

Security - Case Study

Note Title

2/22/2011

Announcements

- Midterm will be March 3
(expect review sheet by Thursday)
- Accept resubmissions of paper 1 before
Spring break
- Paper 2 due Thursday

Testing Lab 2: Netcat

```
client$ man nc
```

```
...
```

```
The nc (or netcat) utility is used for just about anything under the sun involving TCP or UDP. It can open TCP connections, send UDP packets, listen on arbitrary TCP and UDP ports, do port scanning, and deal with both IPv4 and IPv6.
```

Test #1:

Determine if the server is allowing inbound & outbound ~~traffic on port 80~~

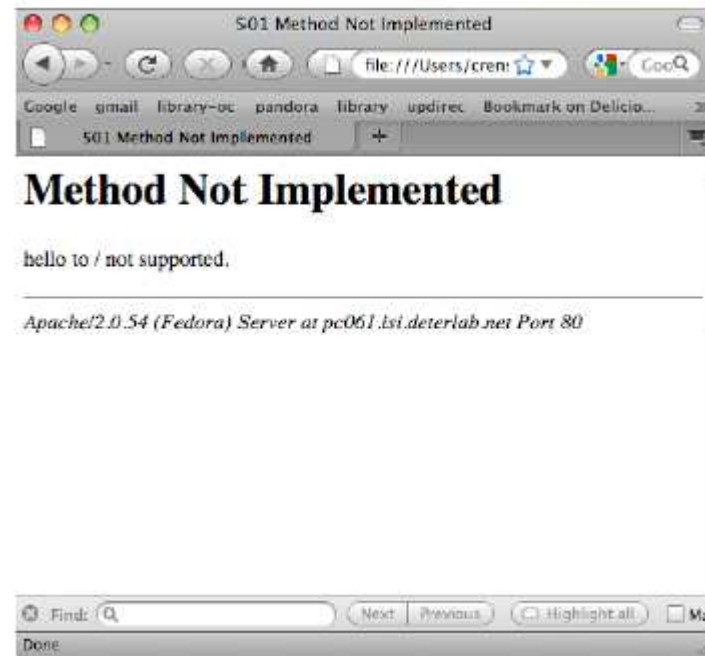
Step 1: Create a file to send

```
client$ echo "hello" > hi.txt
client$ more hi.txt
hello
client$
```

Step 2: Make a request to port 80

```
client$ nc server.CS448Lab2.UP-CS448.isi.deterlab.net 80 < hi.txt
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>501 Method Not Implemented</title>
</head><body>
<h1>Method Not Implemented</h1>
<p>hello to / not supported.<br />
</p>
<hr>
<address>Apache/2.0.54 (Fedora) Server at pc061.isi.deterlab.net
Port 80</address>
</body></html>
```

Equivalent to:



Alternative: get
(makes a well formed request)

```
GET /index.html HTTP/1.1
```

```
Host: server.CS448Lab2.UP-CS448.isi.deterlab.net
```

Test 2:

Check if server is allowing inbound or outbound traffic on 8080

Step 1: Open a port for listening on the server

```
server$ nc -l 80
nc: already in use
server$ nc -l 8080 > output.txt
```

Step 2: Use the client to send data

```
client$ more hi.txt  
hello  
client$ nc server.CS448Lab2.UP-  
CS448.isi.deterlab.net 8080 < hi.txt
```

```
server$ nc -l 8080 > output.txt  
server$ more output.txt  
hello
```


Networking Case Study: Dribbble, Inc.

Company with the following goals:

- Company plans must be kept secret
- Customer data should be available only to those who fill the order
- However, company analysts may use customer data for statistics
- Releasing sensitive data requires consent of company officials

Policy development

- We need to go from these goals to concrete policies and design.

Specify:

- users
- data classes
- organization of network
- firewall permissions

First principles

① Principle of Least Privilege

A subject should only be given privileges necessary to complete its task.

First principles

② Principle of Open design

Security of our system should
not depend on secrecy.

First principles

③ Principle of Separation of Privilege:

A system should not grant permission based on a single condition

access rights - before data is released, must be approved by multiple people with appropriate access rights.

First principles

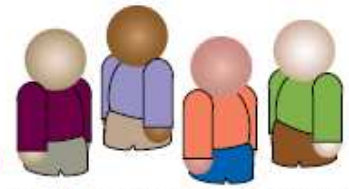
④ Principle of Fail-Safe Defaults:

Unless subject is given explicit rights,
default behavior is to deny access.

Internal Organization - 3 groups

Internal Organization

fill the →
orders



Customer Service Group (CSG)
maintains customer data, interfaces between groups and clients



Development Group (DG)
develops, modifies, and maintains products



access to,
everything

Corporate Group (CG)
handles lawsuits, patents, corporate-level work

Internal Organization - Data



Public Data (PD)
available to anyone; product specifications, price information and marketing literature.



Development Data for Existing Products (DDEP)
available only internally to lawyers and developers.



Development Data for Future Products (DDFP)
available to developers only.



Corporate Data (CpD)
available only to corporate officials and lawyers; privileged information about corporate actions.



Customer Data (CuD)
data supplied by customers, including credit card information.

Data Classes

restricted version

How should we move data around?

Control over flow

User Classes

Who can access what types of data?



Outsiders

members of the public may get access to prices, product descriptions, public corporate information, new drivers, and e-mail addresses.

(only public info)



Developers

allowed access to both classes of development data.



Corporation Executives

allowed access to corporate data; they may view both classes of development data; they may read customer data.



Employees

get access to customer data only.

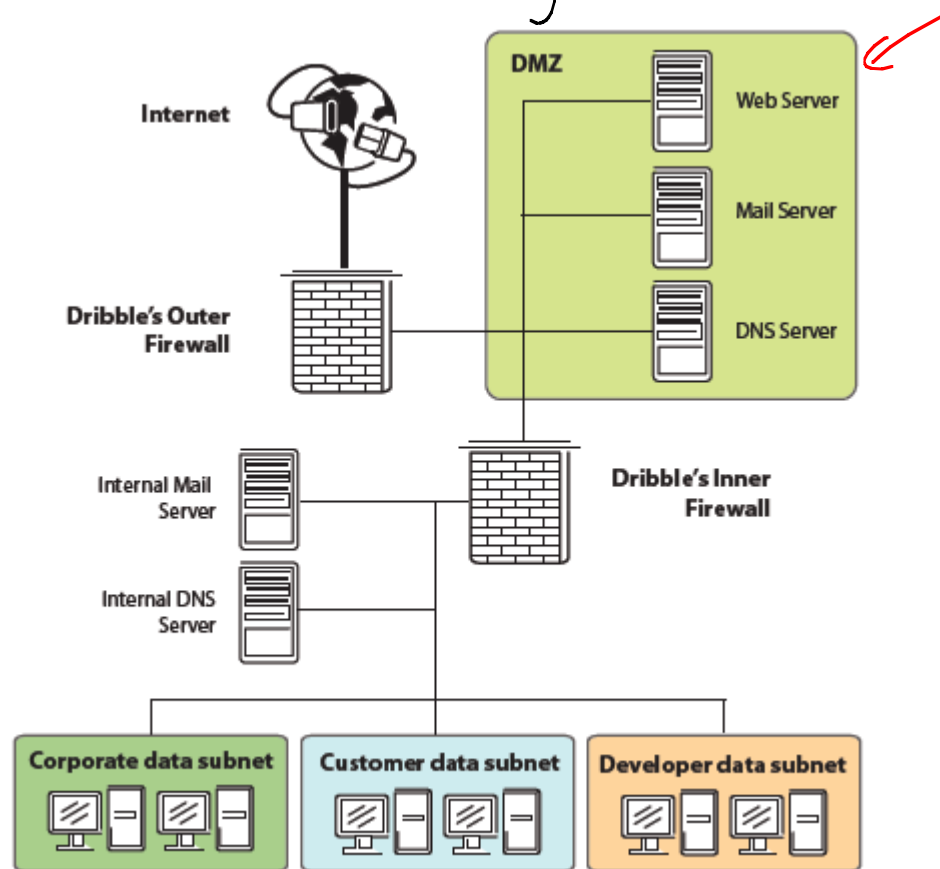
- fill orders (+ public data)



Access Control Matrix

	Outsiders	Developers	Corporation Executives	Employees
Public Data	read	read	read	read
Development Data for Existing Products		read	read	
Development Data for Future Products		read, write	read	
Corporate Data			read, write	
Customer Data	write		read	read, write

Network of the company



Outer Firewall

Goal:

No read
up

- Restrict public access to corporate network

No write
down

- Restrict employee access to the internet

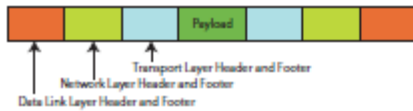
What should be allowed?

- Email

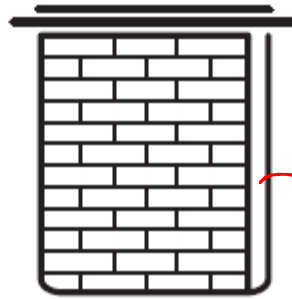
- Web presence

Public Access:

drop everything else



Mail from the Internet



Outer Firewall (proxy firewall)



Mail Server (SMTP proxy)

Web and mail only

Inner Firewall

Blocks everything with a few exceptions:

- Allows SMTP connections using proxies, but only if routed to DMZ mail server
- Allows sys admins to access DMZ via trusted internal server
locked-up + locked down

Which principle? fail safe defaults

Admin Connection

Uses SSH from trusted server.

- Inner firewall only allows ssh connections to DMZ servers

- SSH is on a physically secure, trusted machine, with public-key cryptography used at both ends

Inner and Outer Firewalls work together:

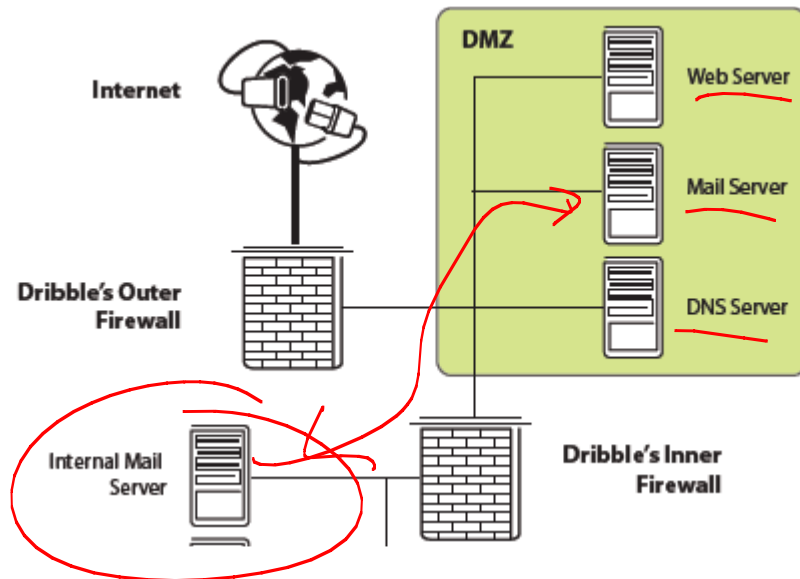
For example, if company uses NFS to share files, the outer firewall disallows those packets from leaving.

In addition, the inner firewall restricts them from entering the DMZ.

DMZ:

4 servers:

- mail: checks address and content
- WWW: accepts and services requests
- DNS: tracks domain name info for all hosts that the DMZ servers must know
- log server: admin logging



A closer look:

DMZ mail server:

- ① Reassembles messages into header, letter, and attachments
- ② Scan letters and attachments
- ③ Destination addresses are rewritten to internal mail server.

Also has SSH server, for trusted admin.

Closer Look: Web Server

- Runs SSH server for admin.
- Accepts and services requests from the internet - including merchandise orders.

Care is needed here, since we want to protect and isolate customer data.

(All traffic will be encrypted.)

Once decrypted:

① Save data to a file.

② Once order is confirmed, the web server invokes a program to check this file.

③ Encipher this file using public key of a system on the internal customer subnet.

On this system, save encrypted version in an area not accessible by web server.

④ Delete the file!

Why this approach?


Defend the data!

Job of DMZ server is simply to check and pass data along.

Since DMZ is "vulnerable", this server is not safe for long term storage.

DMZ DNS Server

Store domain name info for :

- DMZ mail, web, log
- Internal trusted admin host
- Outer firewall 
- Inner firewall

DMZ DNS Server

Note: Does not know the internal mail server.

Why?

(Hint: Principle of least privilege)

Rely on inner firewall, so DNS doesn't need it.

DMZ log Server

Performs admin logging on network.

While these may be compromised, they also can help track attacks.

→ Might also incorporate intrusion detection or other components.

Summary of DMZ

Each server has the minimum knowledge necessary to perform its task.

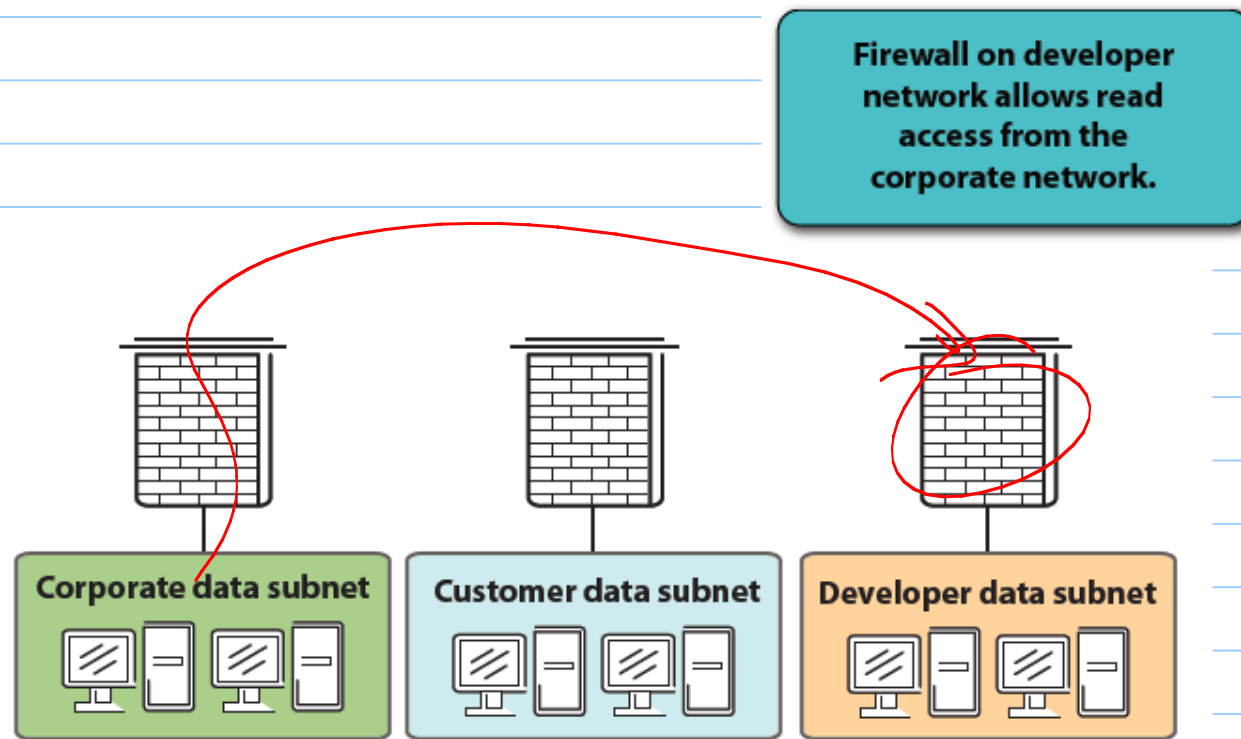
[Each computer will have minimum number of things running.

Why?

- Speed

- Less vulnerabilities

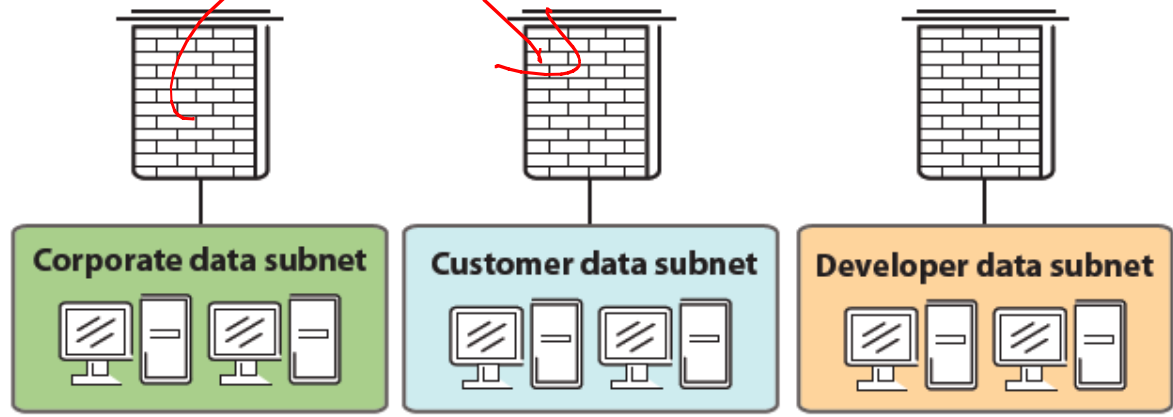
Internal Networks



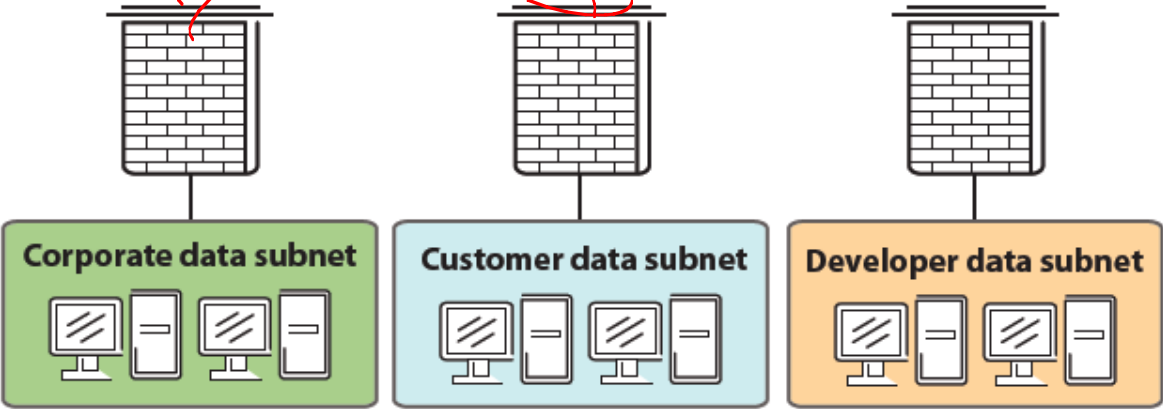
Firewall on customer network allows read access from the corporate network.

Firewall on customer network also allows writes from DMZ Web server

(no read access)



Firewall on corporate network does not allow read or writes from anywhere.



Internal "DMZ"

The network effectively has a second secure DMZ:

