INF3510 Information Security

Lecture 12:
Development and Operations Security

Audun Jøsang

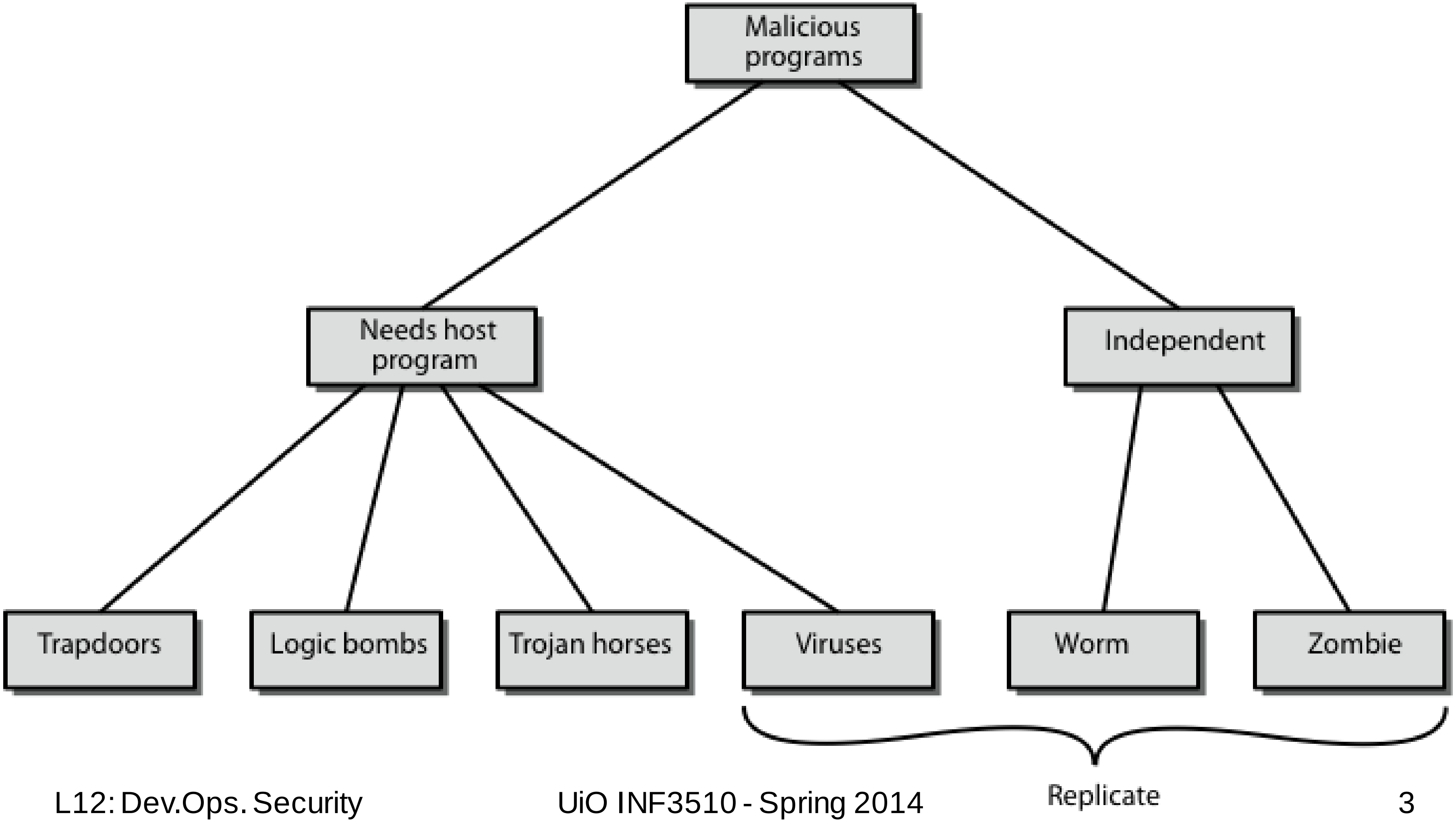
University of Oslo

Spring 2014

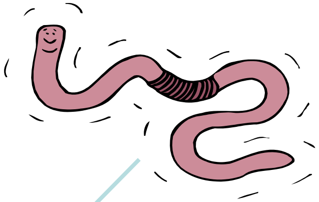
Outline

- Software Development Security
  - Malicious Software
  - Attacks on applications
  - Secure Development Lifecycle
- Operations Security

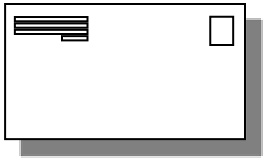
Malicious Software



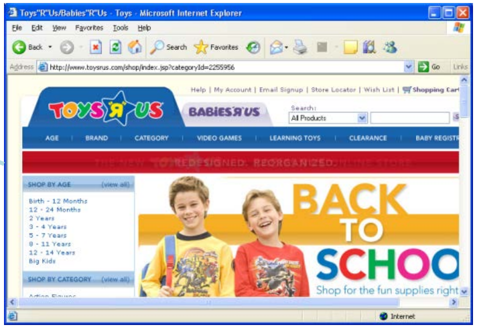
How do computers get infected ?



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



Executing an attachment



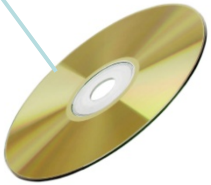
Toys"R"Us/Babies"R"Us - Toys - Microsoft Internet Explorer

Address http://www.toysrus.com/shop/index.jsp?categoryId=2255956

File Edit View Favorites Tools Help
Back
Search
Favorites
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 Trojan horse, spyware, virus, worm,
- 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

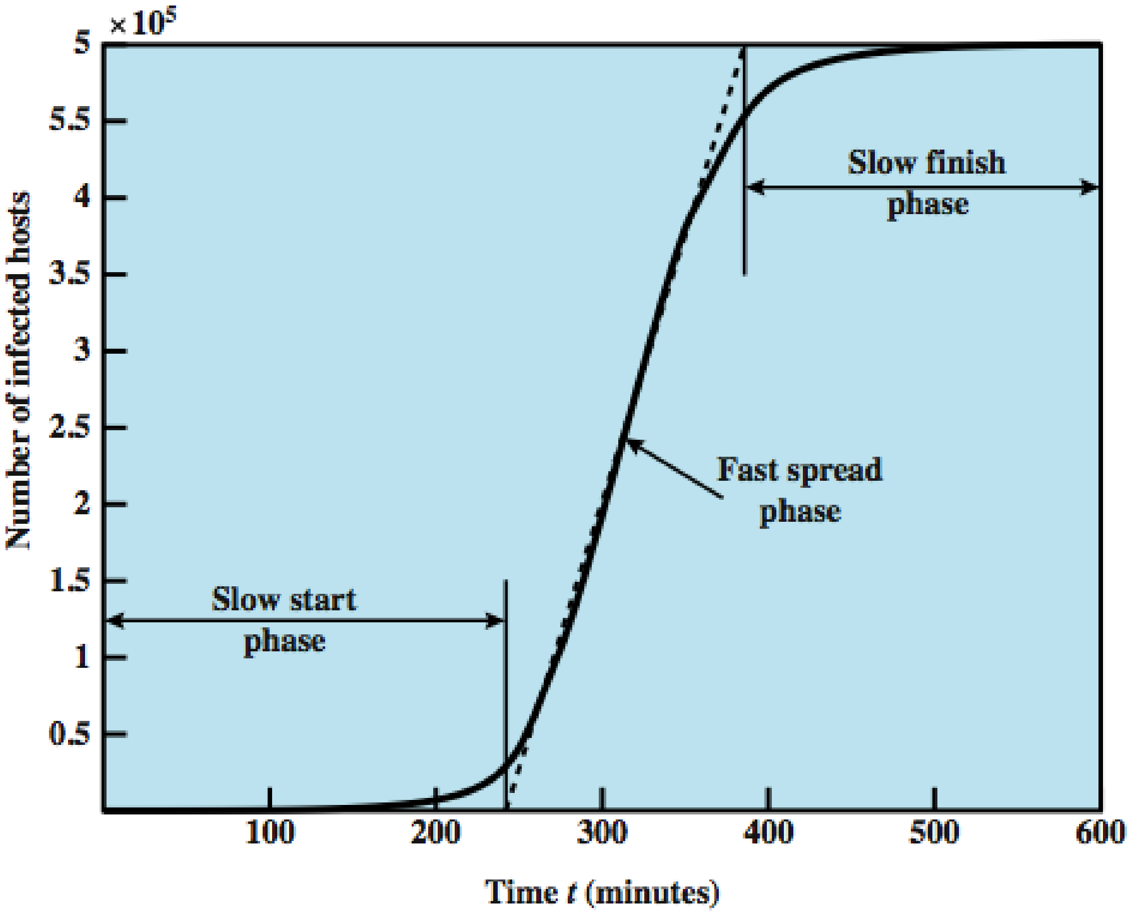
Some virus types

- Boot sector virus
- File infector virus
- Macro virus
- Encrypted virus
- Stealth virus
  - Uses techniques to hide itself
- Polymorphic virus
  - Different for every system
- Metamorphic virus
  - Different after every activation on same system

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
- Morris Worm, one of best know worms
  - released by Robert Morris in 1988
  - exploited vulnerabilities in UNIX systems
  - brought the whole Internet (of 1988) to standstill

Worm Propagation Speed



Worm Technology

- Multiplatform
- Multi-exploit
- Ultrafast spreading
- Polymorphic
- Metamorphic
- Transport vehicles
- Zero-day exploits

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

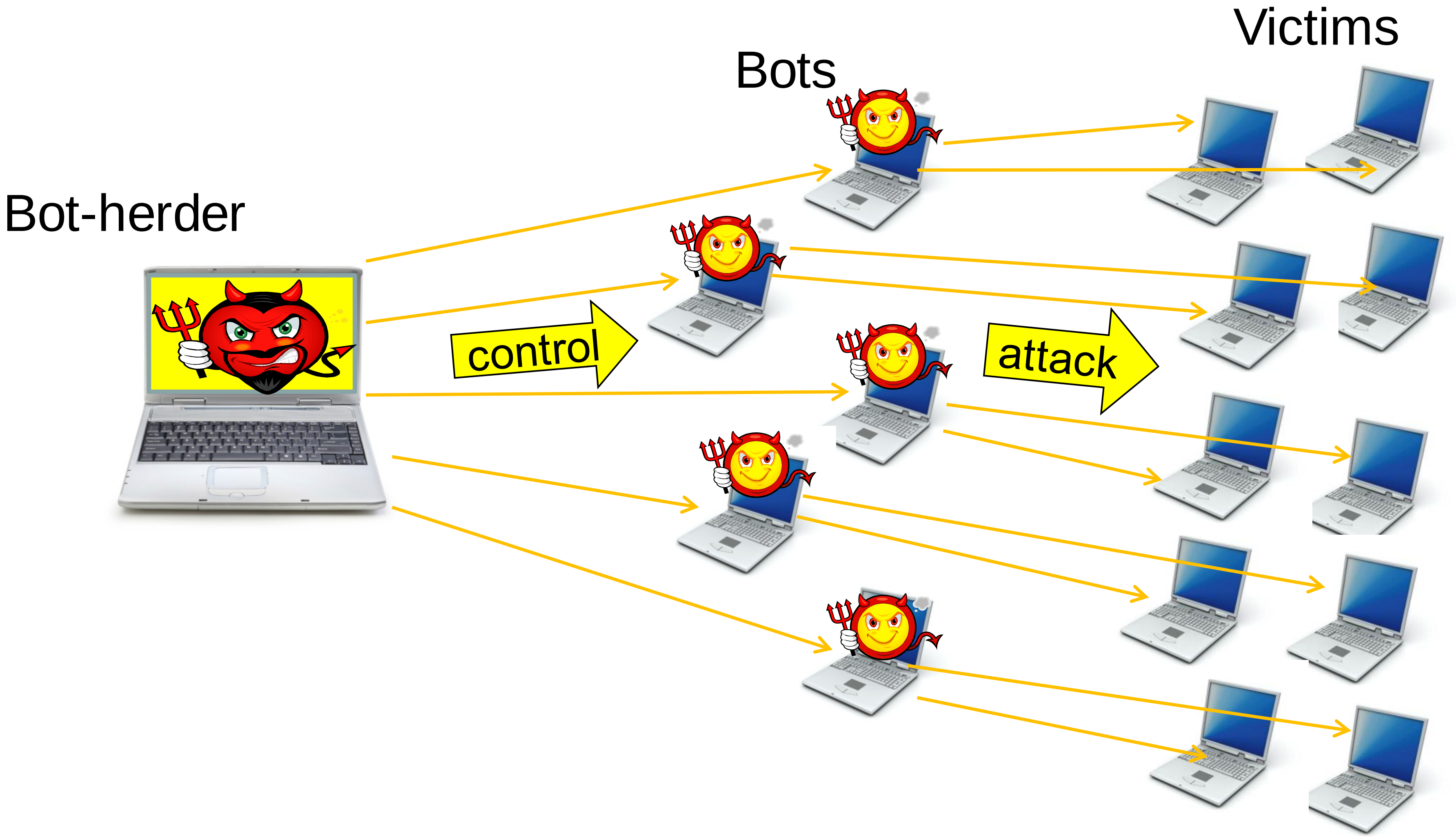
Worm Countermeasures

- 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

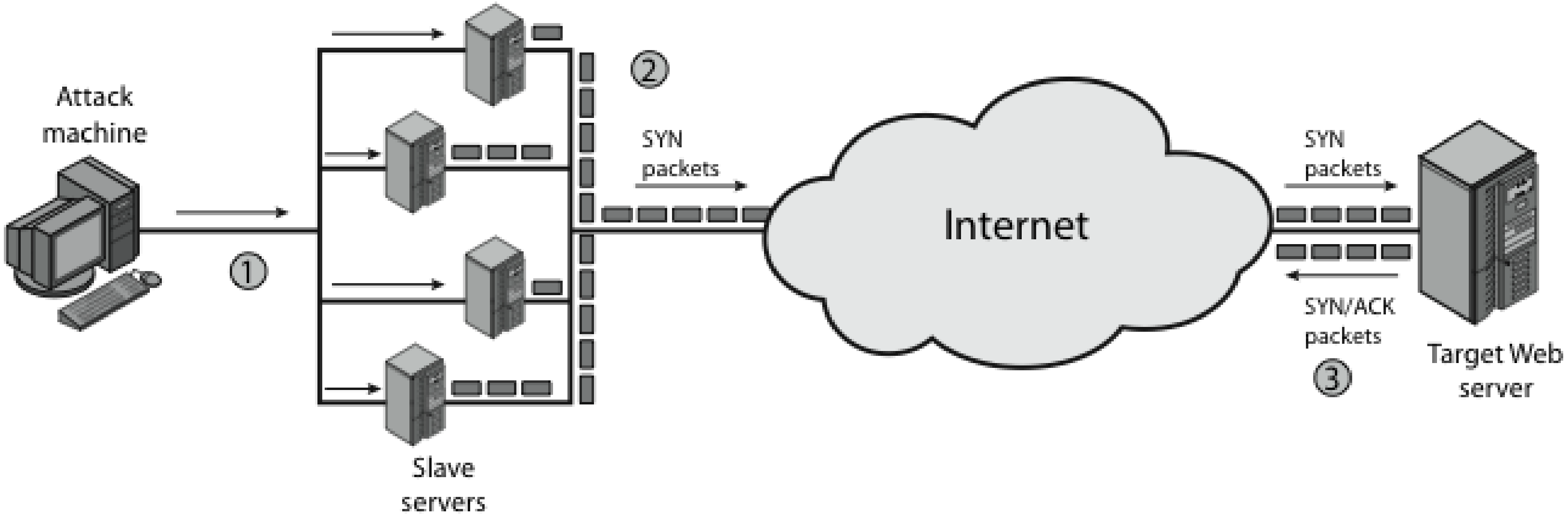
What is a botnet ?

- **A botnet** is a collection of computers infected with malicious software agents (robots) that can be controlled remotely by an attacker.
- Owners of bot computers are typically unaware of infection.
- Botnet controller is called a "bot herder" or "bot master"
- Botnets execute malicious functions in a coordinated way:
  - Send spam email
  - Collect identity information
  - Denial of service attacks
- A botnet is typically named after the malware used to infect
- Multiple botnets can use the same malware, but can still be operated by different criminal groups

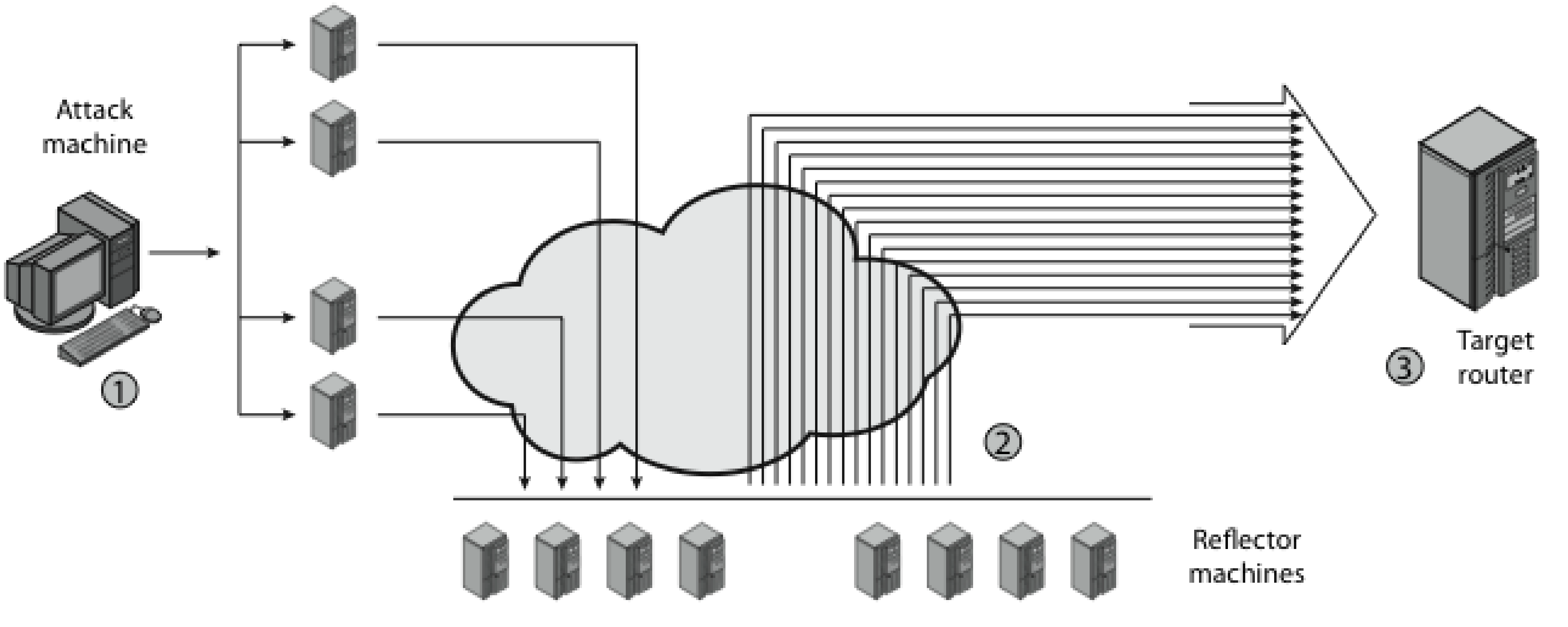
Botnet Architecture



Distributed Denial of Service Attack



(a) Distributed SYN flood attack

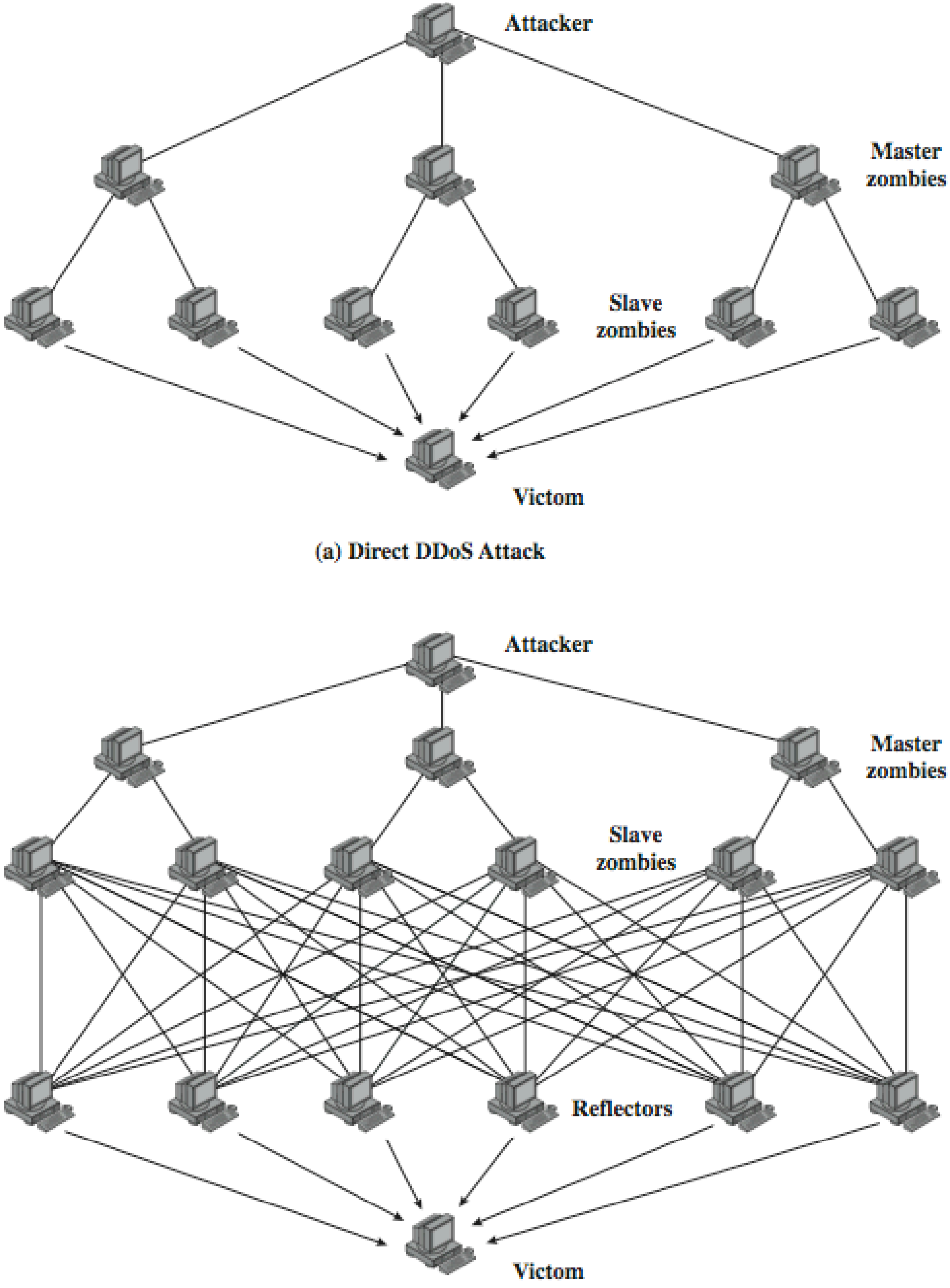


(a) Distributed ICMP attack

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

DDoS Flood Types

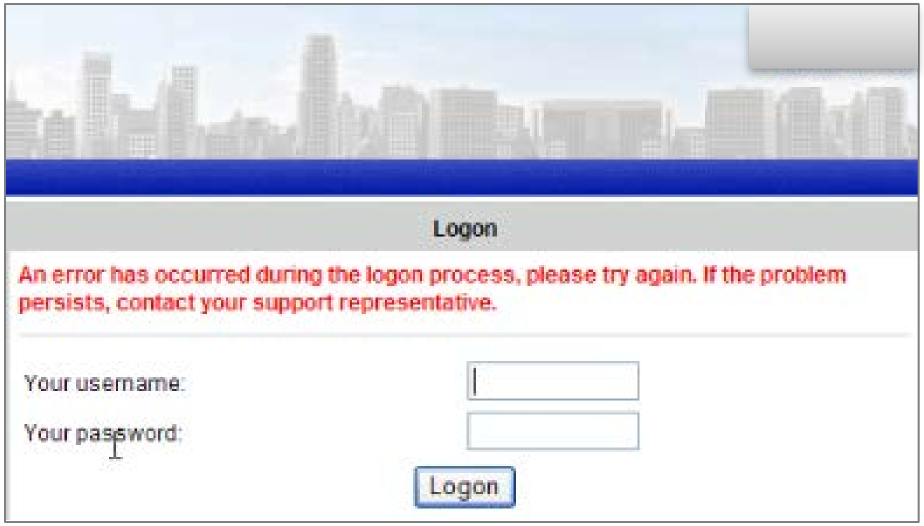


(a) Direct DDoS Attack

(b) Reflector DDoS Attack

Screen Injection by the Zeus bot

Browser NOT infected by Zeus:



Logon

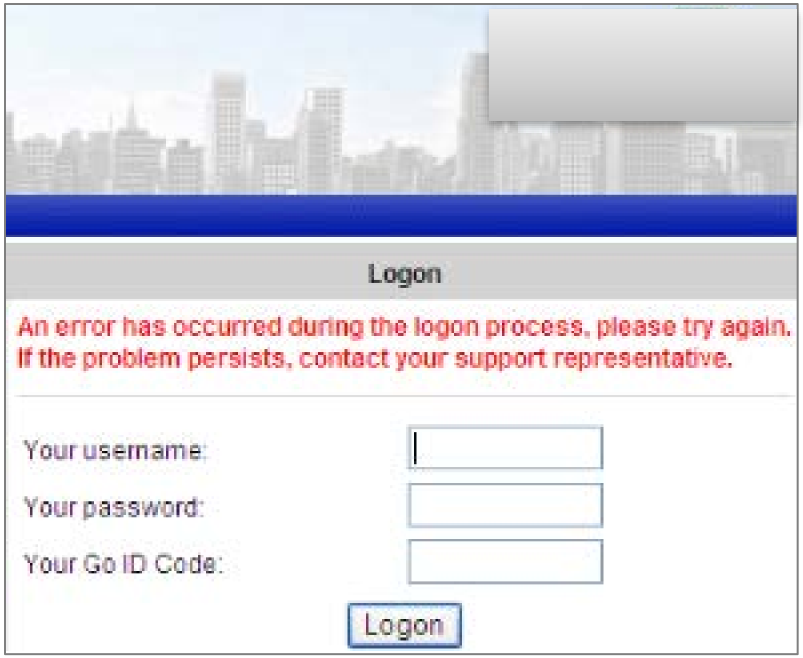
An error has occurred during the logon process, please try again. If the problem persists, contact your support representative.

Your username:

Your password:

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

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:

Zeus bot statistics 2010

- Criminals buy Zeus software to infect client computers
- Each attacker controls own set of infected computers
  - Each set of infected computers is a separate Zeus botnet
- 784 Zeus botnets tracked by Zeus Tracker in 2010
- Estimated total of 1.6M bots in all Zeus botnets
- 1130 victim 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

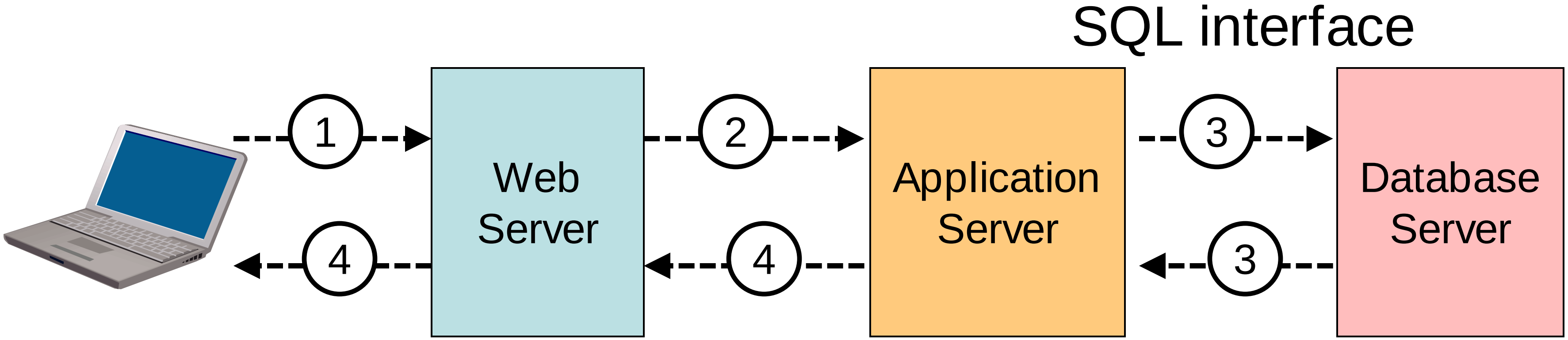
What is SQL?

- 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;
```

SQL at back-end of websites

1. Take input from a web-form via HTTP methods such as POST or GET, and pass it to a server-side application.
2. Application process opens connection to SQL database.
3. Query database with SQL and retrieve reply.
4. Process SQL reply and send results back to user.



What is SQL Injection?

- Misinterpretation of data input to database system
  - Attacker disguises SQL commands as data-input
  - Disguised SQL commands = ‘injected’ SQL commands
- With SQL injection, an attacker can get complete control of database
  - no matter how well the system is patched,
  - no matter how well the firewall is configured,
- Vulnerability exists when web application fails to sanitize data input before sending to it database
- Flaw is in web application, not in SQL database.

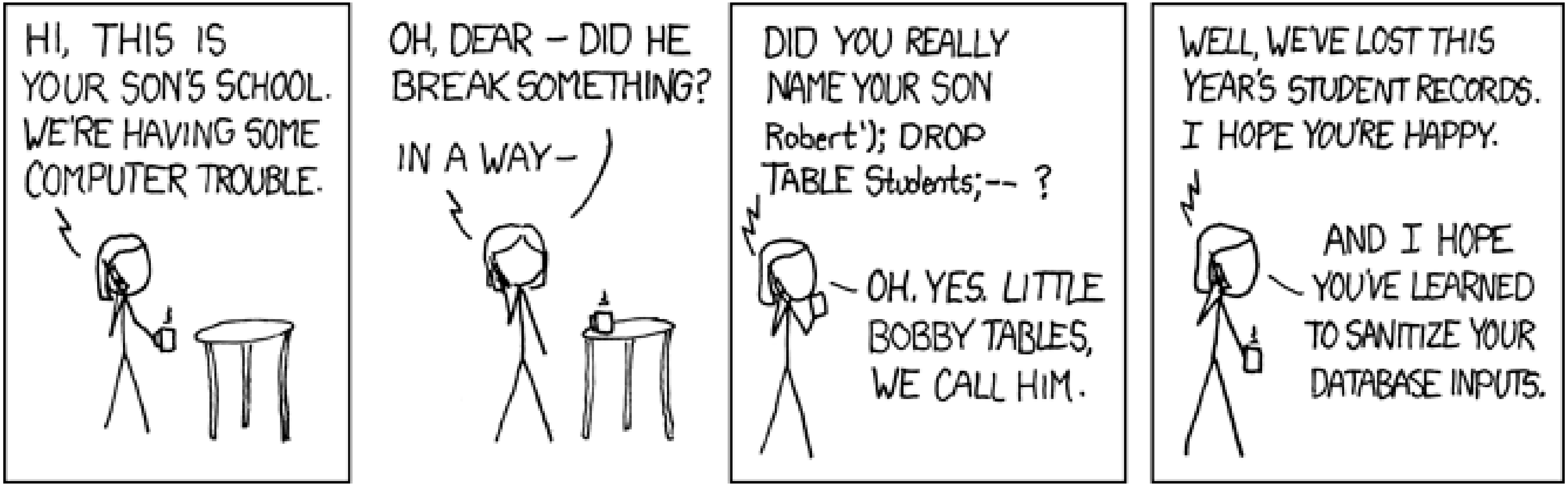
What is SQL Injection?

- For example, if user input is “**40 or 1 = 1**”

```
select ProductName from products where
  ProductID = 40 or 1 = 1
```

- 1=1 is always TRUE so the “where” clause will always be satisfied, even if ProductID ≠ 40.
- All product records will be returned.
- Data leak.

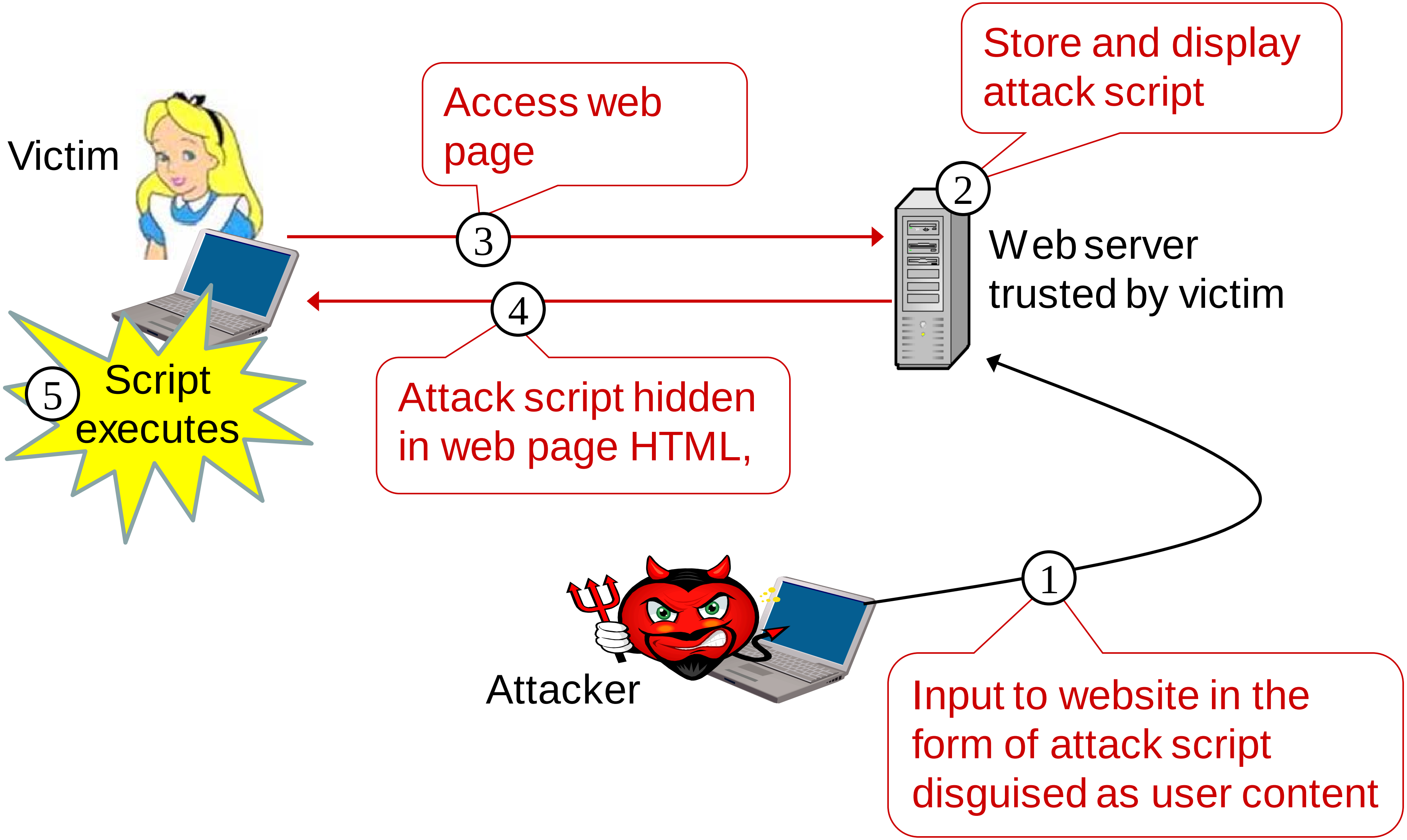
XKCD – Little Bobby tables


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

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.

Stored XSS



Input to website in the form of attack script disguised as user content

Store and display attack script

Access web page

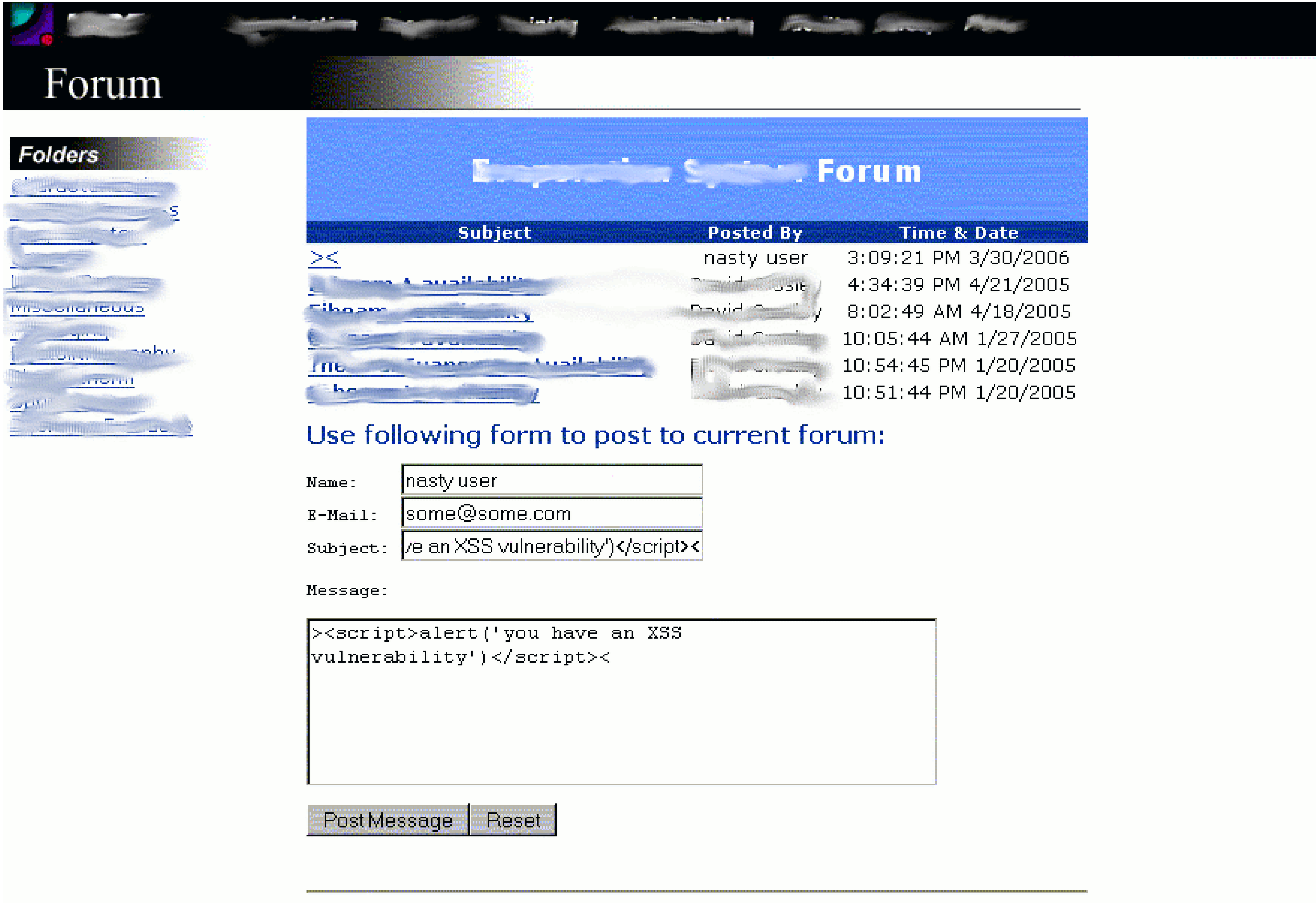
Attack script hidden in web page HTML,

Script executes

Stored XSS

- Stored, persistent, or second-order XSS.
- Data provided by users to a web application is stored persistently on server (in database, file system, …) and later displayed to users in a web page.
- Typical example: online message boards.
- Attacker uploads data containing malicious script to server.
- Every time the vulnerable web page is visited, the malicious script gets executed in client browser.
- Attacker needs to inject script just once.

XSS: Script Injection Demo



[redacted]

Forum

Folders

[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

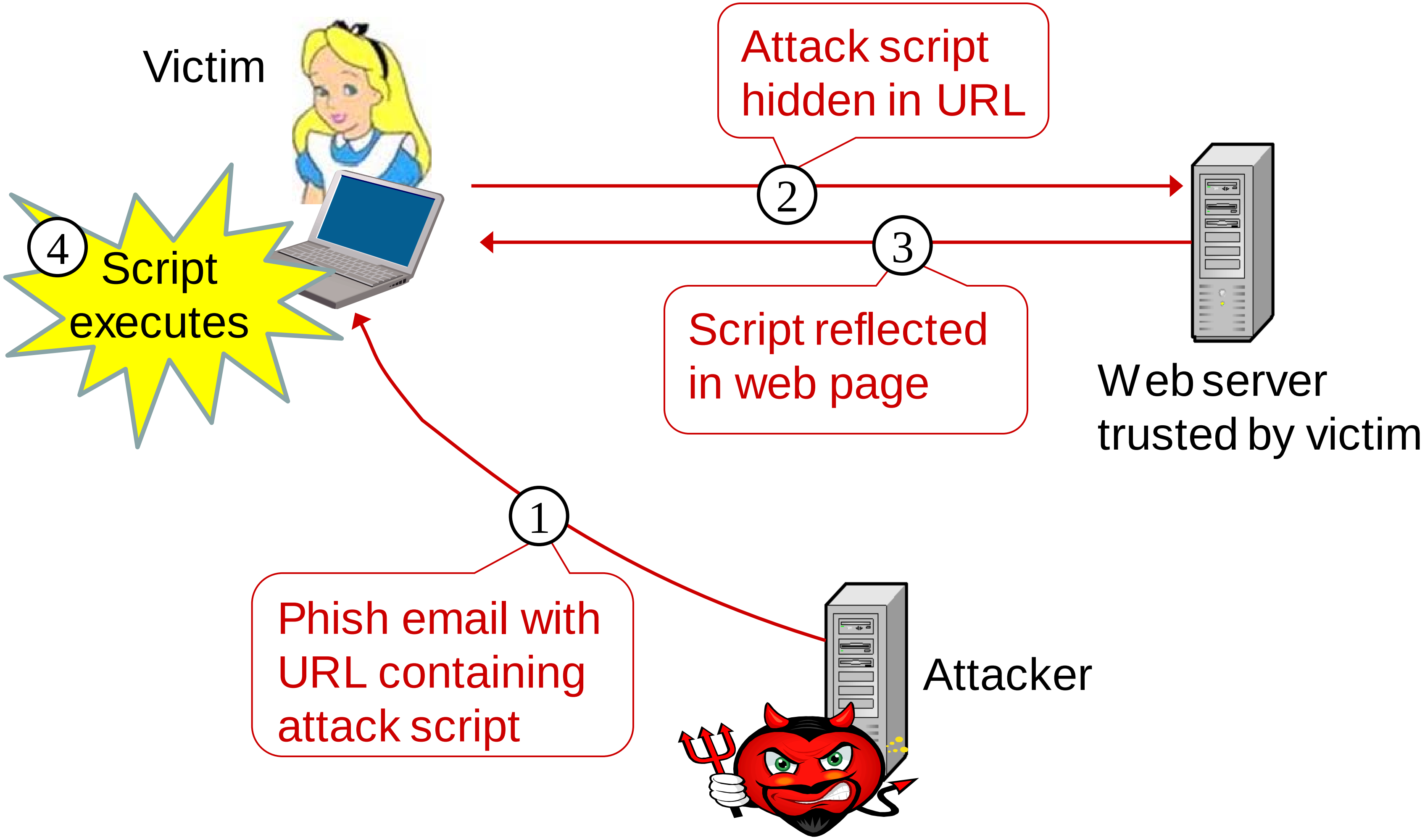
Use following form to post to current forum:

Name:	nasty user
E-Mail:	some@some.com
Subject:	/e an XSS vulnerability')</script><

Message:

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

Reflected XSS



Victim
4
Script executes
2
Attack script hidden in URL
3
Script reflected in web page
Web server trusted by victim
1
Phish email with URL containing attack script
Attacker

Reflected XSS

- Data provided by client is used by server-side scripts to generate results page for user.
- User tricked to click on attacker’s link for attack to be launched; page contains a frame that requests page from server with script as query parameter.
- If unvalidated user data is echoed in results page (without HTML encoding), code can be injected into this page.
- Typically delivered via email, containing an innocently looking URL that contains a script.
  - E.g., search engine redisplays search string on the result page; in a search for a string that includes some HTML special characters code may be injected.

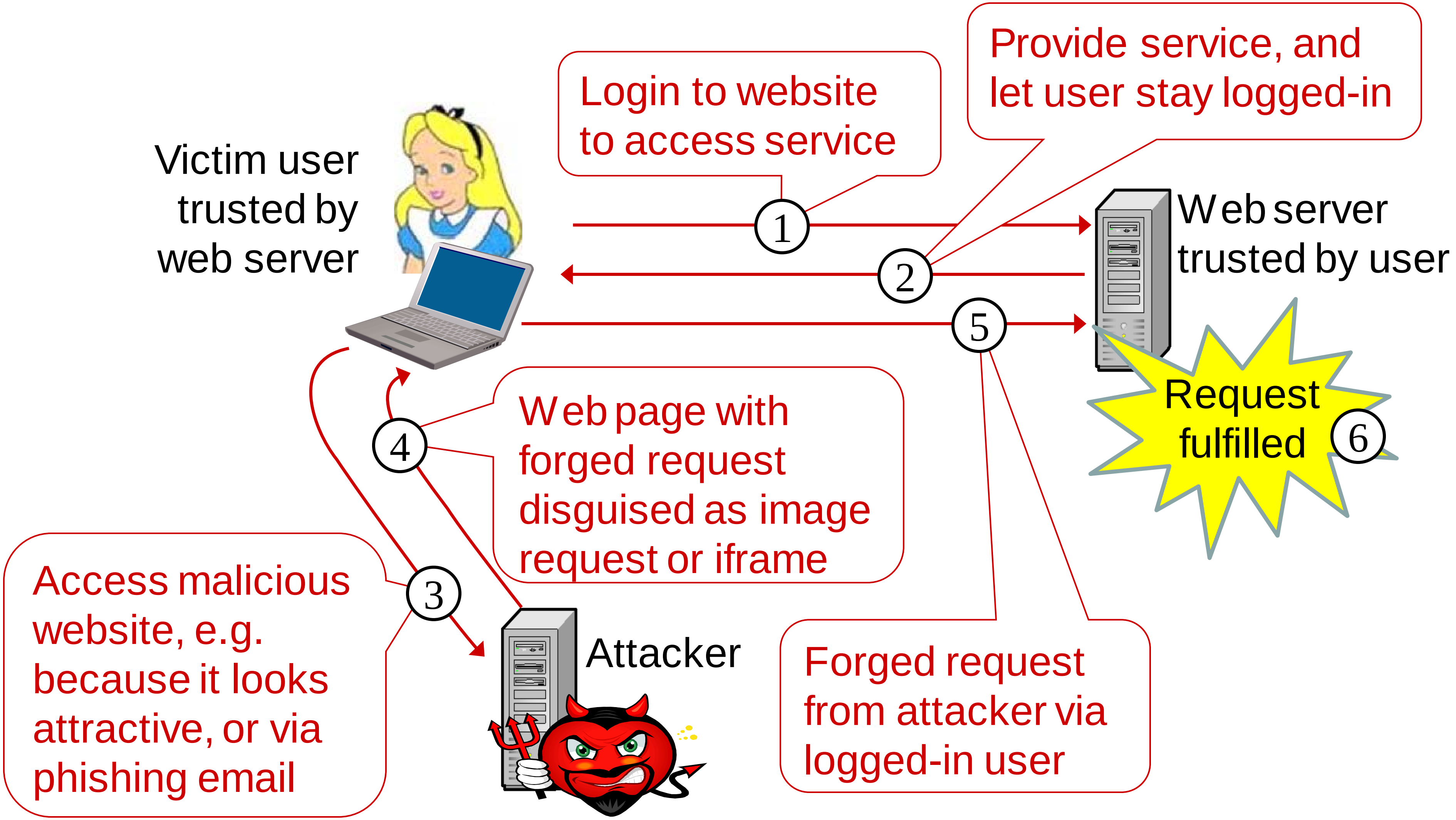
XSS – The Problem

- Ultimate cause of the attack: The client only authenticates ‘the last hop’ of the entire page, but not the true origin of all parts of the page.
- For example, the browser authenticates the bulletin board service but not the user who had placed a particular entry.
- If the browser cannot authenticate the origin of all its inputs, it cannot enforce a code origin policy.

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.)

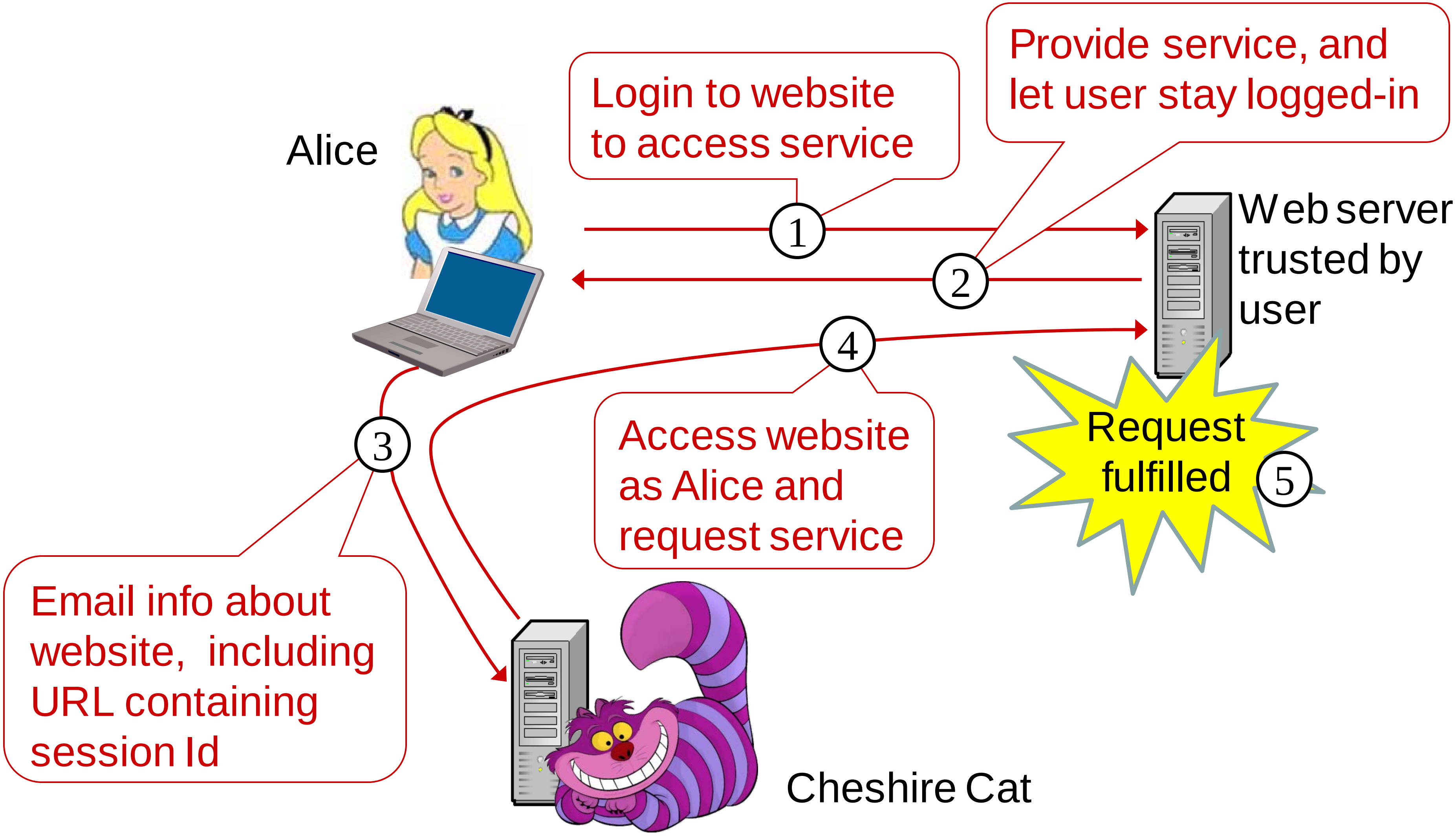
CSRF (Cross-Site Request Forgery)


Victim user trusted by web server
Login to website to access service
1
Provide service, and let user stay logged-in
2
Web server trusted by user
5
Request fulfilled
6
4
Web page with forged request disguised as image request or iframe
3
Access malicious website, e.g. because it looks attractive, or via phishing email
Attacker
Forged request from attacker via logged-in user

CSRF – Problem and Fix

- Users stay logged-in at websites even when not using them
  - Can be exploited by attackers sending fake requests via users
- Forged HTTP requests for a specific website that requires user login are hidden on attacker's webpage in the form of fake image requests, iframes or other elements.
- Browser accesses webpage and forwards forged requests.
- Preventing CSRF usually requires the inclusion of an unpredictable reference token (e.g. a random number) with each HTTP request to websites requiring login. Request tokens should at a minimum be unique per user session.
- Because the request token is unpredictable, the attacker is unable to create a forged request that will be accepted and fulfilled by the web server.

Broken Authentication and Session Mgmt


Alice
Login to website to access service
1
Provide service, and let user stay logged-in
2
Web server trusted by user
4
Access website as Alice and request service
Request fulfilled
5
3
Email info about website, including URL containing session Id
Cheshire Cat

Broken Authentication and Session Mgmnt Problem and Fix

- User authentication does not necessarily provide continuous authentication assurance
  - User authentication is only at one point in time
- Easy for developers to implement session control with a simple session Id which is passed in the URL
  - Unfortunately this can be misused
- Recommendations for session Id must be followed
  - E.g friom OWASP
- Examples of controls for session Id:
  - Link session Id to e.g. IP address, TLS session Id
- .

OWASP
The Open Web Application Security Project

- Non-profit organisation
  - Local chapters in most countries, also in Norway
- OWASP promotes security awareness and security solutions for Web application development.
- OWASP Top-10 security risks identify the most critical security risks of providing online services
  - The Top 10 list also recommends relevant security solutions.
- OWASP ASVS (Application Security Verification Standard) specifies requirements for application-level security.
- Provides and maintains many free tools for scanning and security vulnerability fixing

Top-10 Web Application Risks

1. Injection
2. Broken Authentication and Session Management
3. Cross-Site Scripting (XSS)
4. Insecure Direct Object References
5. Security Misconfiguration
6. Sensitive Data Exposure
7. Missing Function Level Access Control
8. Cross-Site Request Forgery (CSRF)
9. Using Components with Known Vulnerabilities
10. Unvalidated Redirects and Forwards

SDLC: Software Development Life Cycle

- SDLC model contains 6 basic stages:
  1. Requirements Specification
  2. Design
  3. Implementation
  4. Testing
  5. Deployment
  6. Maintenance
- Each SDLC model organises/integrates these basic stages in a specific way
  - XP (Extreme Programming), waterfall, etc.

Secure SDLC

- SDL – Secure Development Lifecycle
  - Used along with traditional/current software development lifecycle/techniques in order to introduce security at every stage of software development
- Three essential elements of secure SDLC
  1. Include security related tasks in each stage of the SDLC
  2. Security education for system engineers
  3. Metrics and accountability to assess security of system

Security Related Tasks of SDLC

1. Requirements Specs.
   - Risk analysis
   - Security Requirements
2. Design
   - Follow security design standards
   - Security Use Cases
3. Implementation
   - Follow secure coding practice
4. Testing
   - Penetration testing
   - Code review
   - Fuzzing
5. Deployment
   - Follow secure deployment practice
6. Maintenance
   - Analyse security incidents
   - Implement patches
   - Fix vulnerabilities

Fuzzing

- Malformed input should be handled in a consistent way by software and systems
  - Should be rejected with/without appropriate error message
- A software bug can lead to system to crash when processing malformed input
- Fuzzing is to generate many forms of malformed input and then to analyse resulting system crashes
  - The software location of a crash points to the location of the bug
- Some crashes can be exploited by attackers
  - Then the bug is a security vulnerability
- Developers and attackers use fuzzing to find vulnerabilities
- Infinitely many different malformed inputs
  - Impossible to test them all $\Rightarrow$ impossible to find all vulnerabilities

Operations Security

Meaning 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

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

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

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

Records Retention and Data Destruction

- Policies that specify how long different types of records must be retained (minimums and maximums)
- 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

Incident Management

- Policy: Define procedures for incident handling
  - Reporting: Who to tell ?
  - Who is responsible ?
  - Which systems can be taken offline ?
- Team: Define who is responsible
- Exercises: Red Team and Blue Team
- Incident response procedures:
  - Triage: Sort the trivial from the serious
  - Investigation and Containment
  - Analysis and tracking
  - Follow-Up

End of Lecture