually bases access control on content as well as source, destination addresses, etc.
  – Also called an applications level or application level firewall
  – Example: virus checking in electronic mail
    • Incoming mail goes to proxy firewall
    • Proxy firewall receives mail, scans it
    • If no virus, mail forwarded to destination
    • If virus, mail rejected or disinfected before forwarding
Views of a Firewall

• Access control mechanism
  – Determines which traffic goes into, out of network

• Audit mechanism
  – Analyzes packets that enter
  – Takes action based upon the analysis
    • Leads to traffic shaping, intrusion response, etc.
Analysis of Drib Network

• Security policy: “public” entities on outside but may need to access corporate resources
  – Those resources provided in DMZ
• No internal system communicates directly with systems on Internet
  – Restricts flow of data to “public”
  – For data to flow out, must pass through DMZ
    • Firewalls, DMZ are “pump”
Implementation

• Conceal all internal addresses
  – Make them all on 10., 172., or 192.168. subnets
    • Inner firewall uses NAT to map addresses to firewall’s address
  – Give each host a non-private IP address
    • Inner firewall never allows those addresses to leave internal network

• Easy as all services are proxied by outer firewall
  – Email is a bit tricky …
Email

• Problem: DMZ mail server must know address in order to send mail to internal destination
  – Could simply be distinguished address that causes inner firewall to forward mail to internal mail server

• Internal mail server needs to know DMZ mail server address
  – Same comment
DMZ Web Server

• In DMZ so external customers can access it without going onto internal network
  – If data needs to be sent to internal network (such as for an order), transmission is made separately and not as part of transaction
Application of Principles

- Least privilege
  - Containment of internal addresses
- Complete mediation
  - Inner firewall mediates every access to DMZ
- Separation of privilege
  - Going to Internet must pass through inner, outer firewalls and DMZ servers
Application of Principles

• Least common mechanism
  – Inner, outer firewalls distinct; DMZ servers separate from inner servers
  – DMZ DNS violates this principle
    • If it fails, multiple systems affected
    • Inner, outer firewall addresses fixed, so they do not depend on DMZ DNS
Outer Firewall Configuration

• Goals: restrict public access to corporate network; restrict corporate access to Internet

• Required: public needs to send, receive email; access web services
  – So outer firewall allows SMTP, HTTP, HTTPS
  – Outer firewall uses its address for those of mail, web servers
Details

• Proxy firewall
• SMTP: mail assembled on firewall
  – Scanned for malicious logic; dropped if found
  – Otherwise forwarded to DMZ mail server
• HTTP, HTTPS: messages checked
  – Checked for suspicious components like very long lines; dropped if found
  – Otherwise, forwarded to DMZ web server
• Note: web, mail servers *different systems*
  – Neither same as firewall
Attack Analysis

• Three points of entry for attackers:
  – Web server ports: proxy checks for invalid, illegal HTTP, HTTPS requests, rejects them
  – Mail server port: proxy checks email for invalid, illegal SMTP requests, rejects them
  – Bypass low-level firewall checks by exploiting vulnerabilities in software, hardware
    • Firewall designed to be as simple as possible
    • Defense in depth
Defense in Depth

• Form of separation of privilege
• To attack system in DMZ by bypassing firewall checks, attacker must know internal addresses
  – Then can try to piggyback unauthorized messages onto authorized packets
• But the rewriting of DMZ addresses prevents this
Inner Firewall Configuration

- Goals: restrict access to corporate internal network
- Rule: block *all* traffic except for that *specifically* authorized to enter
  - Principle of fail-safe defaults
- Example: Drib uses NFS on some internal systems
  - Outer firewall disallows NFS packets crossing
  - Inner firewall disallows NFS packets crossing, too
    - DMZ does not need access to this information (least privilege)
    - If inner firewall fails, outer one will stop leaks, and vice versa (separation of privilege)
More Configuration

• Internal folks require email
  – SMTP proxy required

• Administrators for DMZ need login access
  – So, allow SSH through *provided*:
    • Destination is a DMZ server
    • Originates at specific internal host (administrative host)
  – Violates least privilege, but ameliorated by above

• DMZ DNS needs to know address of administrative host
  – More on this later
DMZ

- Look at servers separately:
  - Web server: handles web requests with Internet
    - May have to send information to internal network
  - Email server: handles email with Internet
    - Must forward email to internal mail server
  - DNS
    - Used to provide addresses for systems DMZ servers talk to
  - Log server
    - DMZ systems log info here
DMZ Mail Server

• Performs address, content checking on all email
• Goal is to hide internal information from outside, but be transparent to inside
• Receives email from Internet, forwards it to internal network
• Receives email from internal network, forwards it to Internet
Mail from Internet

• Reassemble messages into header, letter, attachments as files
• Scan header, letter, attachments looking for “bad” content
  – “Bad” = known malicious logic
  – If none, scan original letter (including attachments and header) for violation of SMTP spec
• Scan recipient address lines
  – Address rewritten to direct mail to internal mail server
  – Forward letter there
Mail to Internet

• Like mail from Internet with 2 changes:
  – Step 2: also scan for sensitive data (like proprietary markings or content, etc.)
  – Step 3: changed to rewrite all header lines containing host names, email addresses, and IP addresses of internal network
    • All are replaced by “drib.org” or IP address of external firewall
Administrative Support

• Runs SSH server
  – Configured to accept connections *only* from trusted administrative host in internal network
  – All public keys for that host fixed; no negotiation to obtain those keys allowed
  – Allows administrators to configure, maintain DMZ mail host remotely while minimizing exposure of host to compromise
DMZ Web Server

- Accepts, services requests from Internet
- Never contacts servers, information sources in internal network
- CGI scripts checked for potential attacks
  - Hardened to prevent attacks from succeeding
  - Server itself contains no confidential data
- Server is www.drib.org and uses IP address of outer firewall when it must supply one
Updating DMZ Web Server

- Clone of web server kept on internal network
  - Called “WWW-clone”
- All updates done to WWW-clone
  - Periodically admins copy contents of WWW-clone to DMZ web server
- DMZ web server runs SSH server
  - Used to do updates as well as maintenance, configuration
  - Secured like that of DMZ mail server
Internet Ordering

• Orders for Drib merchandise from Internet
  – Customer enters data, which is saved to file
  – After user confirms order, web server checks format, content of file and then uses public key of system on internal customer subnet to encipher it
    • This file is placed in a spool area not accessible to web server program
  – Original file deleted
  – Periodically, internal trusted administrative host uploads these files, and forwards them to internal customer subnet system
Analysis

- If attacker breaks into web server, cannot get order information
  - There is a slight window where the information of customers still on system can be obtained
- Attacker can get enciphered files, public key used to encipher them
  - Use of public key cryptography means it is computationally infeasible for attacker to determine private key from public key
DMZ DNS Server

- Supplies DNS information for some hosts to DMZ:
  - DMZ mail, web, log hosts
  - Internal trusted administrative host
    - Not fixed for various reasons; could be …
  - Inner firewall
  - Outer firewall
- Note: Internal server addresses not present
  - Inner firewall can get them, so DMZ hosts do not need them
DMZ Log Server

• DMZ systems all log information
  – Useful in case of problems, attempted compromise
• Problem: attacker will delete or alter them if successful
  – So log them off-line to this server
• Log server saves logs to file, also to write-once media
  – Latter just in case log server compromised
• Runs SSH server
  – Constrained in same way server on DMZ mail server is
Summary

• Each server knows only what is needed to do its task
  – Compromise will restrict flow of information but not reveal info on internal network

• Operating systems and software:
  – All unnecessary features, servers disabled
  – Better: create custom systems

• Proxies prevent direct connection to systems
  – For all services except SSH from internal network to DMZ, which is itself constrained by source, destination
Internal Network

- Goal: guard against unauthorized access to information
  - “read” means fetching file, “write” means depositing file
- For now, ignore email, updating of DMZ web server, internal trusted administrative host
- Internal network organized into 3 subnets, each corresponding to Drib group
  - Firewalls control access to subnets
Internal Mail Server

• Can communicate with hosts on subnets
• Subnet may have mail server
  – Internal DNS need only know subnet mail server’s address
• Subnet may allow mail to go directly to destination host
  – Internal DNS needs to know addresses of all destination hosts
• Either satisfies policy
WWW-close

- Provides staging area for web updates
- All internal firewalls allow access to this
  - WWW-clone controls who can put and get what files and where they can be put
- Synchronized with web pages on server
  - Done via internal trusted administrative host
- Used as testbed for changes in pages
  - Allows corporate review before anything goes public
  - If DMZ web server trashed or compromised, all web pages can be restored quickly
Trusted Administrative Host

- Access tightly controlled
  - Only system administrators authorized to administer DMZ systems have access
- All connections to DMZ through inner firewall must use this host
  - Exceptions: internal mail server, possibly DNS
- All connections use SSH
  - DMZ SSH servers accept connections from this host only
Analysis

• DMZ servers never communicate with internal servers
  – All communications done via inner firewall
• Only client to DMZ that can come from internal network is SSH client from trusted administrative host
  – Authenticity established by public key authentication
• Only data non-administrative folks can alter are web pages
  – Even there, they do not access DMZ
Analysis

- Only data from DMZ is customer orders and email
  - Customer orders already checked for potential errors, enciphered, and transferred in such a way that it cannot be executed
  - Email thoroughly checked before it is sent to internal mail server
Assumptions

• Software, hardware does what it is supposed to
  – If software compromised, or hardware does not work right, defensive mechanisms fail
  – Reason separation of privilege is critical
    • If component A fails, other components provide additional defenses

• Assurance is vital!
Availability

- Access over Internet must be unimpeded
  - Context: flooding attacks, in which attackers try to overwhelm system resources

- Example: SYN flood
  - Problem: server cannot distinguish legitimate handshake from one that is part of this attack
    - Only difference is whether third part of TCP handshake is sent
  - Flood can overwhelm communication medium
    - Can’t do anything about this (except buy a bigger pipe)
  - Flood can overwhelm resources on our system
    - We start here
Intermediate Hosts

• Use routers to divert, eliminate illegitimate traffic
  – Goal: only legitimate traffic reaches firewall
  – Example: Cisco routers try to establish connection with source (TCP intercept mode)
    • On success, router does same with intended destination, merges the two
    • On failure, short time-out protects router resources and target never sees flood
Intermediate Hosts

• Use network monitor to track status of handshake
  – Example: synkill monitors traffic on network
    • Classifies IP addresses as not flooding (good), flooding (bad), unknown (new)
    • Checks IP address of SYN
      – If good, packet ignored
      – If bad, send RST to destination; ends handshake, releasing resources
      – If new, look for ACK or RST from same source; if seen, change to good; if not seen, change to bad
    • Periodically discard stale good addresses
Intermediate Hosts

• Problem: don’t solve problem!
  – They move the locus of the problem to the intermediate system
  – In Drib’s case, Drib does not control these systems

• So, consider endpoints
Endpoint Hosts

• Control how TCP state is stored
  – When SYN received, entry in queue of pending connections created
    • Remains until an ACK received or time-out
    • In first case, entry moved to different queue
    • In second case, entry made available for next SYN

  – In SYN flood, queue is always full
    • So, assure legitimate connections space in queue to some level of probability
    • Two approaches: SYN cookies or adaptive time-outs
SYN Cookies

• Source keeps state
• Example: Linux 2.4.9 kernel
  – Embed state in sequence number
  – When SYN received, compute sequence number to be function of source, destination, counter, and random data
    • Use as reply SYN sequence number
    • When reply ACK arrives, validate it
  – Must be hard to guess
Adaptive Time-Out

- Change time-out time as space available for pending connections decreases
- Example: modified SunOS kernel
  - Time-out period shortened from 75 to 15 sec
  - Formula for queueing pending connections changed:
    - Process allows up to \( b \) pending connections on port
    - \( a \) number of completed connections but awaiting process
    - \( p \) total number of pending connections
    - \( c \) tunable parameter
    - Whenever \( a + p > cb \), drop current SYN message
Anticipating Attacks

- Drib realizes compromise may come through unanticipated means
  - Plans in place to handle this
- Extensive logging
  - DMZ log server does intrusion detection on logs
Against Outer Firewall

- Unsuccessful attacks
  - Logged, then ignored
  - Security folks use these to justify budget, train new personnel
- Successful attack against SMTP proxy
  - Proxy will start non-standard programs
  - Anomaly detection component of IDS on log server will report unusual behavior
    - Security officers monitor this around the clock
In the DMZ

• Very interested in attacks, successful or not
• Means someone who has obtained access to DMZ launched attack
  – Some trusted administrator shouldn’t be trusted
  – Some server on outer firewall is compromised
  – Software on DMZ system not restrictive enough
• IDS system on DMZ log server looks for misuse (known attacks) to detect this
Ignoring Failed Attacks

• Sounds dangerous
  – Successful attacker probably tried and failed earlier

• Drib: “So what?”
  – Not sufficient personnel to handle all alerts
  – Focus is on what Drib cares most about
    • Successful attacks, or failed attacks where there should be none
Checking the IDS

- IDS allows Drib to add attack signatures and tune parameters to control reporting of events
  - Experimented to find good settings
  - Verify this every month by doing manual checks for two 1-hour periods (chosen at random) and comparing with reported events
Key Points

• Begin with policy
• Craft network architecture and security measures from it
• Assume failure will occur
  – Try to minimize it
  – Defend in depth
  – Have plan to handle failures