# Service-Oriented Requirement Analysis and Security Design

1

## Planning for Security

- Security never happens by accident
- *80%* of the users on the network have *100%* access
  - Most users don't think about security
- Reducing scope of access reduces chance of harm:
  - Accidental
  - Deliberate

2

## An Observation

- For some reason
  - Security
  - Robustness against outages
  - Performance

  … all appear to be properties of well-designed networks
- Low cost is not
  - Over long term a good network is *cheaper*

3

## Goals

- This is an attempt to establish a process for reasoning about the security of an Intranet
  - Audit plans
  - Growth plans
  - Connectivity mapping
- *Warning:* It is potentially very time-consuming :(

4

## A Service-Oriented View

- Think of the network as an environment of services to pick and choose from *not* as a mass of packets or protocols
  - Not TCP/IP specific
- Don't think in terms of connecting networks

*... Think in terms of connecting applications*

5

## Security Domains

- More than one security domain may exist within a single set of network wiring
- Each domain is an isolated "network"
  - Physically isolated and firewalled
  - Virtually isolated using hubs
  - Coarsely isolated using router filters
  - Virtually isolated using software

6

## Network Connections

- Identify security domains
  - All members of secure domain *must* share common security policy and practices
  - All entry into secure domain *must* be approved on a case-by-case basis
  - All new connections *must* be approved on a case-by-case basis
  - Periodically audit domain for integrity

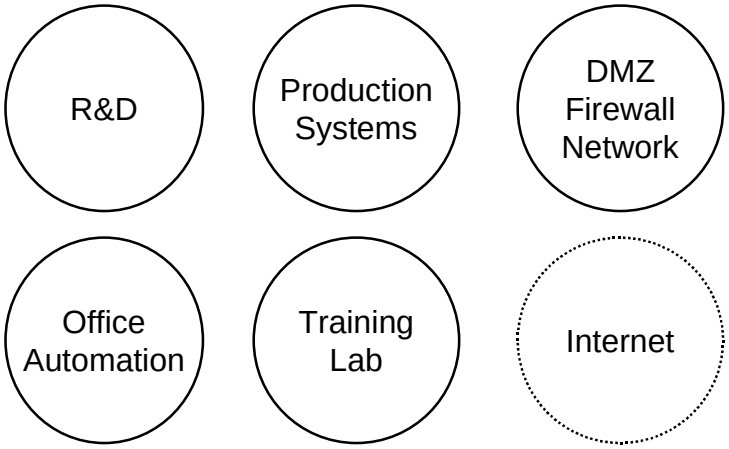
7

## Map out the Domains

- For each domain, build a matrix of what types of access are required with other domains
- Specify access in terms of:
  - Incoming
  - Outgoing
  - Whether to an approved host
  - Special service

8

# Example Domains




9

# Domain Map

| | R&D | Prod | DMZ | Auto | Trn | Inet |
|---|---|---|---|---|---|---|
| R&D | 1 | 2*,3 | 1 | 2,3 | 0 | 1 |
| Production | 3 | 1 | 0 | 3,*X* | 0 | 0 |
| DMZ | 3 | 0 | 1 | 0 | 0 | 1 |
| Automation | 3 | 3 | 0 | 1 | 0 | 4 |
| Training | 0 | *X** | 0 | 0 | 1 | 4 |
| Internet | 0 | 0 | *X** | 3* | 0 | 1 |

0 = No access
1 = Full access
3 = Email

*X* = Special service
2 = Authenticated telnet, FTP
4 = WWW *= Logged

10

## Service Justification

- Adding new inter-domain services:
  - Provide business justification
  - Provide statement of security impact
  - Provide proposed means of addressing security concerns if necessary
  - Provide maintenance procedures
  - Completely fill in the matrix
    - This forces a “sanity check” against all other domain policies

11

## Inter-Domain Services

- For specified services identify security issues and document how they are addressed
  - This amounts to attaching a risk assessment to each matrix cell
  - Couple risk assessment to the appropriate proposed response to the risk
  - Include maintenance / review plan
  - Assign responsibility / accountability

12

## Inter-Domain Services *(cont)*

- Only permit new services between domain when there is a business need
  - This entails performing a risk assessment for that service
  - Couple risk assessment to the business justification
  - This provides an enterprise-wide blueprint for how to securely provide a given service

13

## Inter-Domain Services *(cont)*

- Appropriate security gets put in place for each service
  - To maximize leverage use the same solution wherever possible
    - Apply same risk assessment
    - Sometimes apply same technology solution
    - Sometimes apply same maintenance plan
- Even if the matrix is big the number of services should be small

14

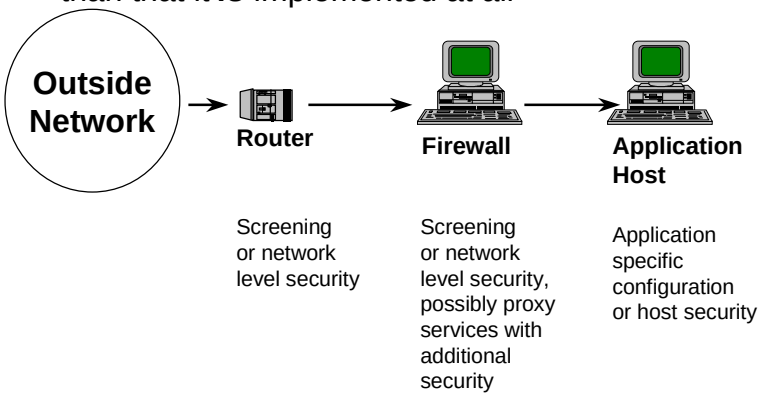
# Adding a New Domain

- Adding a new domain to the matrix entails a policy review vis-a-vis existing connectivity
  - Many organizations don't perform this type of review
  - Result is often surprisingly high connectivity to networks that should be more isolated

15

# Service Security

It is less important ***where*** security is implemented than that it ***is*** implemented at all




16

## Typical Services

- Ignore implementation details and list high-level services that are required
- A typical set of core services:
  - Email
  - Web
  - FTP
  - USENET news
  - Telnet/Rlogin

17

## Service Oriented Firewalls

- A firewall is a system that exists between multiple domains for a service
- Often a firewall's physical system will gateway more than one service
- A single firewall may exist in *all* domains effectively becoming a corporate service hub

18

## Example Firewall

- Suppose the company is building a data repository
  - Requirements are strong authentication access and access from all over the enterprise
- Put the repository where all domains can reach it for the data service and then secure the data service

19

## Example Firewall *(cont)*

- Sample implementation:
  - A company-wide POP server
  - A company-wide web server that uses SSL to protect all data access
  - A company-wide database engine that uses secure SQL access for all queries
- Access is restricted using router screening to a single TCP port

20

## Example Firewall *(cont)*

- Sample implementation:
  - A company-wide DNS server which is a lightweight system running *only* nameserver software
  - A shared web site between R&D and production that allows engineers to post bugfixes and upgrades using SSL

21

## Virtual Networks

- Eventual hope is to take advantage of virtual network capabilities
  - Port Switching hubs
  - LAN switches
  - LAN switches with encryption capabilities
  - Routers that provide crypto-tunneling
- Domains map to virtual networks

22

## Summary

- Most sites perform no organized analysis before adding connections
- Most sites perform no policy review prior to deploying applications
- Most sites have more connectivity than they need
- Security requires a clean architecture that is maintained in an orderly manner

23