

INF3510 Information Security

University of Oslo

Spring 2010

Lecture 13

Application Security and Trust Management



Audun Jøsang

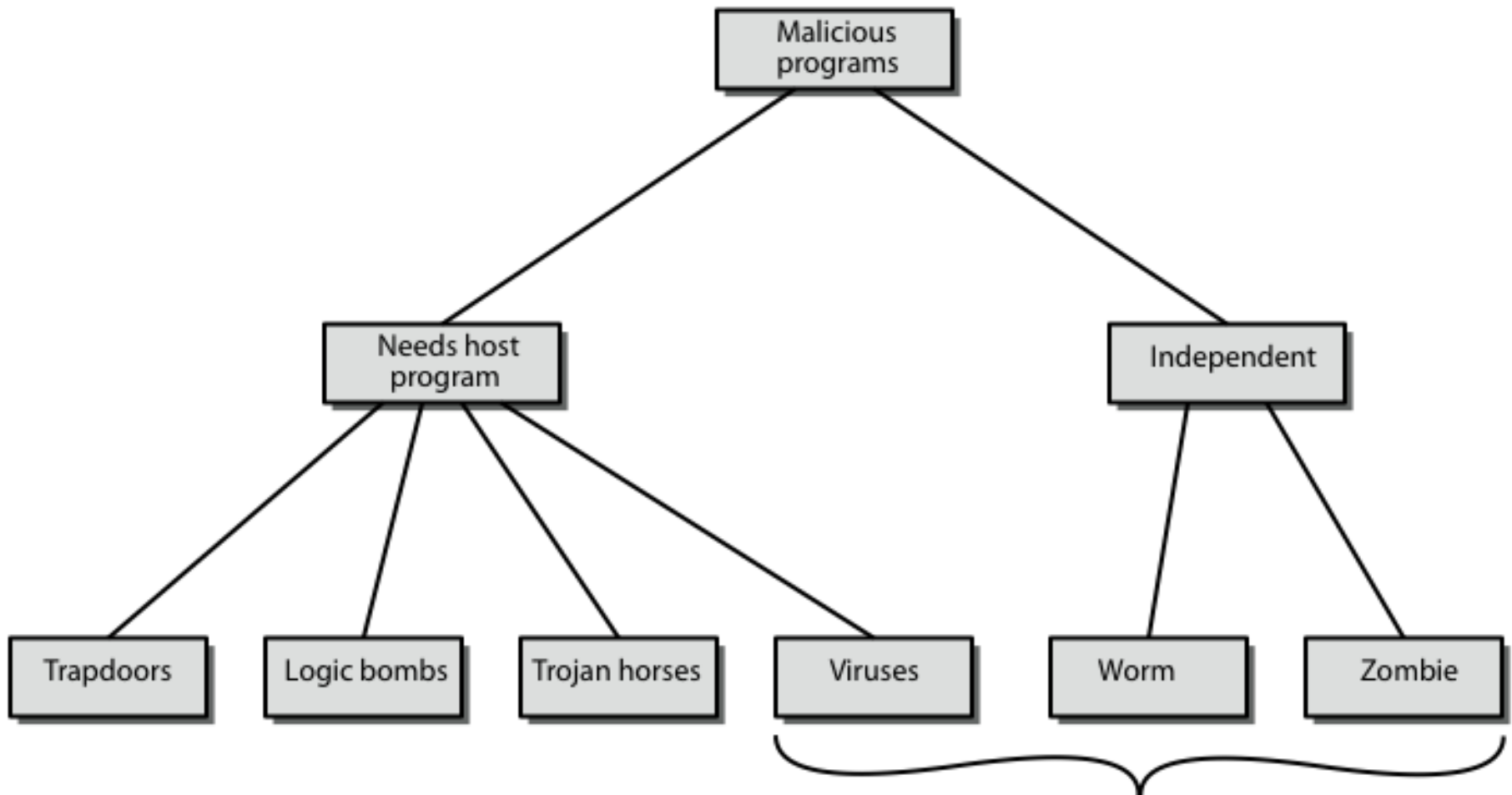
Outline

- Malicious Software
 - various malicious programs
 - trapdoor, logic bomb, trojan horse, zombie
 - viruses
 - worms
 - distributed denial of service attacks
- Attacks on applications
 - Buffer overflos
 - SQL Injection
 - Cross-Site Scripting
- Trust Management

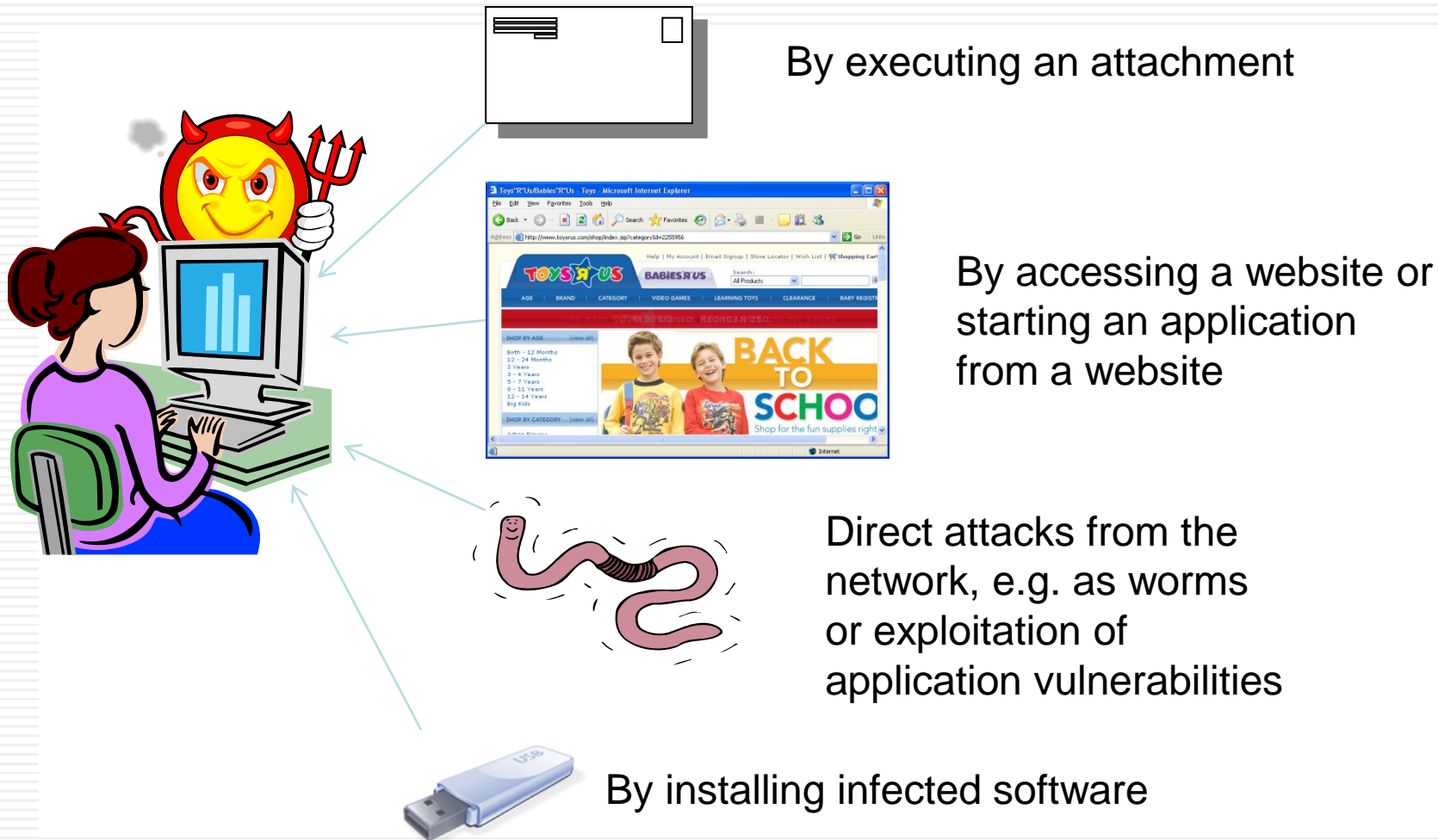
Malware: Malicious Content

- Malware receive a lot of publicity
- Many different forms
- Many different effects
- Difficult to know when infected
- More advanced forms emerge
- A growing concern

Malicious Software



How do you get infected



Backdoor or Trapdoor

- secret entry point into a program
- allows those who know access bypassing usual security procedures
- have been commonly used by developers
- a threat when left in production programs allowing exploited by attackers
- very hard to block in O/S
- requires good s/w development & update

Logic Bomb

- one of oldest types of malicious software
- code embedded in legitimate program
- activated when specified conditions met
 - eg presence/absence of some file
 - particular date/time
 - particular user
- when triggered typically damage system
 - modify/delete files/disks, halt machine, etc

Trojan Horse

- program with hidden side-effects
- which is usually superficially attractive
 - eg game, s/w upgrade etc
- when run performs some additional tasks
 - allows attacker to indirectly gain access they do not have directly
- often used to propagate a virus/worm or install a backdoor
- or simply to destroy data

Mobile Code

- program/script/macro that runs unchanged
 - on heterogeneous collection of platforms
 - on large homogeneous collection (Windows)
- transmitted from remote system to local system & then executed on local system
- often to inject virus, worm, or Trojan horse
- or to perform own exploits
 - unauthorized data access, root compromise

Multiple-Threat Malware

- malware may operate in multiple ways
- **multipartite** virus infects in multiple ways
 - eg. multiple file types
- **blended** attack uses multiple methods of infection or transmission
 - to maximize speed of contagion and severity
 - may include multiple types of malware
 - eg. Nimda has worm, virus, mobile code
 - can also use IM & P2P

Viruses

- piece of software that infects programs
 - modifying them to include a copy of the virus
 - so it executes secretly when host program is run
- specific to operating system and hardware
 - taking advantage of their details and weaknesses
- a typical virus goes through phases of:
 - dormant
 - propagation
 - triggering
 - execution

Virus Structure

- components:
 - infection mechanism - enables replication
 - trigger - event that makes payload activate
 - payload - what it does, malicious or benign
- prepended / postpended / embedded
- when infected program invoked, executes virus code then original program code
- can block initial infection (difficult)
- or propagation (with access controls)

Virus Classification

- boot sector
- file infector
- macro virus
- encrypted virus
- stealth virus
- polymorphic virus
- metamorphic virus

Macro Virus

- became very common in mid-1990s since
 - platform independent
 - infect documents
 - easily spread
- exploit macro capability of office apps
 - executable program embedded in office doc
 - often a form of Basic
- more recent releases include protection
- recognized by many anti-virus programs

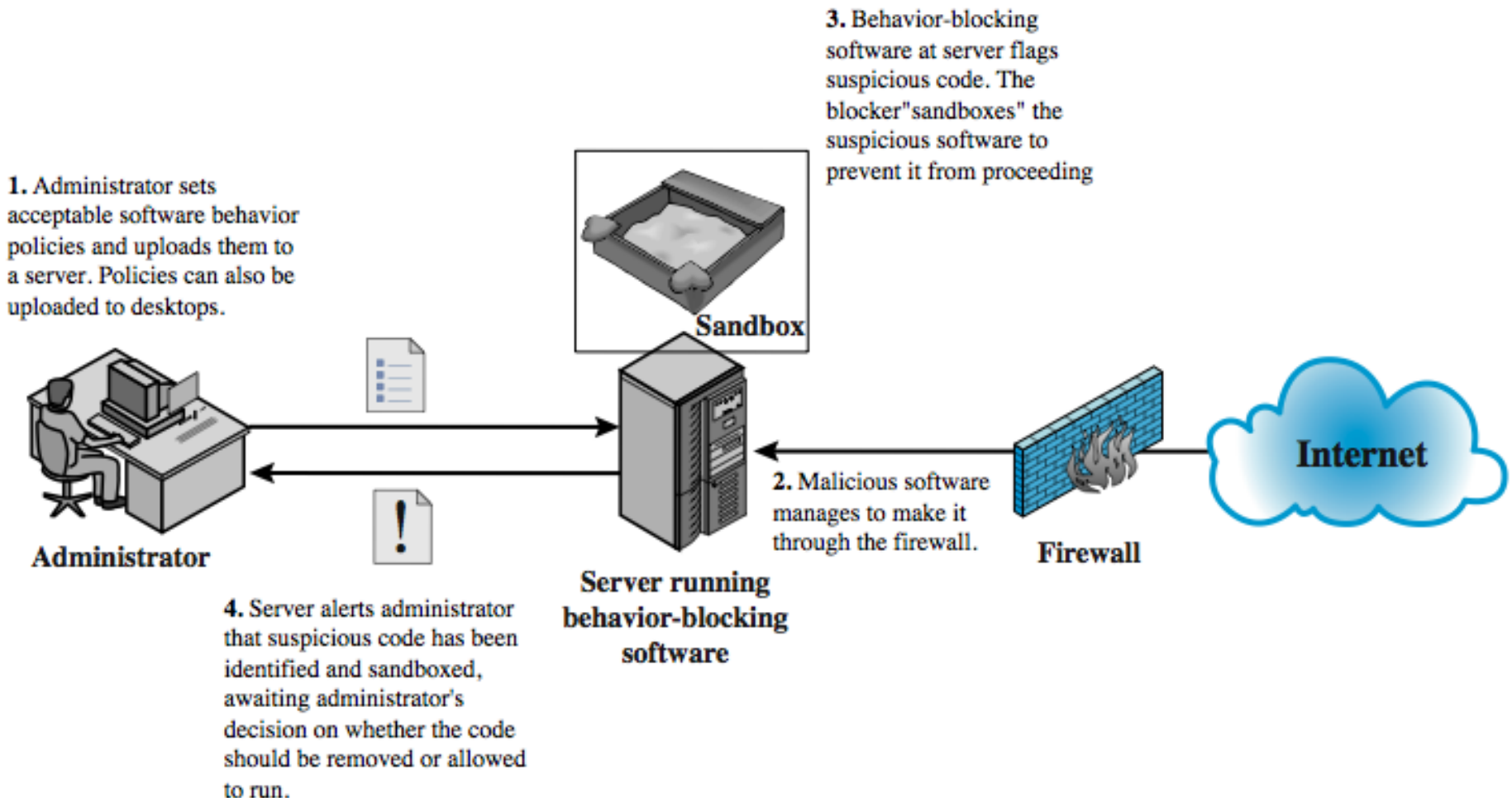
E-Mail Viruses

- more recent development
- e.g. Melissa
 - exploits MS Word macro in attached doc
 - if attachment opened, macro activates
 - sends email to all on users address list
 - and does local damage
- then saw versions triggered reading email
- hence much faster propagation

Virus Countermeasures

- prevention - ideal solution but difficult
- realistically need:
 - detection
 - identification
 - removal
- if detect but can't identify or remove, must discard and replace infected program, or reformat hard drive

Behavior-Blocking Software



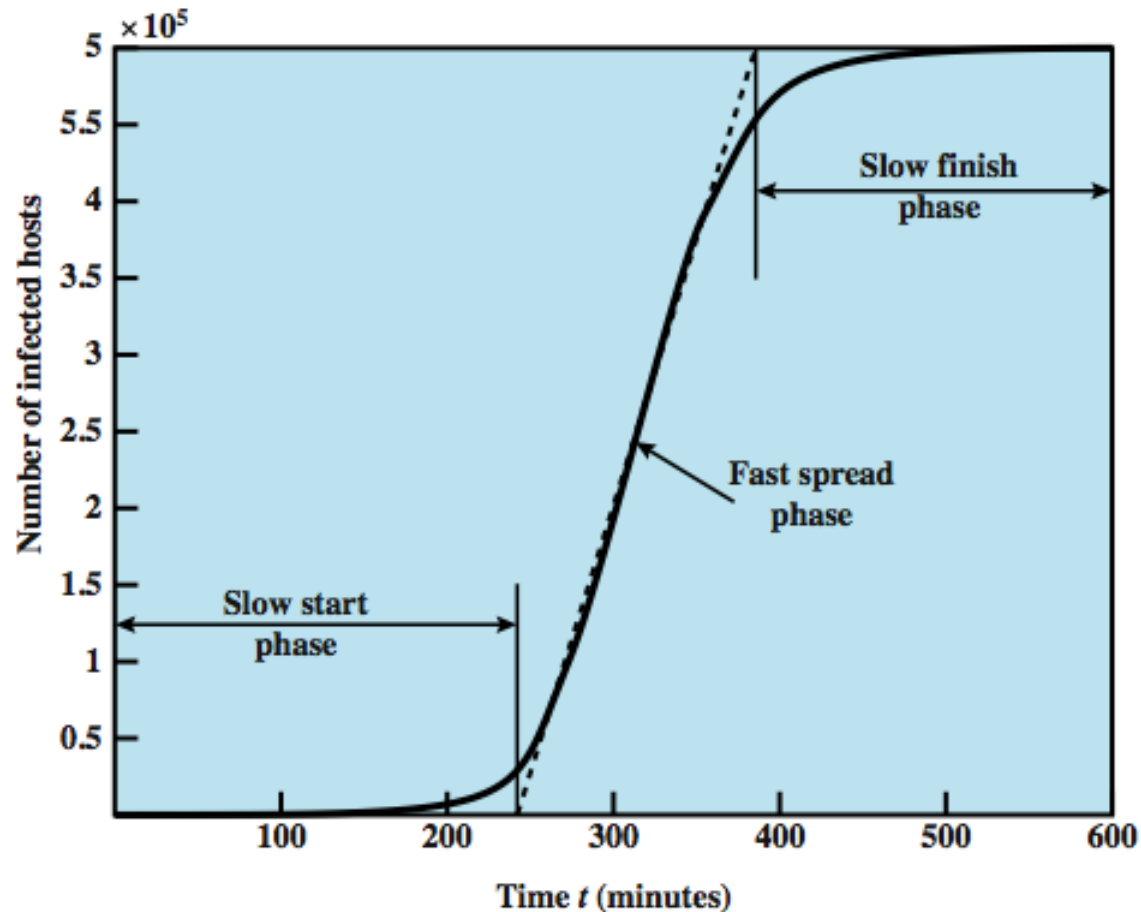
Worms

- replicating program that propagates over net
 - using email, remote exec, remote login
- has phases like a virus:
 - dormant, propagation, triggering, execution
 - propagation phase: searches for other systems, connects to it, copies self to it and runs
- may disguise itself as a system process
- concept seen in Brunner's "Shockwave Rider"
- implemented by Xerox Palo Alto labs in 1980's

Morris Worm

- one of best know worms
- released by Robert Morris in 1988
- various attacks on UNIX systems
 - cracking password file to use login/password to logon to other systems
 - exploiting a bug in the finger protocol
 - exploiting a bug in sendmail
- if succeed have remote shell access
 - sent bootstrap program to copy worm over

Worm Propagation Model



Recent Worm Attacks

- Code Red
 - July 2001 exploiting MS IIS bug
 - probes random IP address, does DDoS attack
- Code Red II variant includes backdoor
- SQL Slammer
 - early 2003, attacks MS SQL Server
- Mydoom
 - mass-mailing e-mail worm that appeared in 2004
 - installed remote access backdoor in infected systems
- Warezov family of worms
 - scan for e-mail addresses, send in attachment

Worm Technology

- multiplatform
- multi-exploit
- ultrafast spreading
- polymorphic
- metamorphic
- transport vehicles
- zero-day exploit

Mobile Phone Worms

- first appeared on mobile phones in 2004
 - target smartphone which can install s/w
- they communicate via Bluetooth or MMS
- to disable phone, delete data on phone, or send premium-priced messages
- CommWarrior, launched in 2005
 - replicates using Bluetooth to nearby phones
 - and via MMS using address-book numbers

Worm Countermeasures

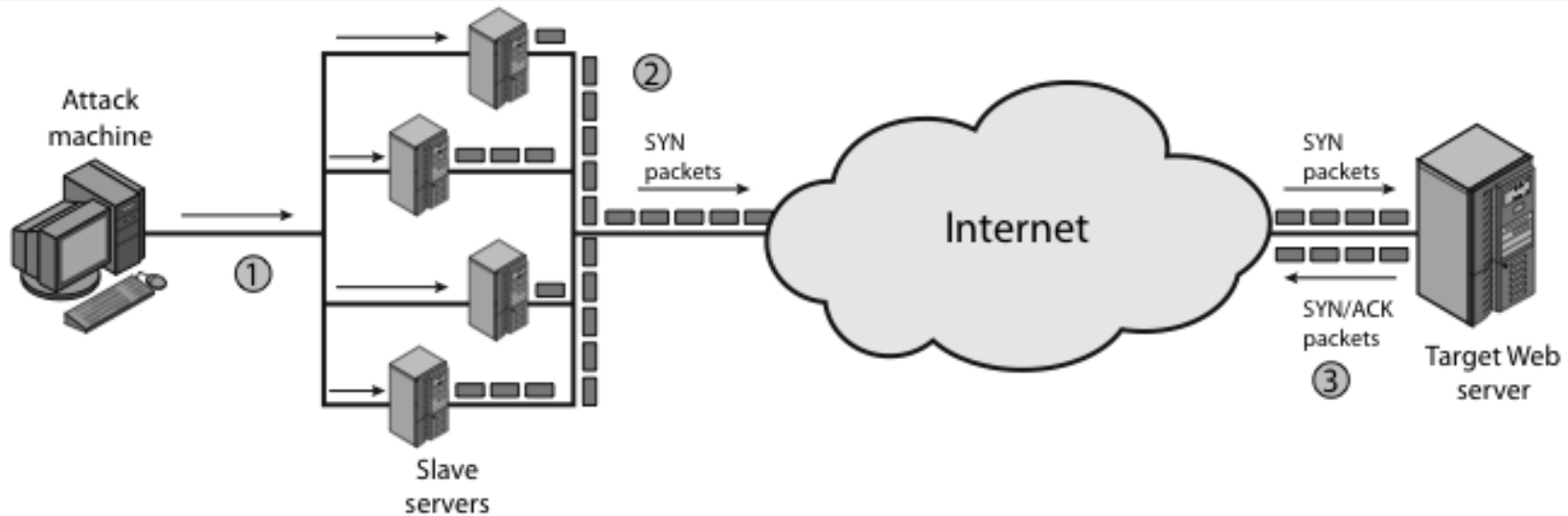
- overlaps with anti-virus techniques
- once worm on system A/V can detect
- worms also cause significant net activity
- worm defense approaches include:
 - signature-based worm scan filtering
 - filter-based worm containment
 - payload-classification-based worm containment
 - threshold random walk scan detection
 - rate limiting and rate halting

DDoS

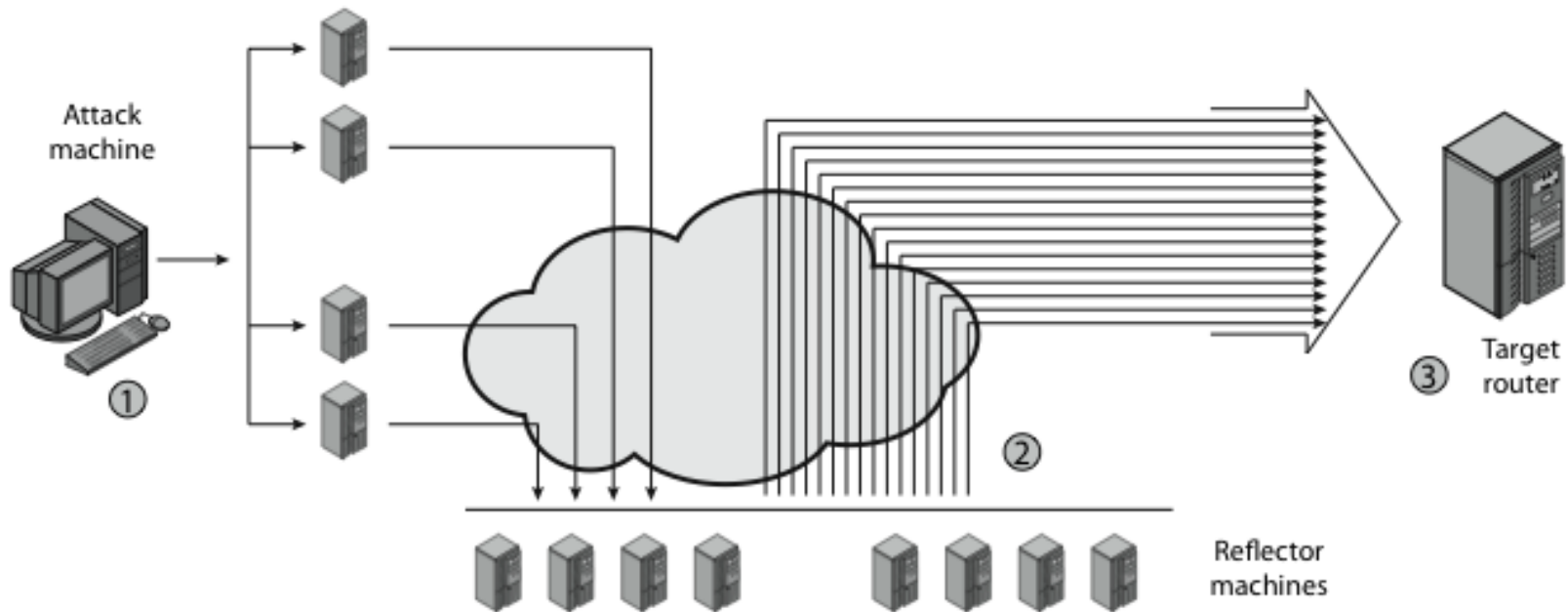
Distributed Denial of Service Attacks

- Distributed Denial of Service (DDoS) attacks form a significant security threat
- making networked systems unavailable
- by flooding with useless traffic
- using large numbers of “zombies”
- growing sophistication of attacks
- defense technologies struggling to cope

Distributed Denial of Service Attack

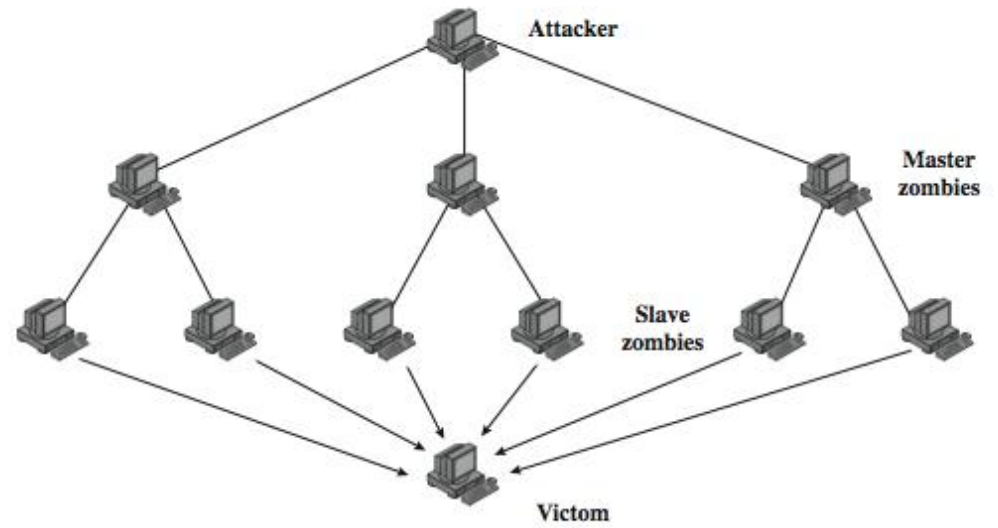


(a) Distributed SYN flood attack

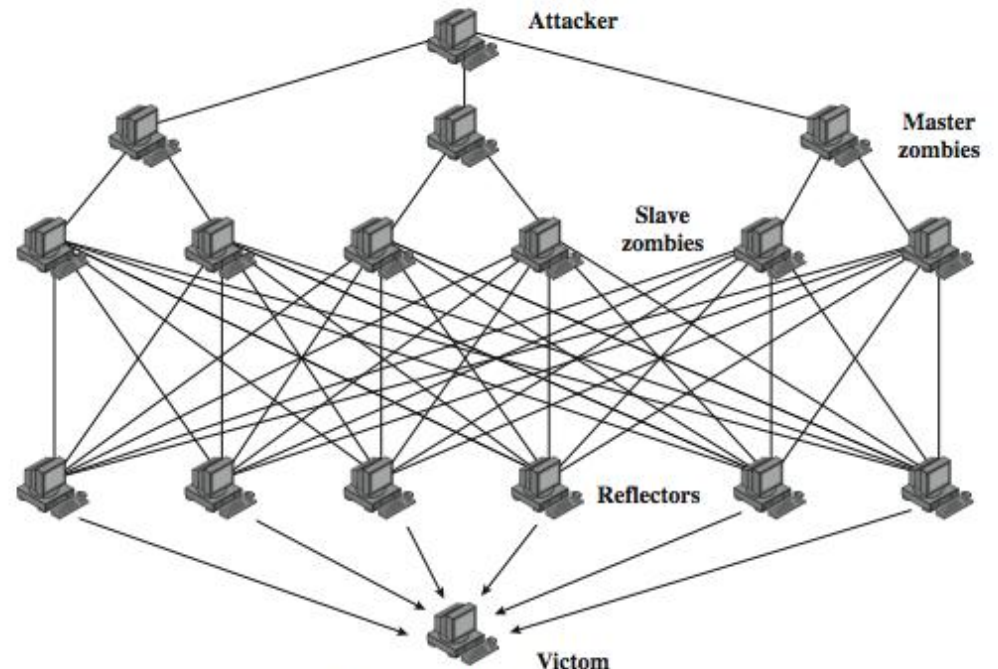


(a) Distributed ICMP attack

DDoS Flood Types



(a) Direct DDoS Attack



(b) Reflector DDoS Attack

Constructing an Attack Network

- must infect large number of zombies
- needs:
 1. software to implement the DDoS attack
 2. an unpatched vulnerability on many systems
 3. scanning strategy to find vulnerable systems
 - random, hit-list, topological, local subnet

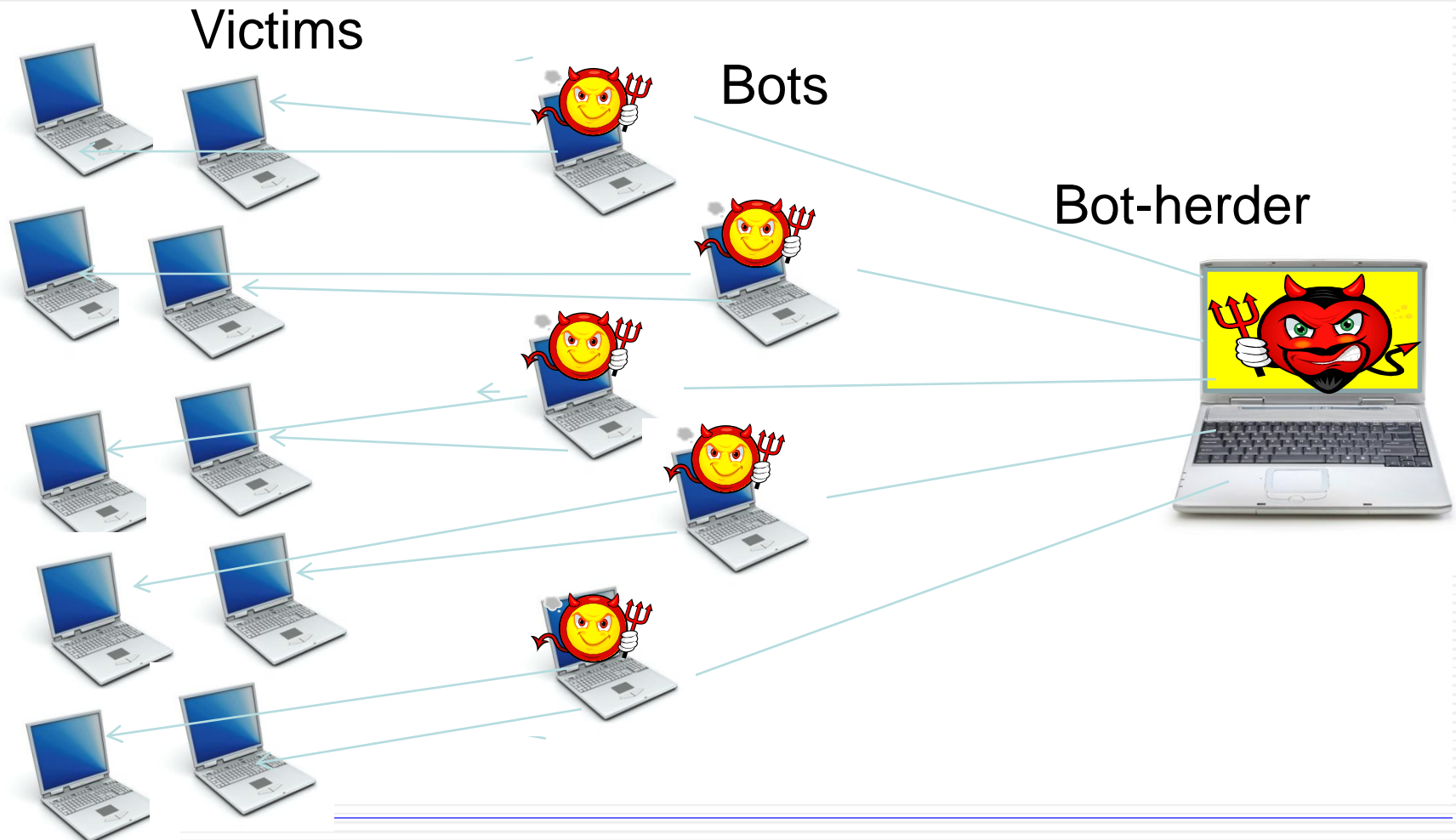
DDoS Countermeasures

- three broad lines of defense:
 1. attack prevention & preemption (before)
 2. attack detection & filtering (during)
 3. attack source traceback & ident (after)
- huge range of attack possibilities
- hence evolving countermeasures

What is a botnet

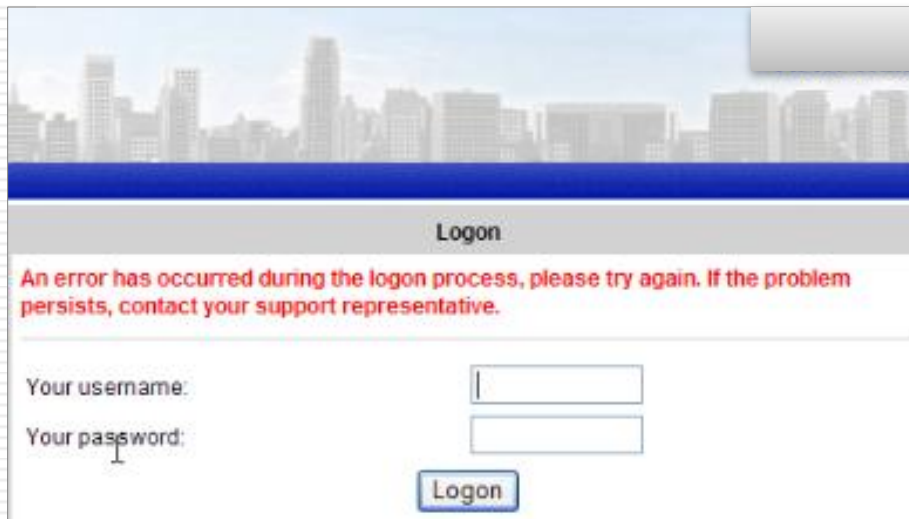
- **Botnet** is a collection of software agents (robots) that run autonomously and automatically.
- Execute malicious functions in a coordinated way
 - Send spam email
 - Collect identity information
 - Denial of service attacks
- Named after their malicious software, but there are multiple botnets using the same malicious software families operated by different criminal groups
- A botnet's originator (aka "bot herder" or "bot master") can control the group remotely

What is a botnet



Screen Injection by Zeus bot

Browser NOT infected by Zeus:



Logon

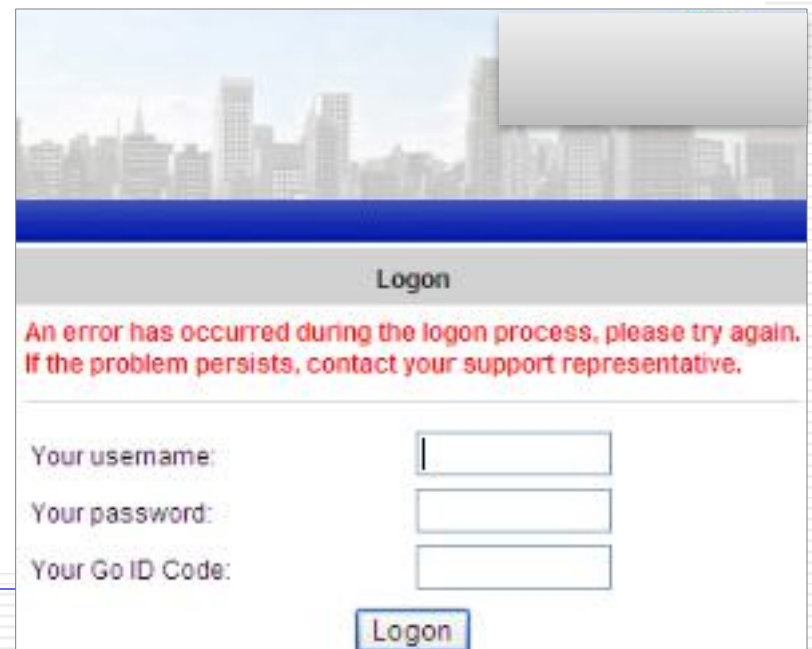
An error has occurred during the logon process, please try again. If the problem persists, contact your support representative.

Your username:

Your password:

Logon

Browser infected by Zeus:



Logon

An error has occurred during the logon process, please try again. If the problem persists, contact your support representative.

Your username:

Your password:

Your Go ID Code:

Logon

Zeus bot statistics

- 784 Zeus Botnets tracked by Zeus Tracker in 2009
- Estimate of 1.6M bots in Zeus botnets
- 1130 organisations targeted
- 960 financial organisations targeted (85%)
- Each of the top 5 US banks targeted by over 500 Zeus botnets

Application Security



The Buffer Overflow Problem

```
void foo(char *s) {  
    char buf[10];  
    strcpy(buf,s);  
    printf("buf is %s\n",s);  
}  
...  
foo("thisstringistolongforfoo");
```

Buffer Overflow Exploitation

- The general idea is to give programs (servers) very large strings that will overflow a buffer.
- For a server with sloppy code – it's easy to crash the server by overflowing a buffer.
- It's sometimes possible to actually make the server do whatever you want (instead of crashing).

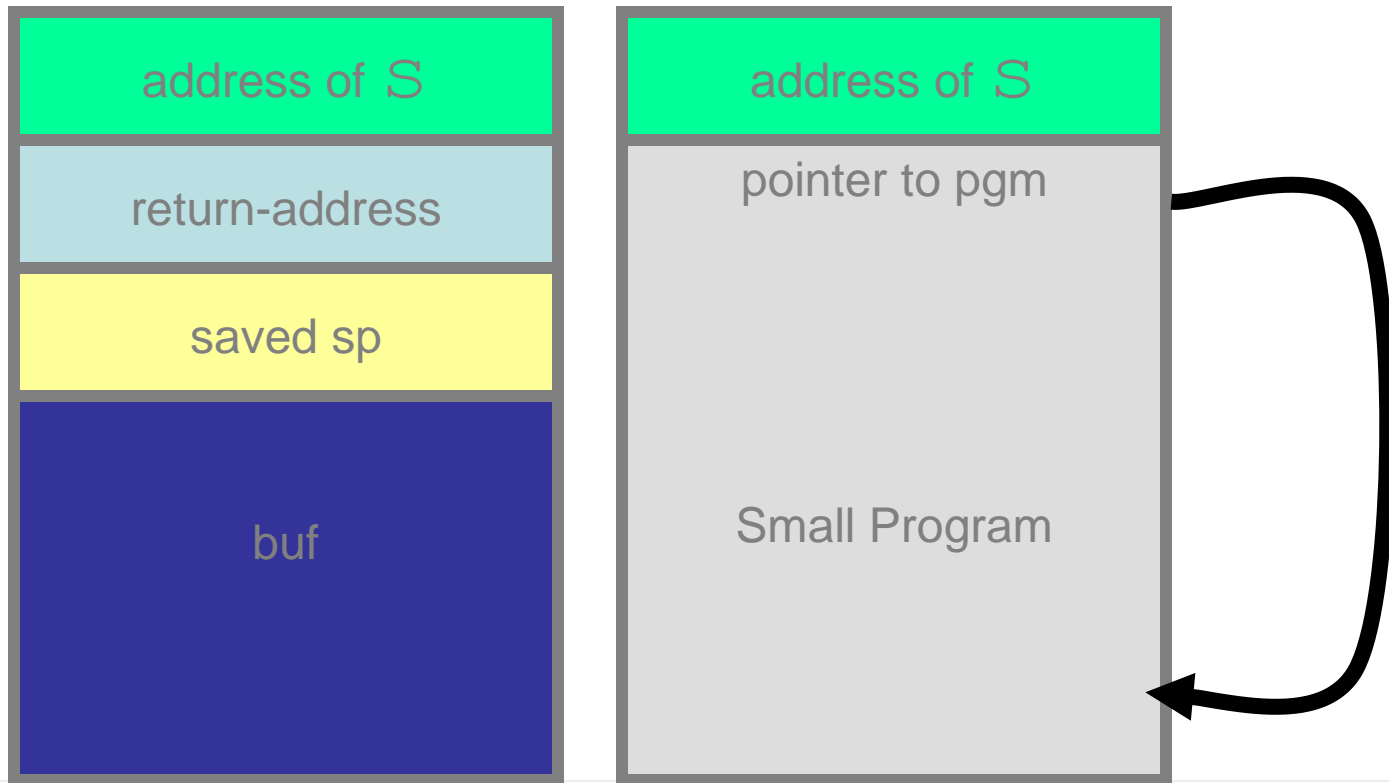
“Smashing the Stack”*

- The general idea is to overflow a buffer so that it overwrites the return address.
- When the function is done it will jump to whatever address is on the stack.
- We put some code in the buffer and set the return address to point to it!

*taken from the title of an article in Phrack 49-7

Before and After

```
void foo(char *s) {  
    char buf[100];  
    strcpy(buf, s);  
    ...  
}
```



Prevention of Buffer Overflow

- Use a programming language that provides control of string types and sizes
- Check during software design
- Test with fuzzing-up tools

*taken from the title of an article in Phrack 49-7

SQL Injection: What is SQL?

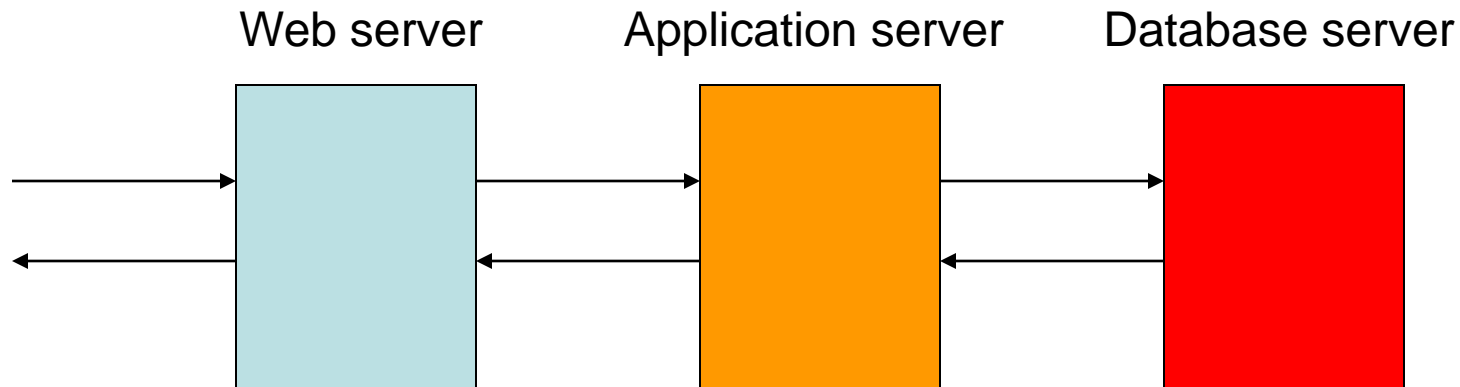
- Structured Query Language: interface to relational database systems.
- Allows for insert, update, delete, and retrieval of data in a database.
- ANSI, ISO Standard, used extensively in web applications.

- **Example:**

```
select ProductName from products where  
ProductID = 40;
```


How is it normally used in websites?

1. Take user input from a web form and pass it to a server-side script via HTTP methods such as POST or GET.
2. Process request, open connection to database.
3. Query database and retrieve results.
4. Send processed results back to user.



PHP example

```
$name = $_HTTP_POST_VARS["name"];  
$query = "select * from restaurants where  
    name = '". $name ."'";  
$result = mysql_query($query);
```

What is SQL Injection?

- The ability to inject SQL commands into the database engine through existing application.

- For example, if user input is **“23 or 1 = 1”**

```
select ProductName from products where  
ProductID = 23 or 1 = 1
```

- All product names will be returned. Data leak.

What is SQL Injection?

- Flaw in **web application** not in database or web server.
- No matter how patched your system is, no matter how many ports you close, an attacker can get complete ownership of your database.
- NMap or Nessus will not help you against sloppy code.
- In essence client supplied data without validation.

What can SQL Injection do?

- **Delete:**

```
Select productinfo from table where  
productname = 'whatever'; DROP TABLE  
productinfo; -- '
```

- **Bypass Authentication**

- Select * from users where username='user
' and password='passwd ';
- select * from users where
username='admin'--' and
password='whocares' ;

Possibilities are endless

- Some examples:
 - Brute forcing passwords using attacked server to do the processing.
 - Interact with OS, reading and writing files.
 - Gather IP information through reverse lookup.
 - Start FTP service on attacked server.
 - Retrieve VNC passwords from registry.
 - File uploading.

Prevention of SQL Injection

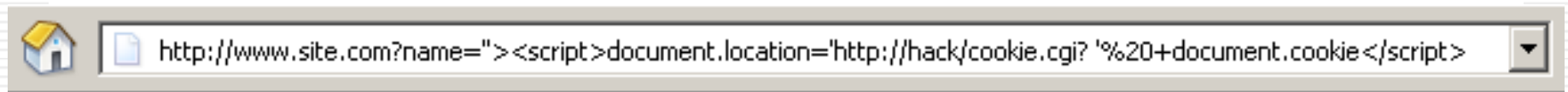
- **Check and filter user input.**
 - Length limit on input (most attacks depend on long query strings).
 - Different types of inputs have a specific language and syntax associated with them, i.e. name, email, etc
 - Do not allow suspicious keywords (DROP, INSERT, SELECT, SHUTDOWN) as name for example.
 - Try to bind variables to specific types.

Cross-Site Scripting (XSS) Attacks

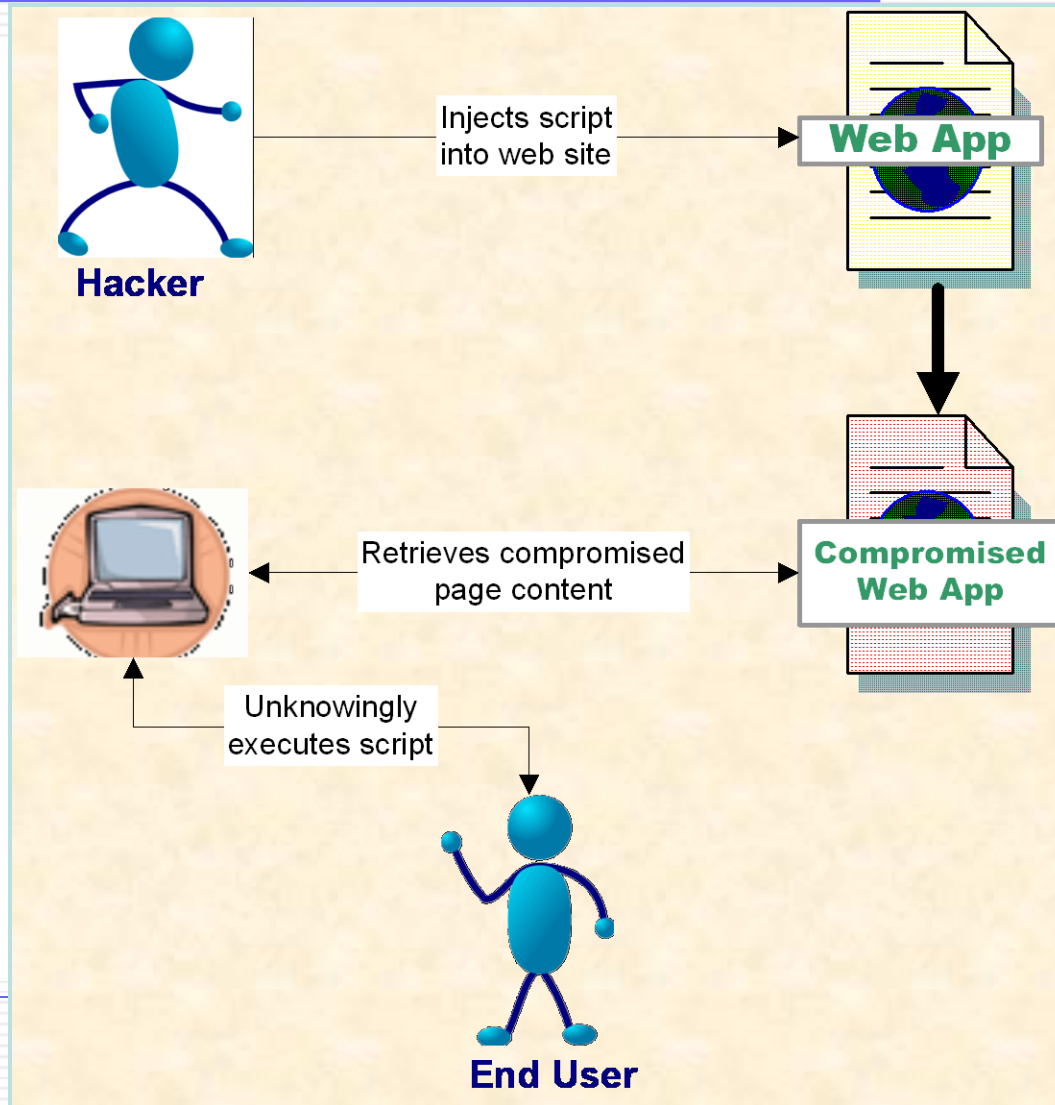
- Malicious code can secretly gather sensitive data from user while using authentic website (login, password, cookie)

Cross-Site Scripting (XSS) Attacks

- Modified URL
 - URL parameters are modified on the URL to contain script code
 - Input is not validated and displayed as entered on the resulting dynamic webpage



Cross-Site Scripting (XSS) Attacks



XSS: Script Injection Demo

Forum

Folders

Empowerment Systems Forum

Subject	Posted By	Time & Date
<<	nasty user	3:09:21 PM 3/30/2006
... & availability	David G...	4:34:39 PM 4/21/2005
... ..	David G...	8:02:49 AM 4/18/2005
... ..	David G...	10:05:44 AM 1/27/2005
...	10:54:45 PM 1/20/2005
...	10:51:44 PM 1/20/2005

Use following form to post to current forum:

Name:

E-Mail:

Subject:

Message:

```
><script>alert('you have an XSS  
vulnerability')</script><
```

Preventing SQL injection and XSS

- **SCRUB Error handling**
 - Error messages divulge information that can be used by hacker...
- **VALIDATE** all user entered parameters
 - **CHECK** data types and lengths
 - **DISALLOW** unwanted data (e.g. HTML tags, JavaScript)
 - **ESCAPE** questionable characters (ticks, --, semi-colon, brackets, etc.)

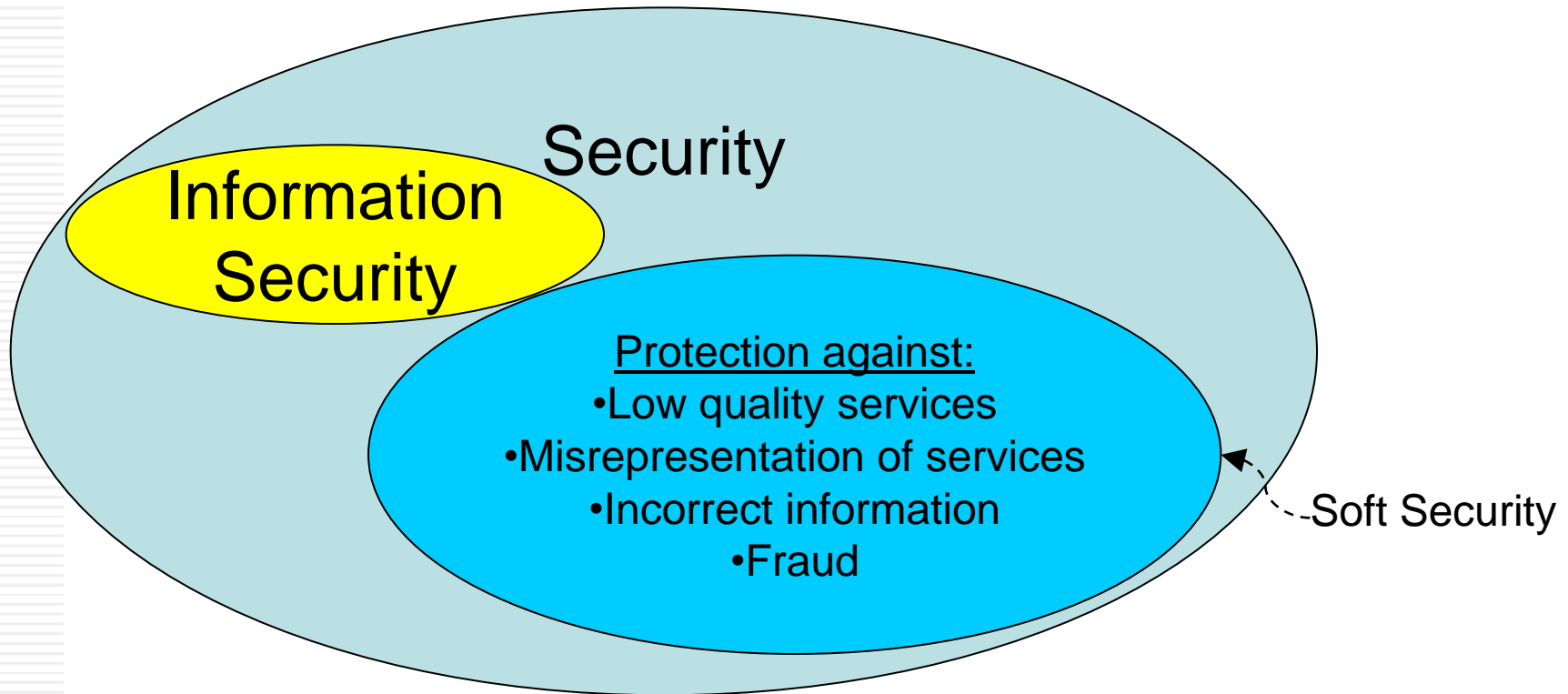
Trust Management and Soft Security



What is Security?

- General definition of security:
 - *Protection from danger*
 - Oxford English Online Dictionary: <http://dictionary.oed.com/>
- Traditional definition of information security:
 - *Preservation of confidentiality, integrity & availability of information*
 - ISO/IEC 27001:2005 Specification for an Information Security Management System
 - Assumes that the owner of information resources
 - defines a security policy (explicitly or implicitly)
 - implements measures to preserves CIA properties

Gap analysis of security and information security



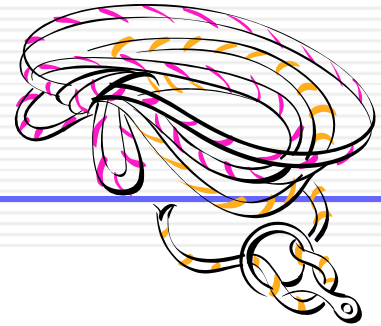
Soft Security

- Impossible to define security policies for open communities
- Common ethical norms instead of security policy
 - Can be partly formal and partly dynamic/collaborative
- Definition:
 - ***Adherence to common (ethical) norms***
- Stimulates the quality of communities in terms of ethical behaviour and integrity of its members
- Enforced by collaborative mechanisms such as trust and reputation systems

Two definitions of trust

- Evaluation trust
 - The **subjective probability** by which an individual, *A*, expects that another individual, *B*, performs a given action on which its welfare depends. (Gambetta 1988)
- Decision trust
 - The **willingness to depend** on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible. (McKnight & Chervany 1996)

Would you trust this rope?



For what?

To climb down from the 3rd floor window of a house

The rope looks very old



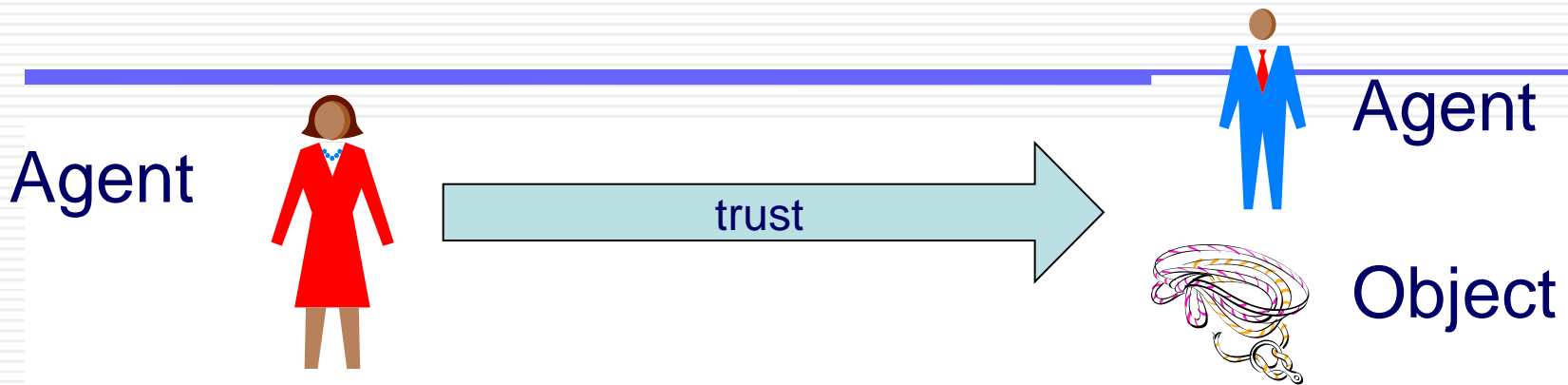
Fire drill:

No!

Real fire:

Yes!

Trust is a relationship



- Trusting party

- Also called

- “relying party”
- “trustor”

- Is in a situation of

- Dependence

- Trusted party

- Also called

- “trustee”

- Is in a situation of

- Power
- Expectation to deliver

Two sides of trust management

Trusting party

Wants to **assess** and make **decisions** w.r.t. the dependability of the trusted party for a given transaction and context



Trusted party

Wants to **represent** and put in a **positive light** own competence, honesty, reliability and quality of service.



Reputation and trust

REPUTATION

- Public info
- Common opinion
- Not necessarily objective

TRUST

- Both private and public info
- Private info carries more weight
- Subjective

- *“I trust you because of your good reputation”*
- *“I trust you despite your bad reputation”*

Extrinsic and intrinsic trust

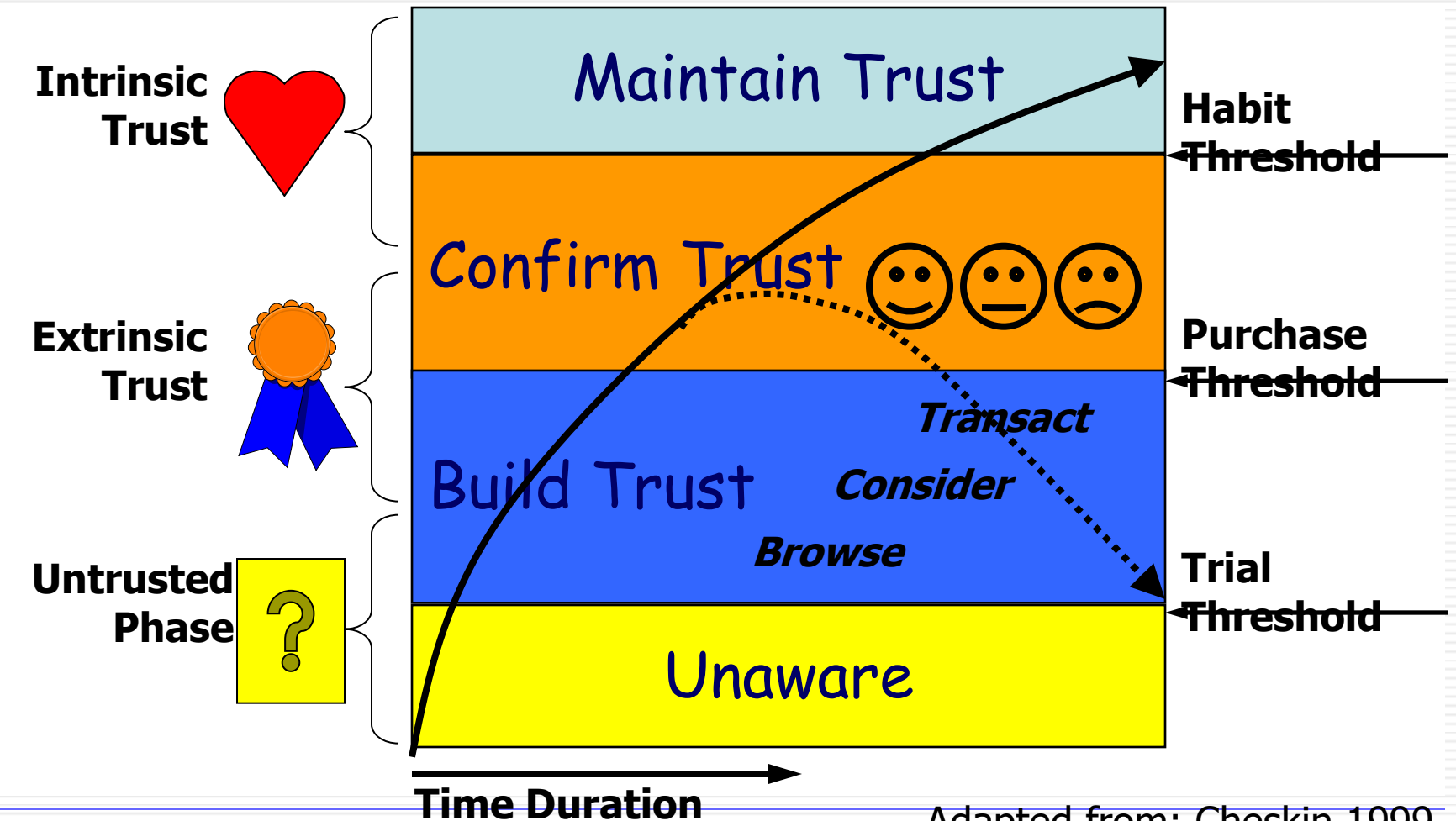
Extrinsic Factors

- Cognitive
- Observed
- Recommendation
- Reputation
- External evidence
- Easy to manufacture

Intrinsic Factors

- Affective
- Experienced
- Intimate relationship
- Internalised pattern
- Take time to build
- Override extrinsic

A model for e-commerce trust



We trust what we depend on

Trust in people
& organisations

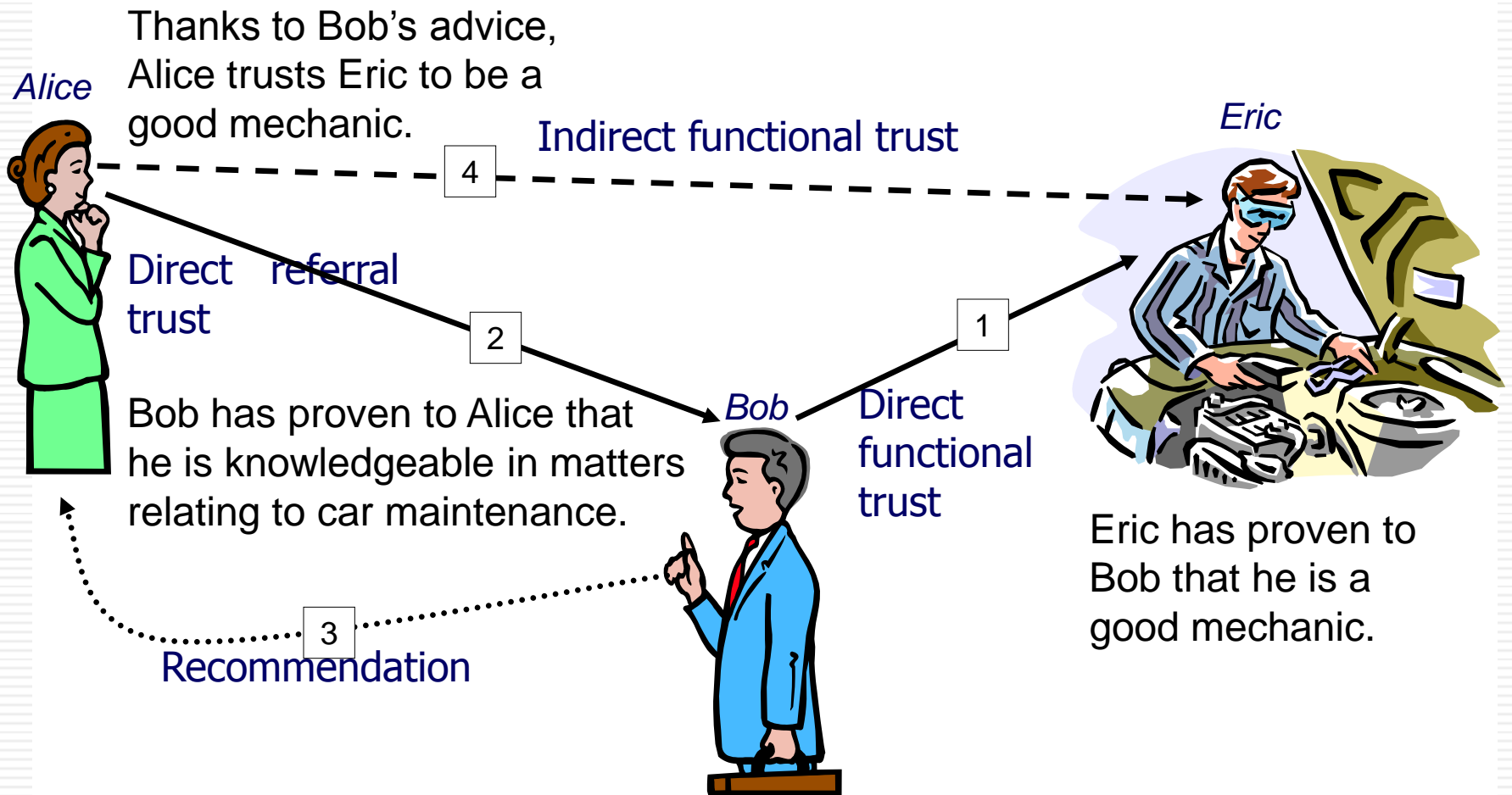
Trust in legal,
social and market
institutions

Trust in ICT

Formal aspects of trust

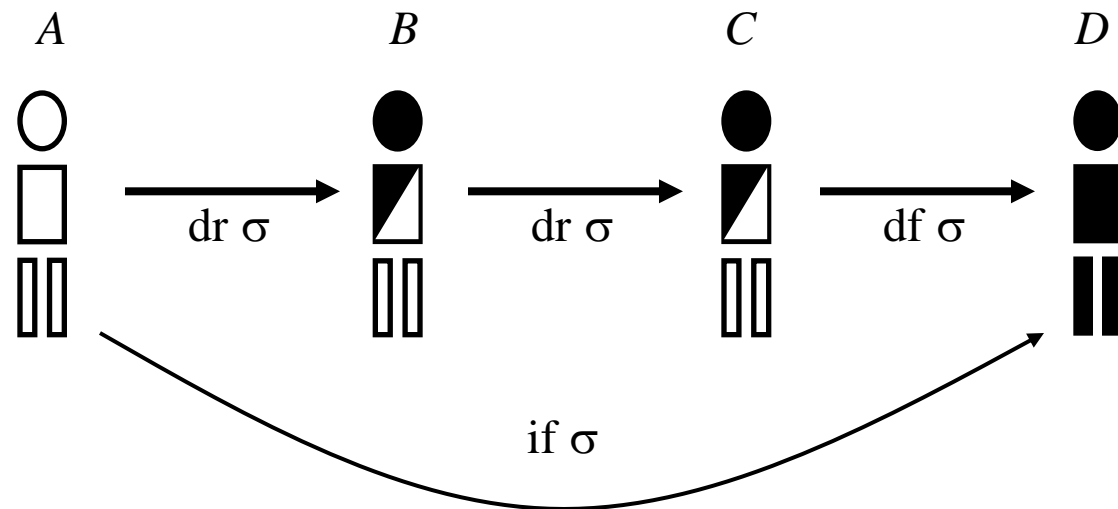
- **Trust scope**
 - Function that the relying party depends on and trusts
- **Functional trust:**
 - The trusted party performs the function
- **Referral trust:**
 - The trusted party recommends a party (who recommends a party) that can perform the function
- **Direct trust:**
 - From direct experience
- **Indirect trust:**
 - From recommendations

Computational Trust

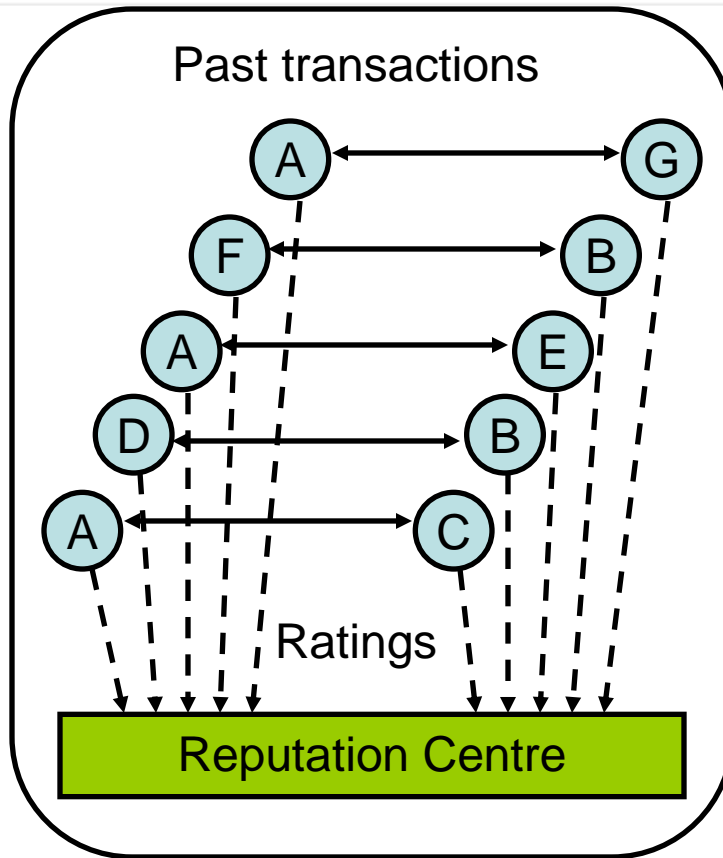


Valid transitive trust chains

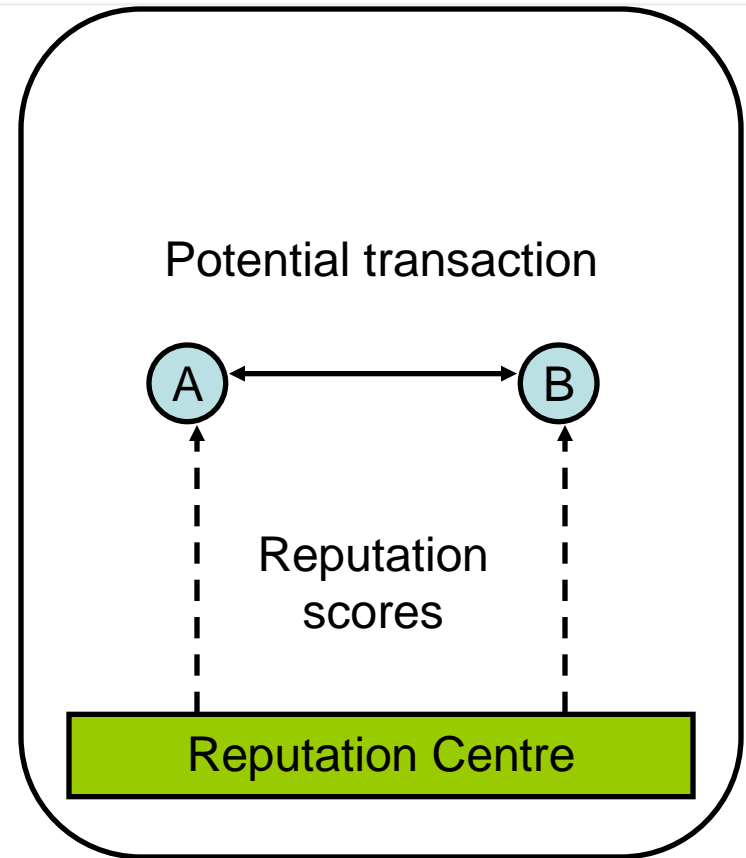
- Every leg in the chain contains the same trust scope $[\sigma]$. (It doesn't make any sense otherwise!)
- The last trust link is **direct functional** trust $[df\sigma]$.
- All other trust links are **direct referral** trust $[dr\sigma]$.



Centralised reputation system

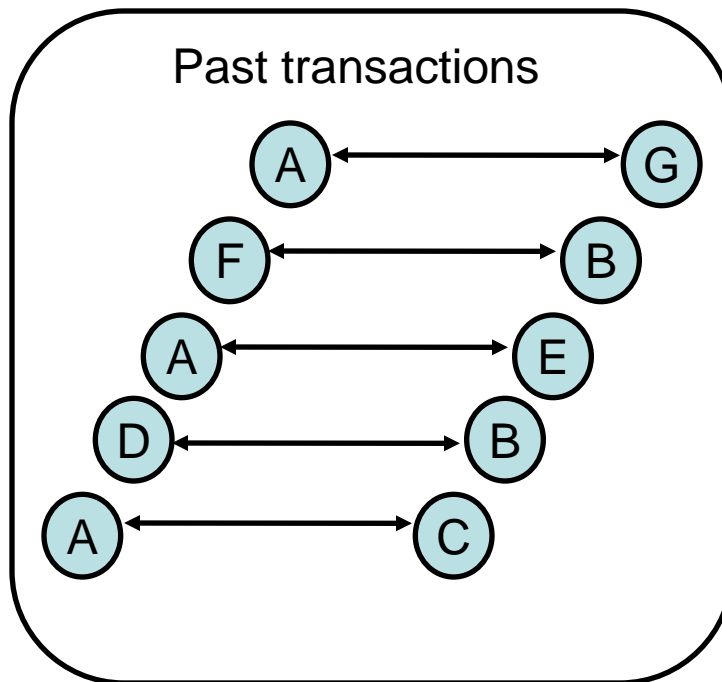


a) Past

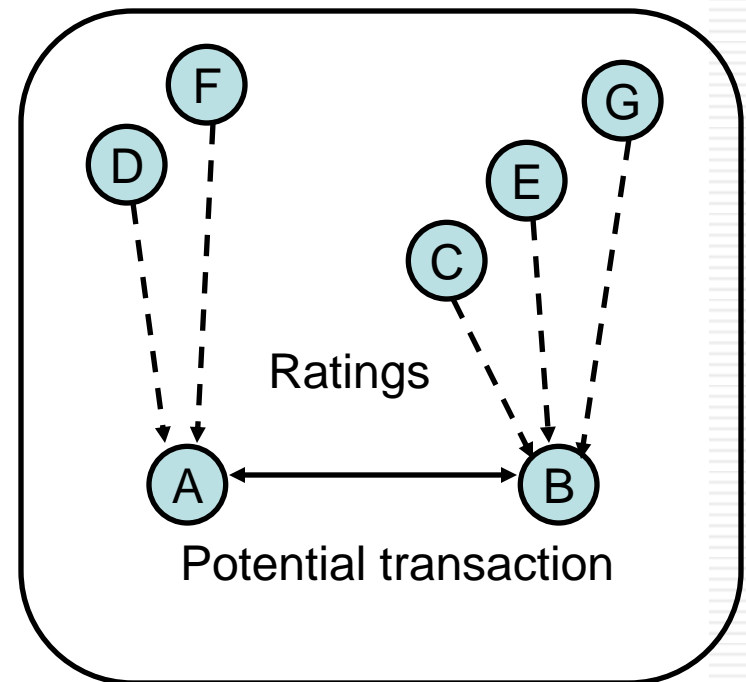


b) Present

Distributed reputation system



a) Past



b) Present

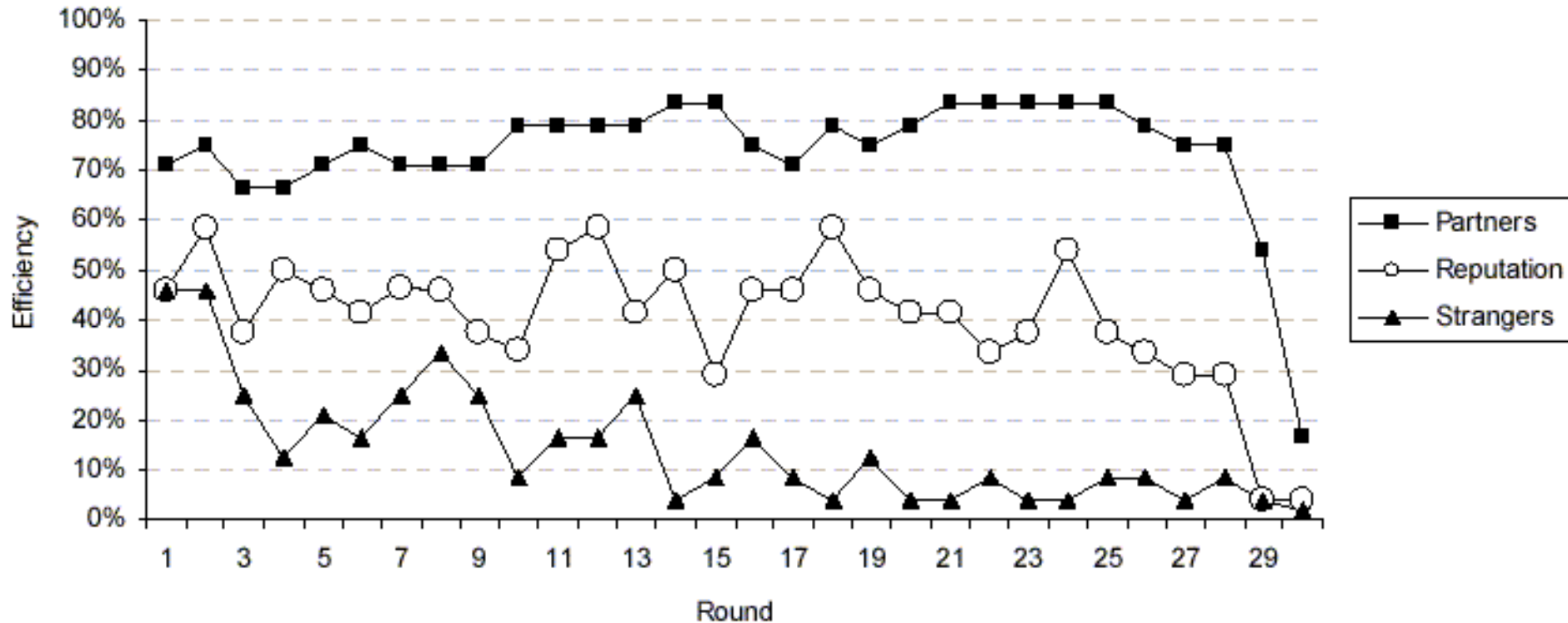
Trust/Reputation System Categories

	Private Scorers	Public Scores
Transitivity	Trust systems, e.g. Rummbles.com	Public trust systems, e.g. PageRank
No transitivity	Private reputation systems, e.g. customer feedback analysis	Reputation systems, e.g. eBay.com

Applications of trust and reputation systems

- e-Auctions
- Social websites
- Online markets: B2C, B2B, C2C
- P2P networks
- Software agent communities
- Contract negotiations
- Web service search and selection
- Spam filtering

Market Efficiency Experiment



Source: Bolton, Katok, Ockenfels, 2002

Google's PageRank

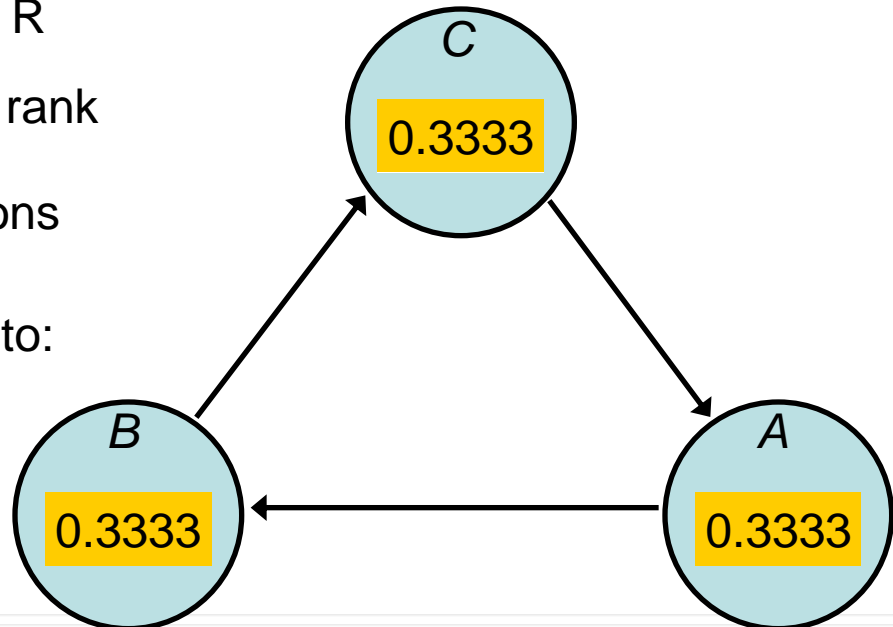
- Purpose to provide quality search results
- Based on:
 - Number of incoming links, weighted by the
 - PageRank of the sites behind incoming links
- Hyperlinks interpreted as positive ratings.
- No negative ratings.
- Random surfer model.
- PageRank is a reputation system

PageRank visualisation

- $R(A) = (1-d)/N(\text{Web}) + d \cdot \sum R(\text{prev}(A))/N(\text{next}(\text{prev}(A)))$
- Damping factor $d \approx 0.85$
- $\sum R(A) \approx 1$, i.e. $R(A)$ is the probability of the random surfer
- $\text{PageRank}(A) = I + \log_{\approx 10} R(A)$, where $I \approx 11$

Example
with $N(\text{Web})=3$

Initial rank R
+ imported rank
and iterations
Converges to:



Link spam and “nofollow”

- Survival of e-commerce sites depends on rank
- Attempts to increase rank with link spam
 - consists of putting URLs to own Web site in wikis (publicly editable Web sites) and in postings to public discussion groups
- The “nofollow” tag, introduced in 2005, instructs Web crawlers not to follow a link

```
<a href=http://spam_site.com  
rel="nofollow">Link</a>
```

- Wikis and discussion groups now add “nofollow” to all URLs, thereby eliminating the link spam problem

SERP Rank

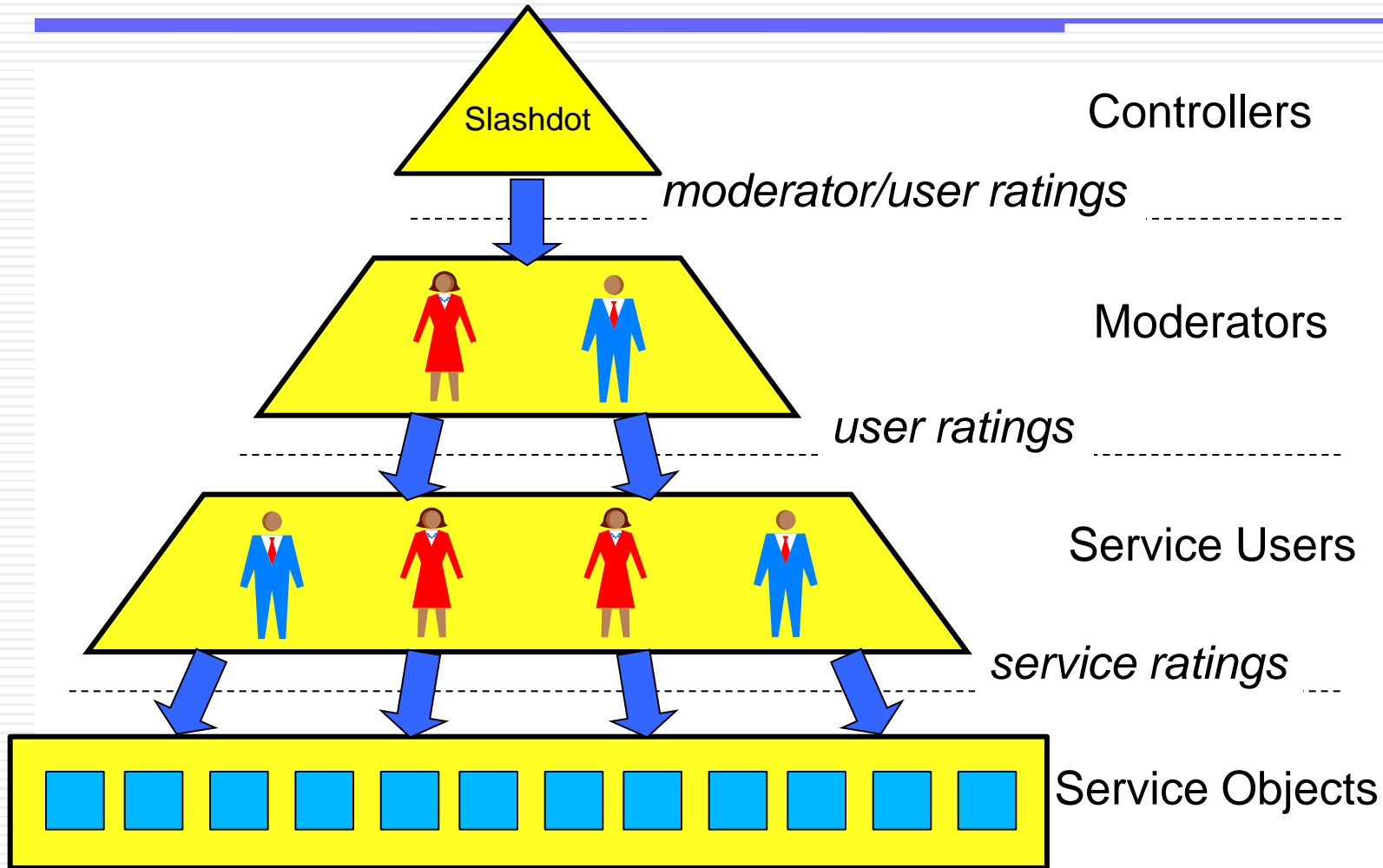
- SERP: Search Engine Results Page
- SERP Rank: Position of page reference on SERP
- \neq PageRank
- SERP Rank is a function of PageRank + constantly tuned factors:
 - Keyword position and frequency
 - Linking to good neighbourhoods
 - Freshness
 - etc.

Slashdot “News for nerds” message board and the Slashdot Reputation System

- Article postings, at Slasdot’s discretion
- Comments to articles posted by members
- Comment moderation by members
 - Positive: insightful, interesting, informative funny, underrated
 - Negative: offtopic, flamebait, troll, redundant, overrated
 - Comment score $\approx \Sigma \text{positive(Karma)} - \Sigma \text{negative(Karma)}$,
 - Moderation by members with high Karma carries more weight
- Comment viewing filtered by score
- Member Karma
 - Terrible, bad, neutral, positive, good, excellent
 - Based on moderation of comments.
- Metamoderation, to combat unfair moderation
 - Rate the moderations: fair, unfair, neutral
 - Affects Karma of member who gave the moderation
- Arbitrary moderation by Slashdot staff
- Purpose: Directing massive collaborative moderation effort

Hierarchic reputation architecture

Slashdot model



End of Lecture
