INF3510 Information Security
University of Oslo
Spring 2011

Lecture 13

Application & Operations Security

Audun Jøsang

Outline

- Application Security
  - **Malicous Software**
    - various malicious programs
    - distributed denial of service attacks
  - **Attacks on applications**
    - Buffer overflos
    - SQL Injection
    - Cross-Site Scripting

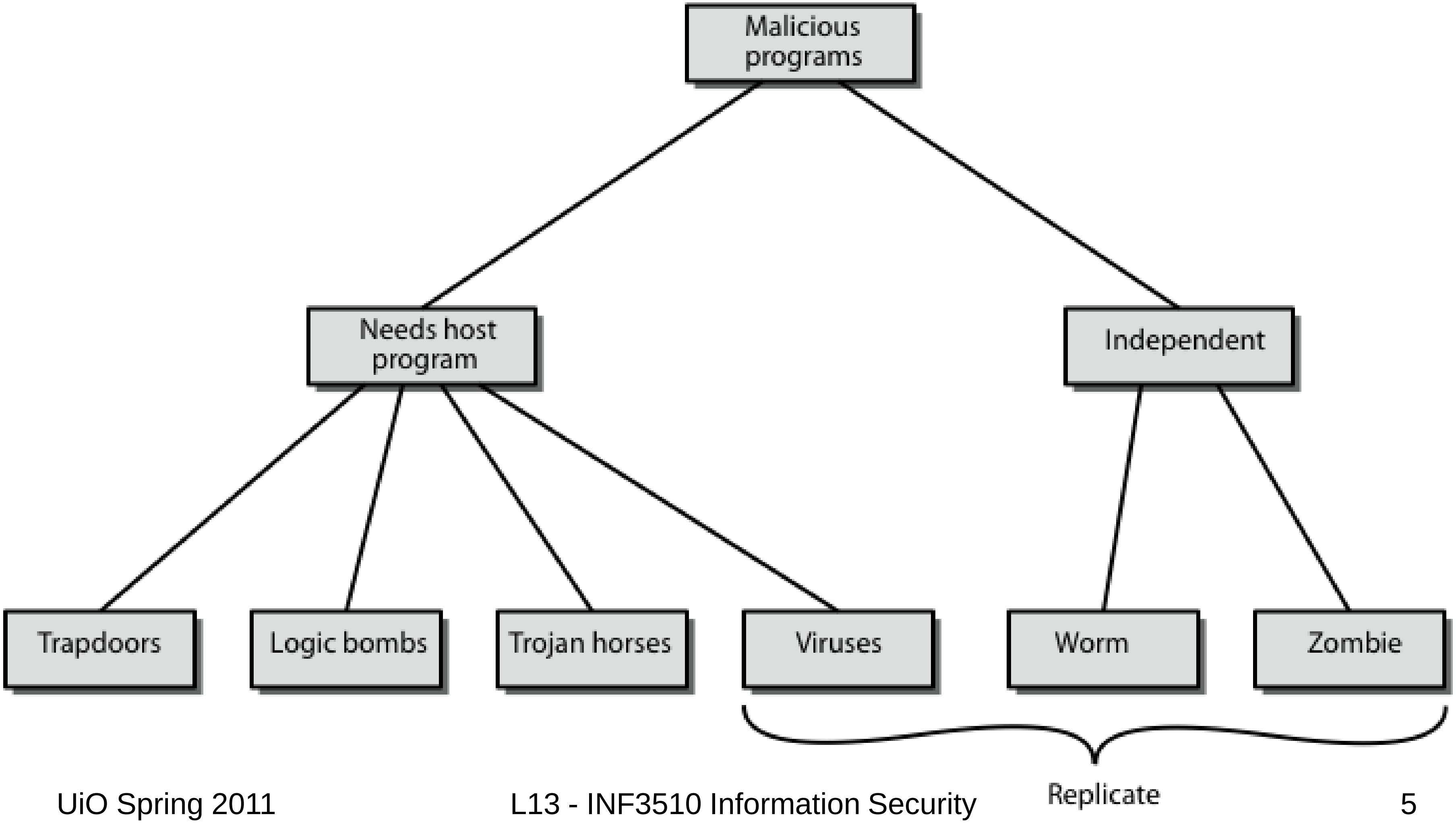
- Operations Security

Application Security

Malware: Malicious Content

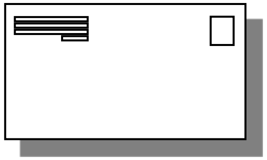
- Many different forms
- Many different effects
- Difficult to know when infected
- More advanced forms emerge
- A growing concern

Malicious Software

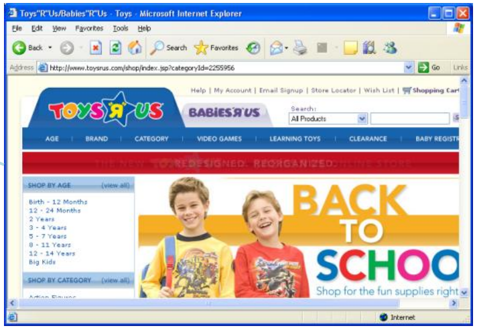


How do computers get infected ?

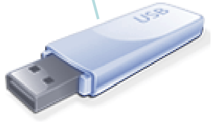
Direct attacks from the network, as worms or exploitation of application vulnerabilities such as SQL injection or buffer overflows



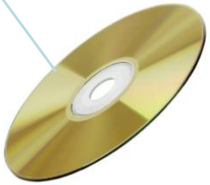
Executing an attachment


Toys"R"Us/Babies"R"Us - Toys - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.toysrus.com/shop/index.jsp?categoryId=2255956 Go Links
Help | My Account | Email Signup | Store Locator | Wish List | Shopping Cart
TOYS "R" US
BABIES "R" US
Search: All Products
AGE | BRAND | CATEGORY | VIDEO GAMES | LEARNING TOYS | CLEARANCE | BABY REGISTR
REDESIGNED. REORGANIZED.
SHOP BY AGE (view all)
Birth - 12 Months
12 - 24 Months
2 Years
3 - 4 Years
5 - 7 Years
8 - 11 Years
12 - 14 Years
Big Kids
SHOP BY CATEGORY (view all)
BACK TO SCHOO
Shop for the fun supplies right
Internet

Accessing a malicious or infected website or starting application from a website



Plugging in external devices



Installing infected software

Backdoor or Trapdoor

- secret entry point into a program
- allows those who know access bypassing usual security procedures
- have been commonly used by developers for testing
- 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
- causes damage when triggered
  - modify/delete files/disks, halt machine, etc

Trojan Horse

- program with hidden side-effects
- program is usually superficially attractive
  - eg game, s/w upgrade etc
- performs additional tasks when executed
  - allows attacker to indirectly gain access they do not have directly
- often used to propagate a virus/worm or to 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 programs 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
- Virus defenses:
  - Block initial infection (difficult)
  - Block further propagation (with access controls)
  - Detect and remove after infection
  - Re-install OS + programs + data

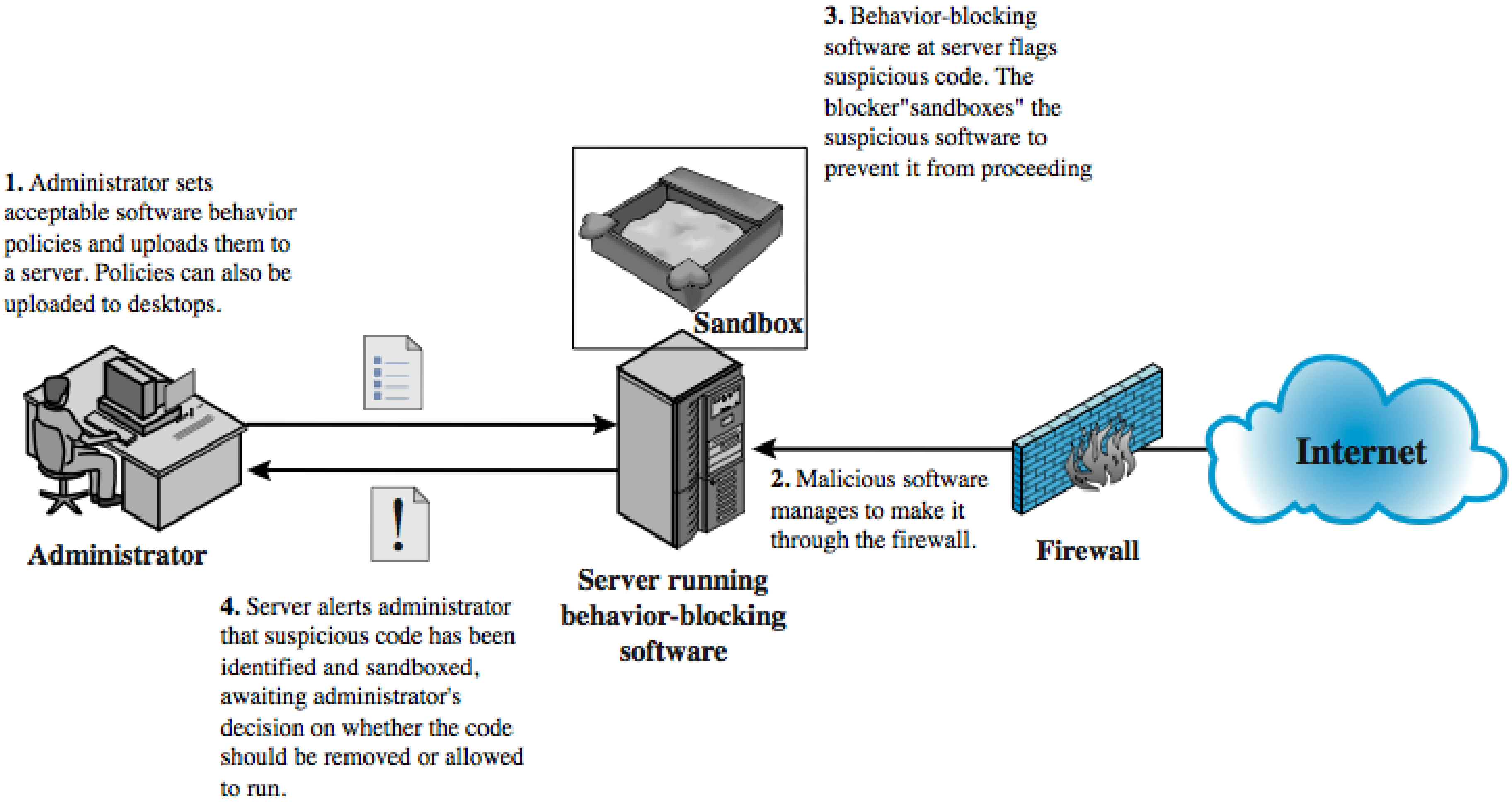
Some virus types

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

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



1. Administrator sets acceptable software behavior policies and uploads them to a server. Policies can also be uploaded to desktops.

2. Malicious software manages to make it through the firewall.

3. Behavior-blocking software at server flags suspicious code. The blocker"sandboxes" the suspicious software to prevent it from proceeding

4. Server alerts administrator that suspicious code has been identified and sandboxed, awaiting administrator's decision on whether the code should be removed or allowed to run.

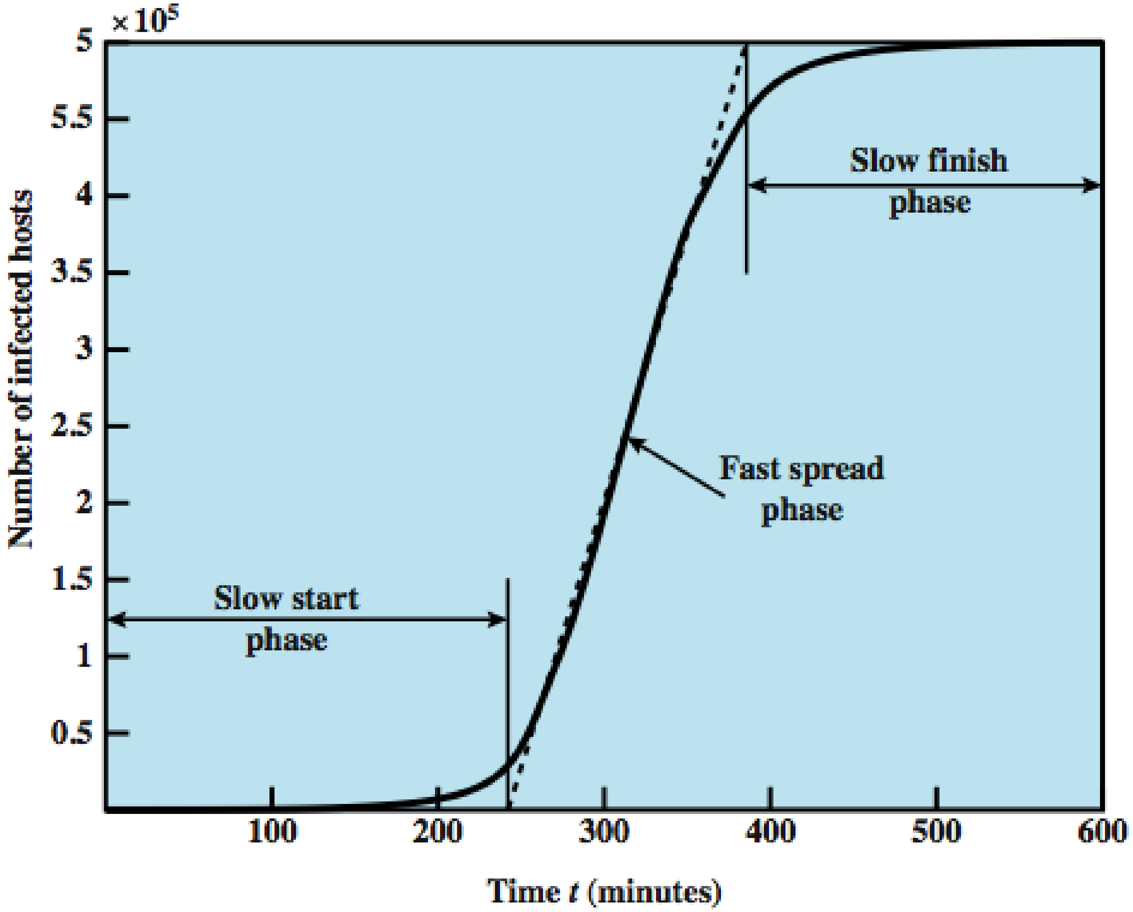
Worms

- 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

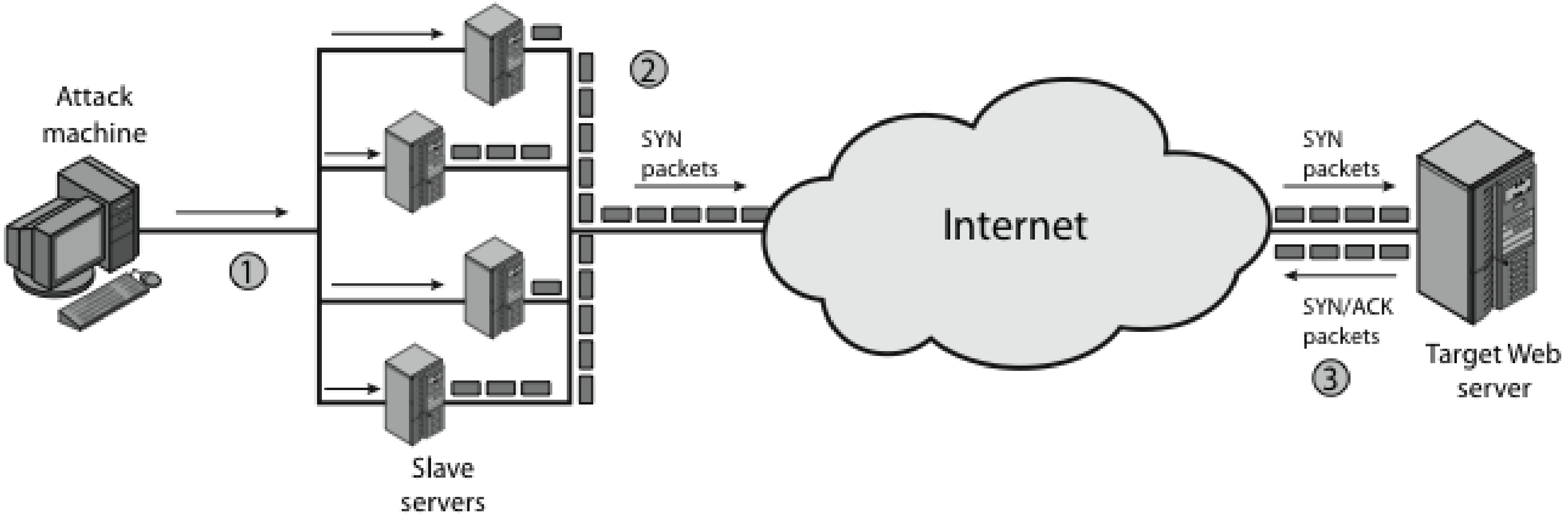
Worm Countermeasures

- 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

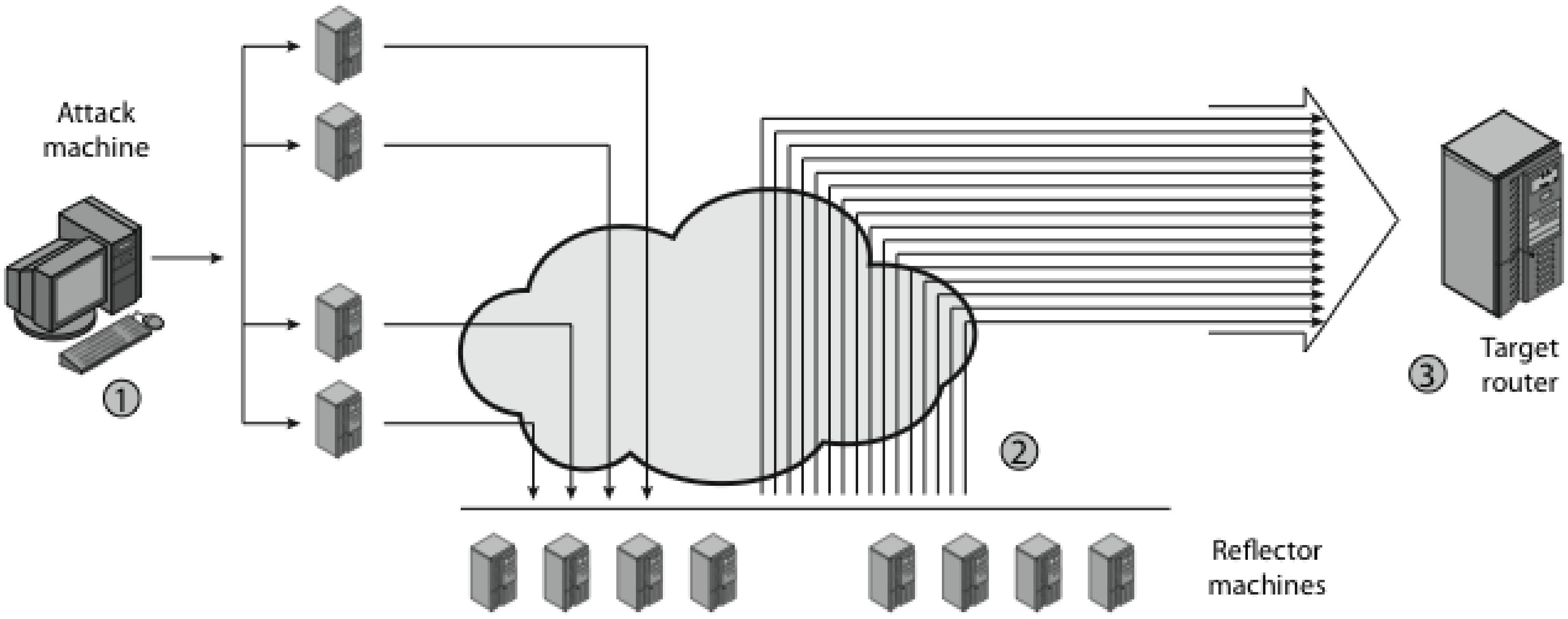
DDoS
Distributed Denial of Service Attacks

- 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

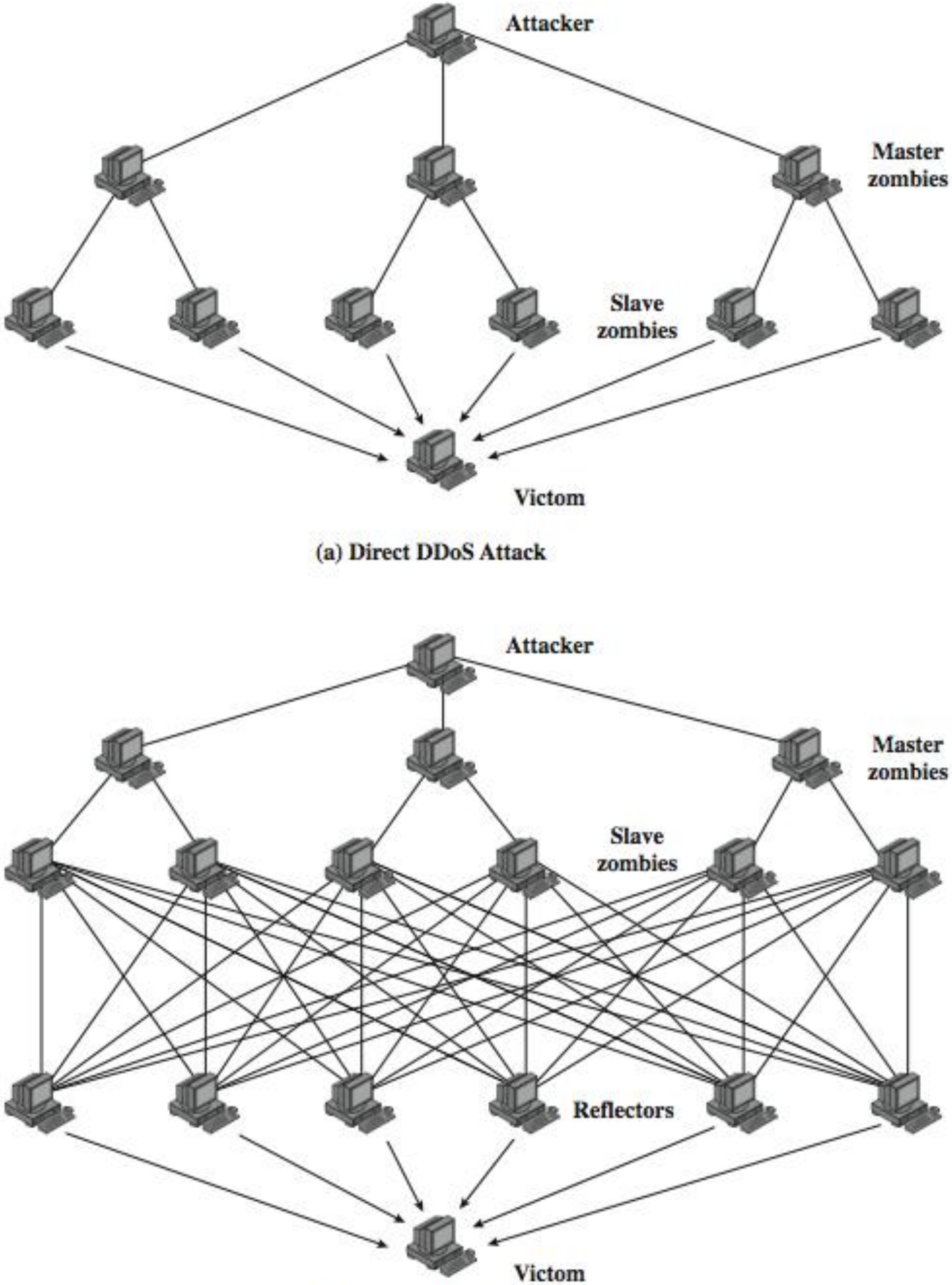


(a) Distributed SYN flood attack



(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

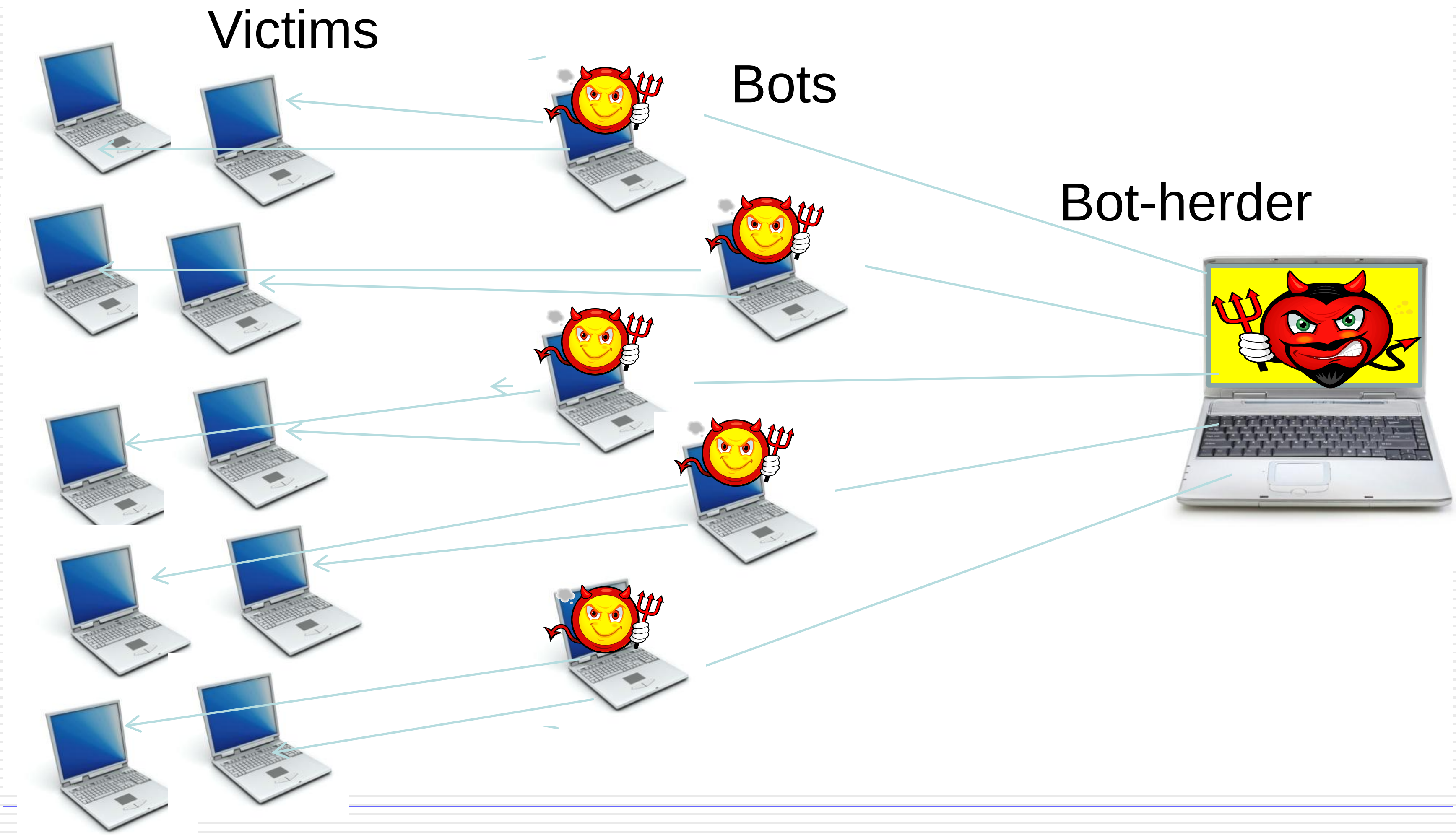
DDoS Countermeasures

- 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

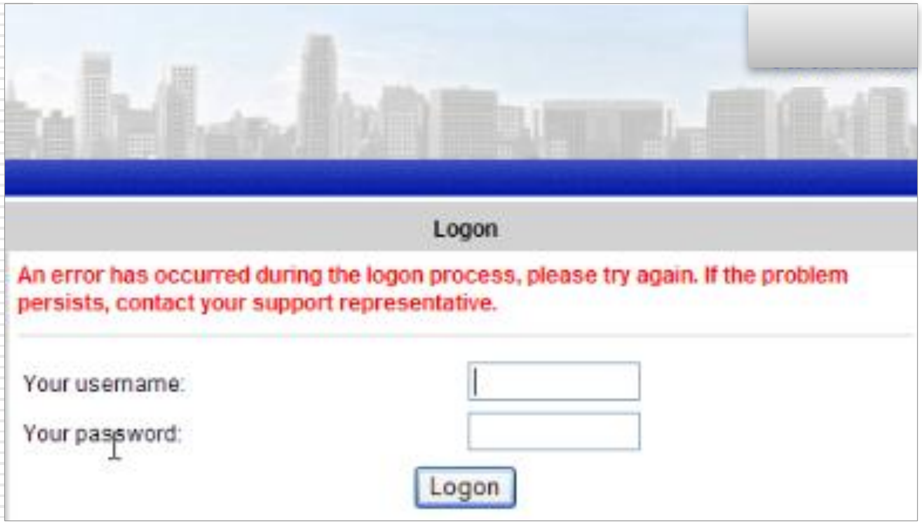
- **Botnet** is a collection of software agents (robots) that run autonomously and automatically.
- Execute malicious functions in a coordinated way
  - Send spam email
  - Collect identity information
  - Denial of service attacks
- A botnet is named after the malicious software, but there can be multiple botnets using the same malicious software, but 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:

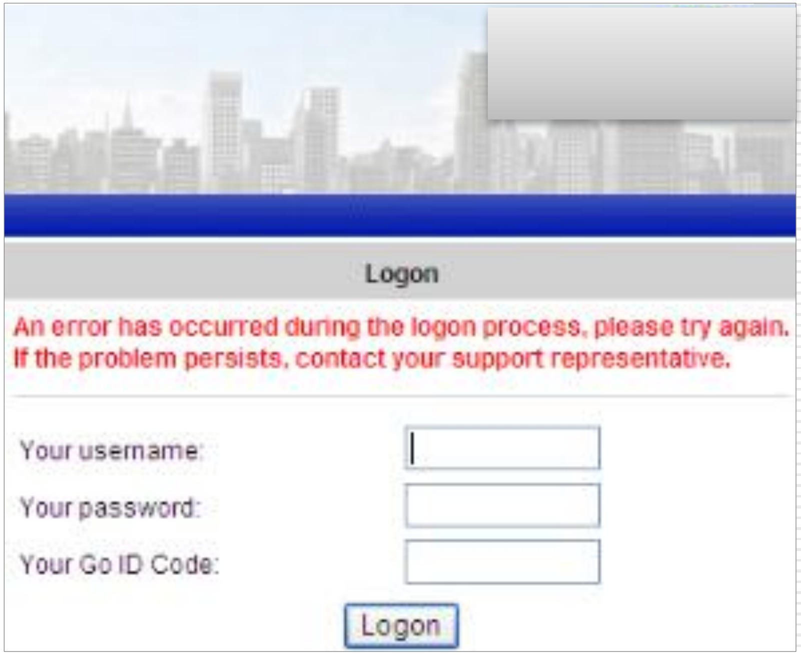


An error has occurred during the logon process, please try again. If the problem persists, contact your support representative.

Your username:

Your password:

Browser infected by Zeus:



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:

- Zeus is used to execute MitB (man-in-the-browser) attacks
- Asks for Go Id Code (OTP) which will be sent to attacker

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
- Norwegian banks attacked in February 2011

The Buffer Overflow Problem

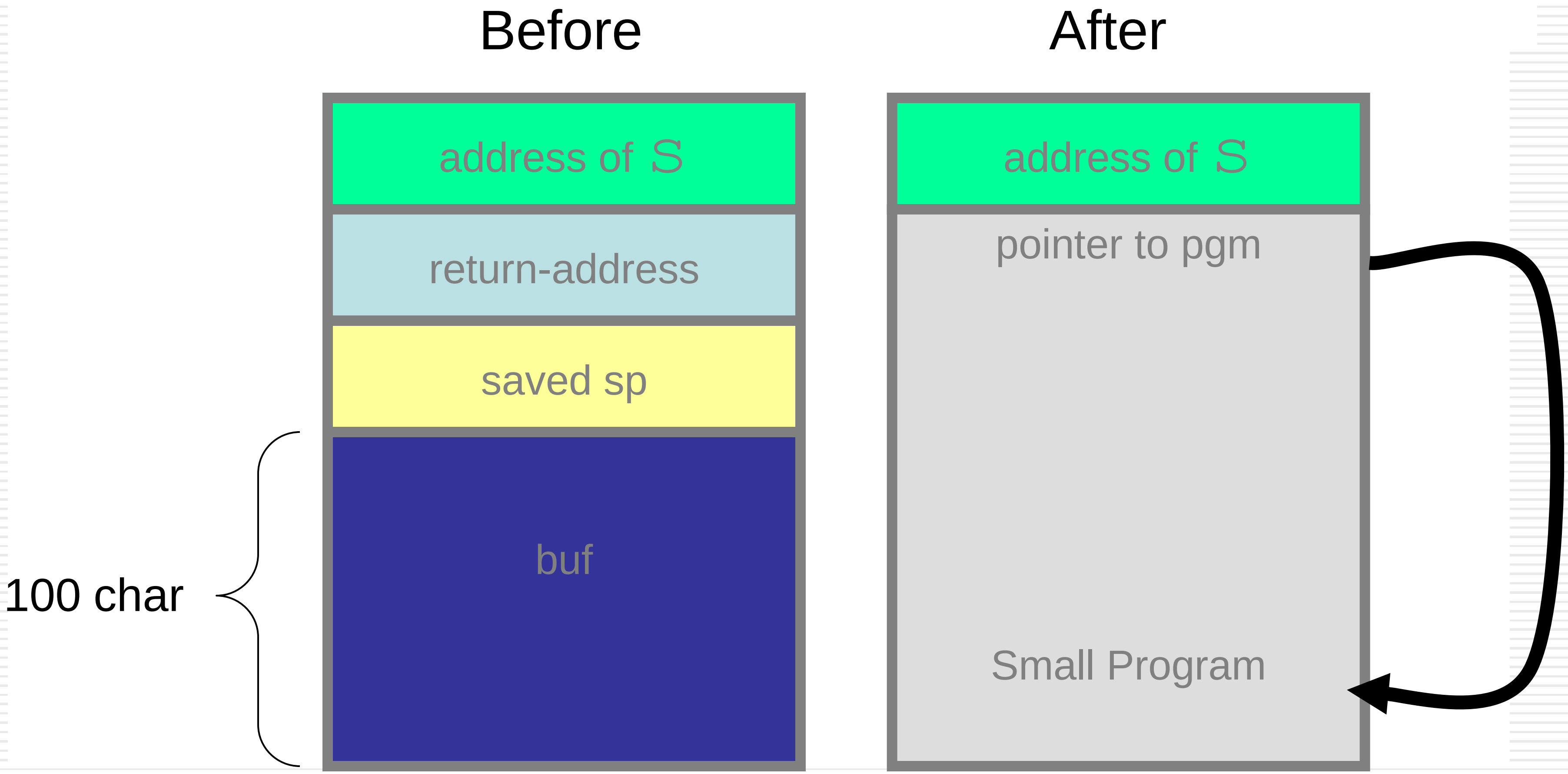
```
void foo(char *s) {
   char buf[10];
   strcpy(buf,s);
   printf(“buf is %s\n”,s);
}
…
foo(“thisstringistolongforfoo”);
```

Buffer Overflow Exploitation

- The attack is to give programs (servers) very large strings that will overflow a buffer.
- It’s easy to crash a server with sloppy code by overflowing a buffer.
- Attacker’s goal is to inject instructions into the buffer and make the server execute those instructions (instead of crashing).
- The overflow data in buffer overwrites return address on the program stack so that it points to the instructions written to the same stack

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

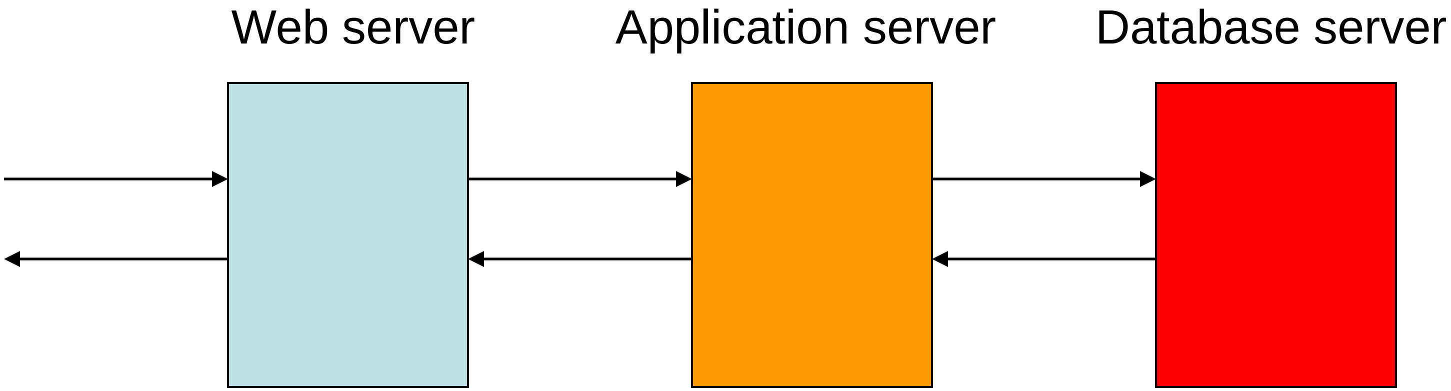
SQL Injection: What is SQL?

- 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.



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.

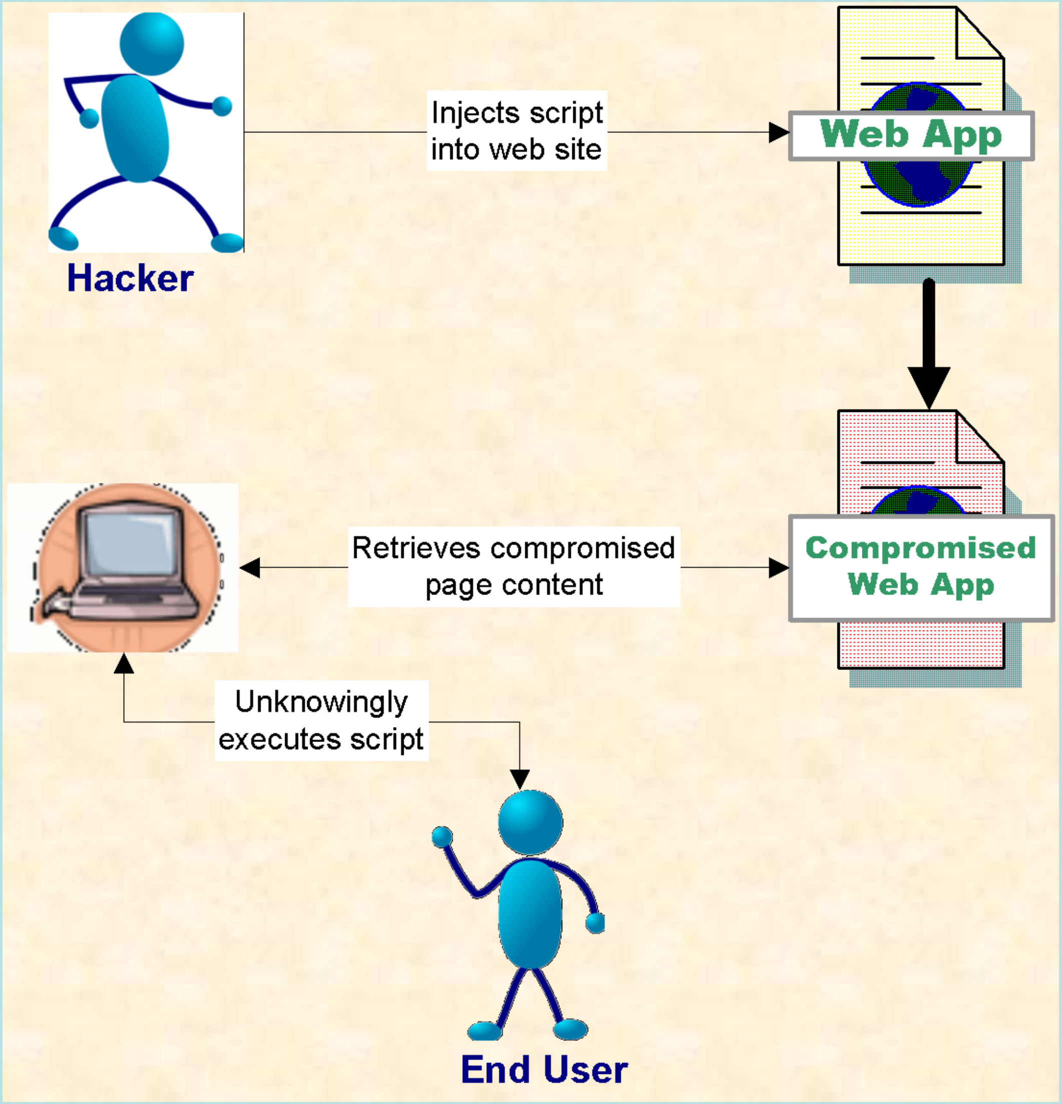
SQL injection possibilities are endless

- 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.

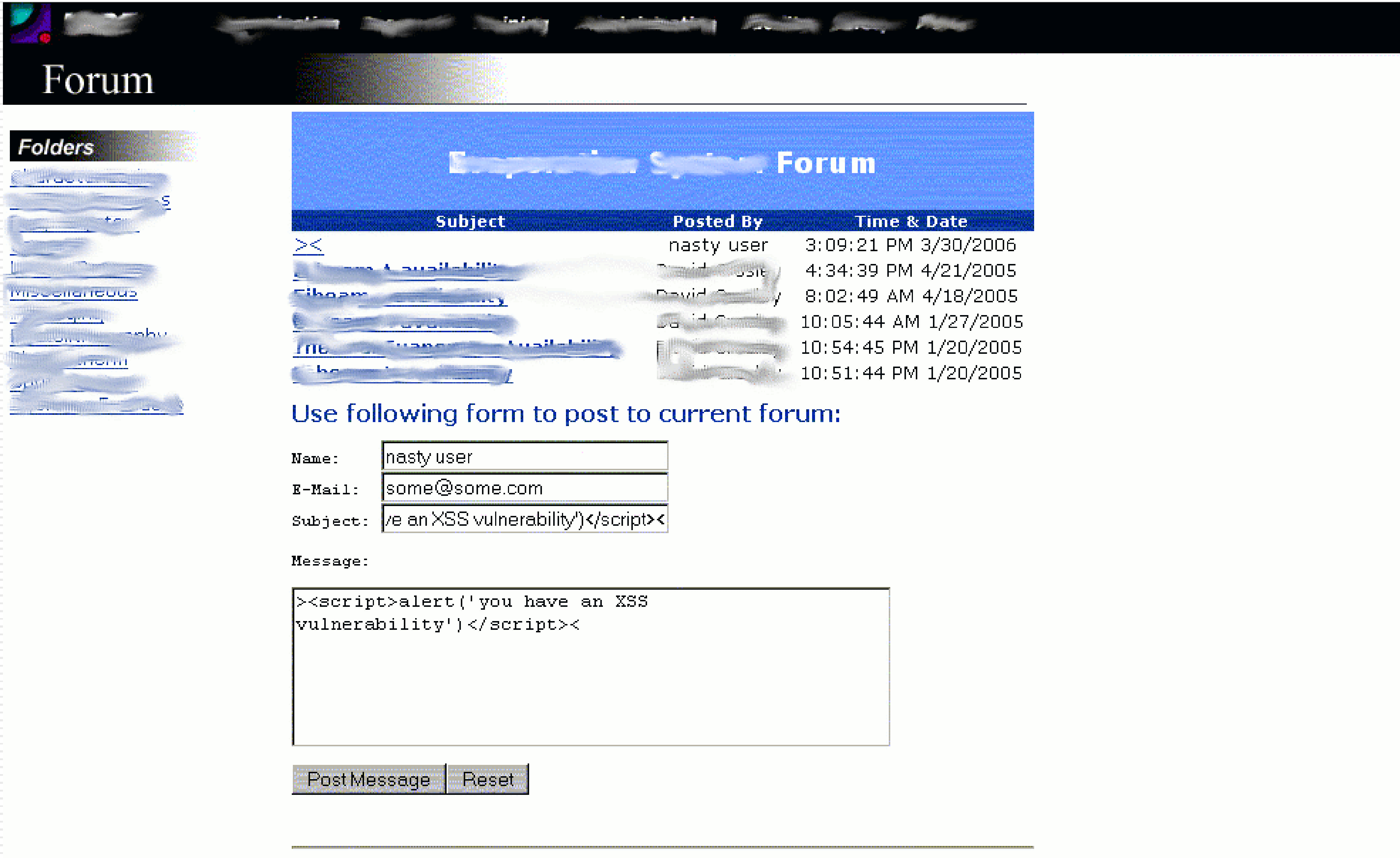
Prevention of SQL Injection

- **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


Hacker
Injects script into web site
Web App
Compromised Web App
Retrieves compromised page content
Unknowingly executes script
End User

XSS: Script Injection Demo



[redacted]

Forum

Folders

[redacted]

Miscellaneous

[redacted]

[redacted] [redacted] Forum

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

Use following form to post to current forum:

Name: nasty user

E-Mail: some@some.com

Subject: /e an XSS vulnerability')</script><

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
  - Error messages must not reveal potentially sensitive information

- 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.)

Operations Security

Interpretations of Operations Security

- **Military Operations Security** (**OPSEC**) is a process that identifies critical information related to military operations, and then executes selected measures that eliminate or reduce adversary exploitation of this information.
- **Commercial Operations Security** is to apply security principles and practices to computer and business operations.

This lecture focuses on commercial operations security

Due Diligence and Due Care

- In general, due diligence is to make necessary investigations in order to be well informed
- Information security due diligence is the process of investigating security risks
  - Risk assessment is an essential element of due diligence
- To show due care means that a company implements security policies, procedures, technologies and standards that balances the security risks.
- Practicing due diligence and due care together means that a company acts responsibly by taking the necessary steps to protect the company, it's assets, and employees

Security control categories

- Physical controls
  - Gates, guards, locks, surveillance
- Technical controls
  - Access control, encryption, network and system protection
- Administrative controls
  - Policies, procedures, awareness training

- Most aspects of security controls have been explained in previous lectures

Privilege management

- Need to know / Least Privilege
  - Access to *only* the information that required to perform duties.
  - Reduces risk but causes overhead and a barrier to innovation
- Separation of duties
  - High-risk tasks require different individuals to complete
  - Examples: Provision privileged-access; Change a firewall rule
- Job rotation
  - Move individual workers through a range of job assignments
  - Rotation provides control and reduces likelihood of illegal actions
- Monitoring of special privileges
  - Review activities of Network/System/ administrators

Access Management

- Policies, procedures, and controls that determine how information is accessed and by whom
  - User account provisioning
  - Privilege management
  - Password management
  - Review of access rights
  - Secure log on

Asset identification and management

- Tangible asset management
  - Type, location, status of all hardware
  - Version of all installed software and firmware
  - Patch status of software
  - Backup media for all software
- Data classification
  - Establish sensitivity levels
  - Establish handling procedures for each level
  - Creation, storage, transmittal, destruction

Patch management

1. Provide patch management infrastructure
   - Requires procedures, staff end computing environment
2. Research newly released patches
   - Compatibility issues, authenticity and integrity of patches
3. Test new patches on isolated platforms
   - Patches often break functions, so better find out first
4. Provide procedures for rollback
   - Always have the possibility to return to previous status
5. Deploy patches to production platforms
   - Progressive , from least sensitive to most sensitive systems
6. Validate, log and report patching activities

Records Retention

- Policies that specify how long different types of records must be retained (minimums and maximums)
- Manage risks related to business records
  - Risk of compromise of sensitive information
  - Risk of loss of important information
  - E-Discovery
  - Regulation

Backups

- Protection against loss due to malfunctions, failures, mistakes, and disasters
- Activities
  - Data restoration when needed
  - Periodic testing of data restoration
  - Protection of backup media on-site
  - Off-site storage of backup media, consider:
    - distance,
    - transportation,
    - security and resilience of storage center

Data Destruction

- Ensure that discarded information is truly destroyed and not salvageable by either employees or outsiders

- Once information has reached the end of its need, its destruction needs to be carried out in a manner proportional to its sensitivity
  - Zeroisation/wiping/shredding: Overwrite media with dummy data
  - Degaussing: Strong magnetic field that reorients atoms on media
  - Physical destruction: melting, wrecking of media

End of Lecture