Security Strategies in Web Applications and Social Networking

Chapter 6
Mitigating Web Site Risks, Threats, and Vulnerabilities

Learning Objective and Key Concepts

Learning Objective

- Compare and contrast Web-based risks.

Key Concepts

- Different types of traffic to Web sites
- Common vulnerabilities and attacks impacting Web applications
- Best practices for mitigating known Web application risks, threats, and vulnerabilities

Web Site Visitors

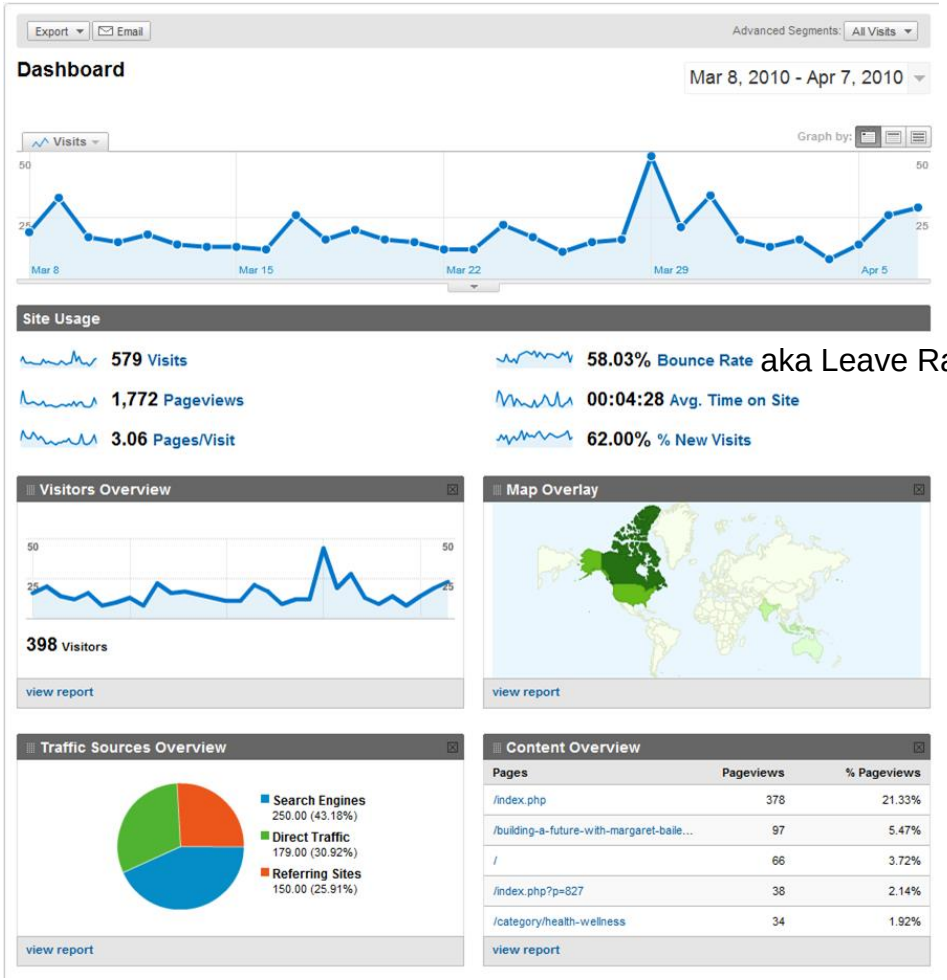
- Who visits your Web site?
- Who do you want to visit?

Google Analytics Results

helps u profile ur visitors

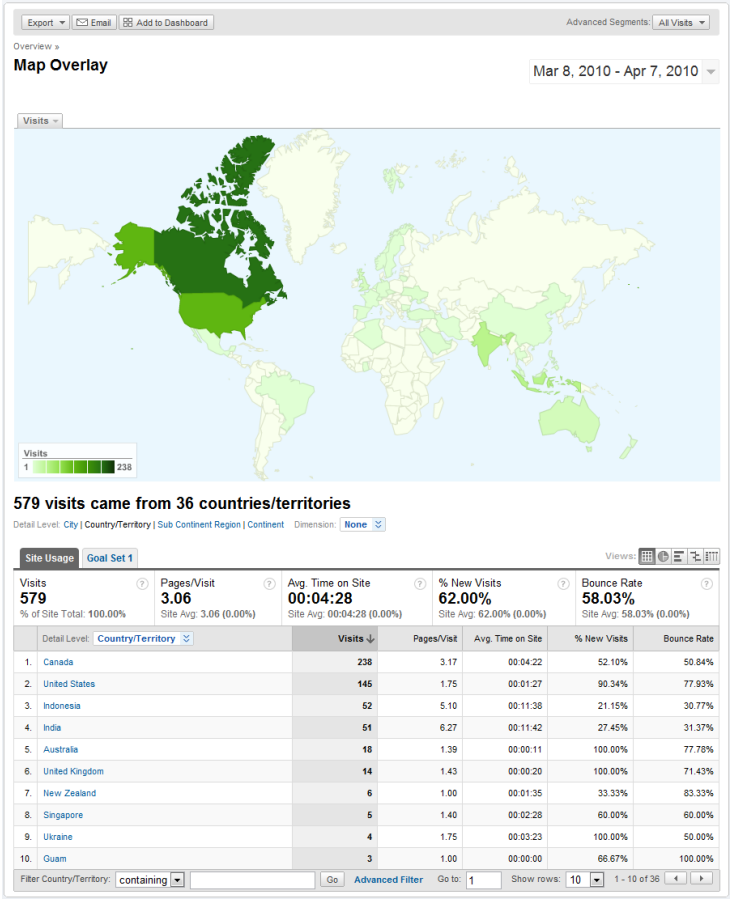
understand how ur website is doing

why use this?
- helps in profiling visitors to facilitate conversion into customers



aka Leave Rate

Google Analytics Map Overlay Statistics



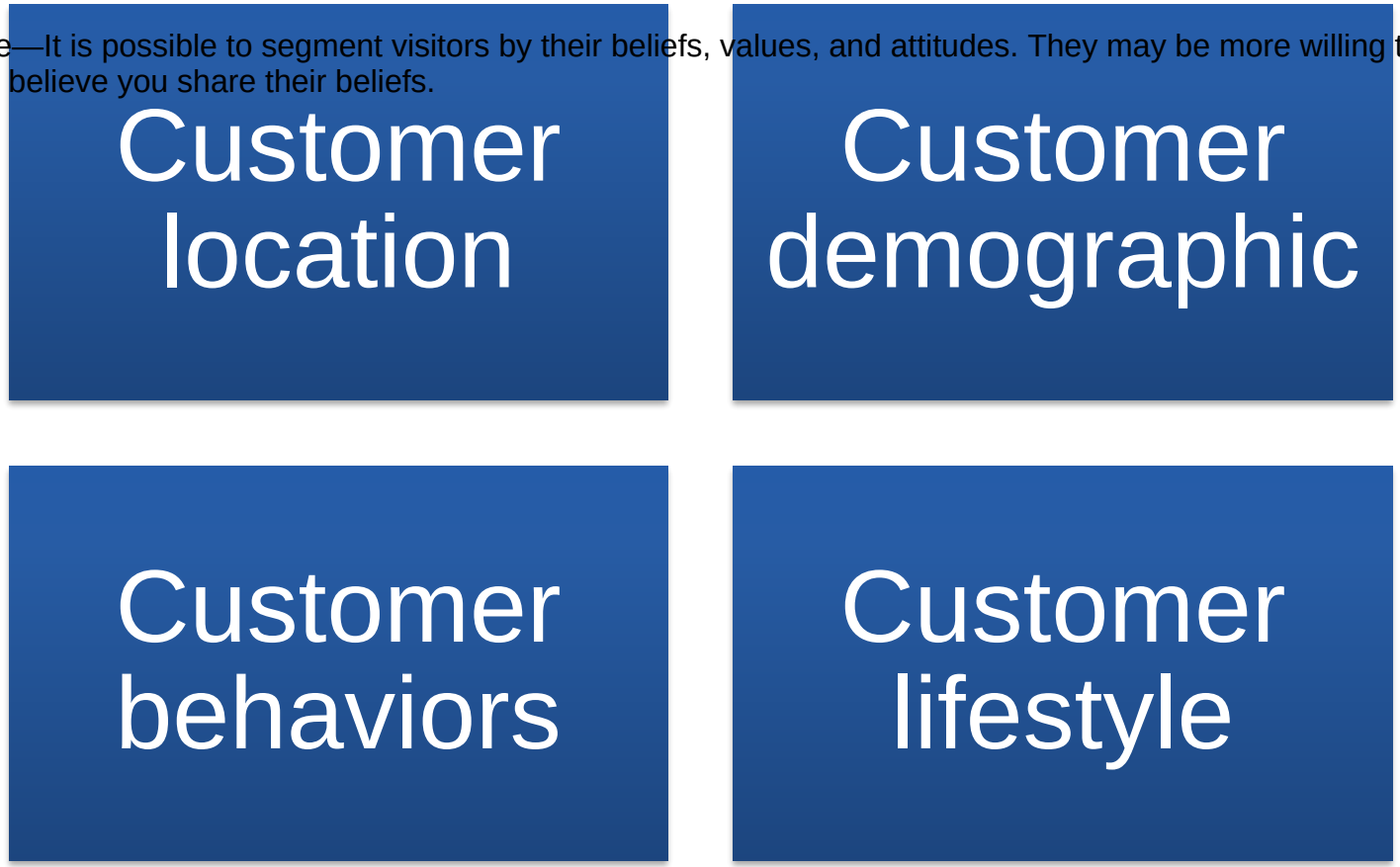
Customer location—Analytics software can segment visitors according to their location. Administrators can group visitors by country, state, city, territory, province, and more.

Market Segmentation

Customer demographic—Visitors can be segmented according to their unique demographic characteristics—age, education, income level, or more.

Customer behaviors—Visitors can be grouped and segmented according to their behaviors, such as surfing habits, online-spending trends, forum participation etc.

Customer lifestyle—It is possible to segment visitors by their beliefs, values, and attitudes. They may be more willing to become customers if they believe you share their beliefs.

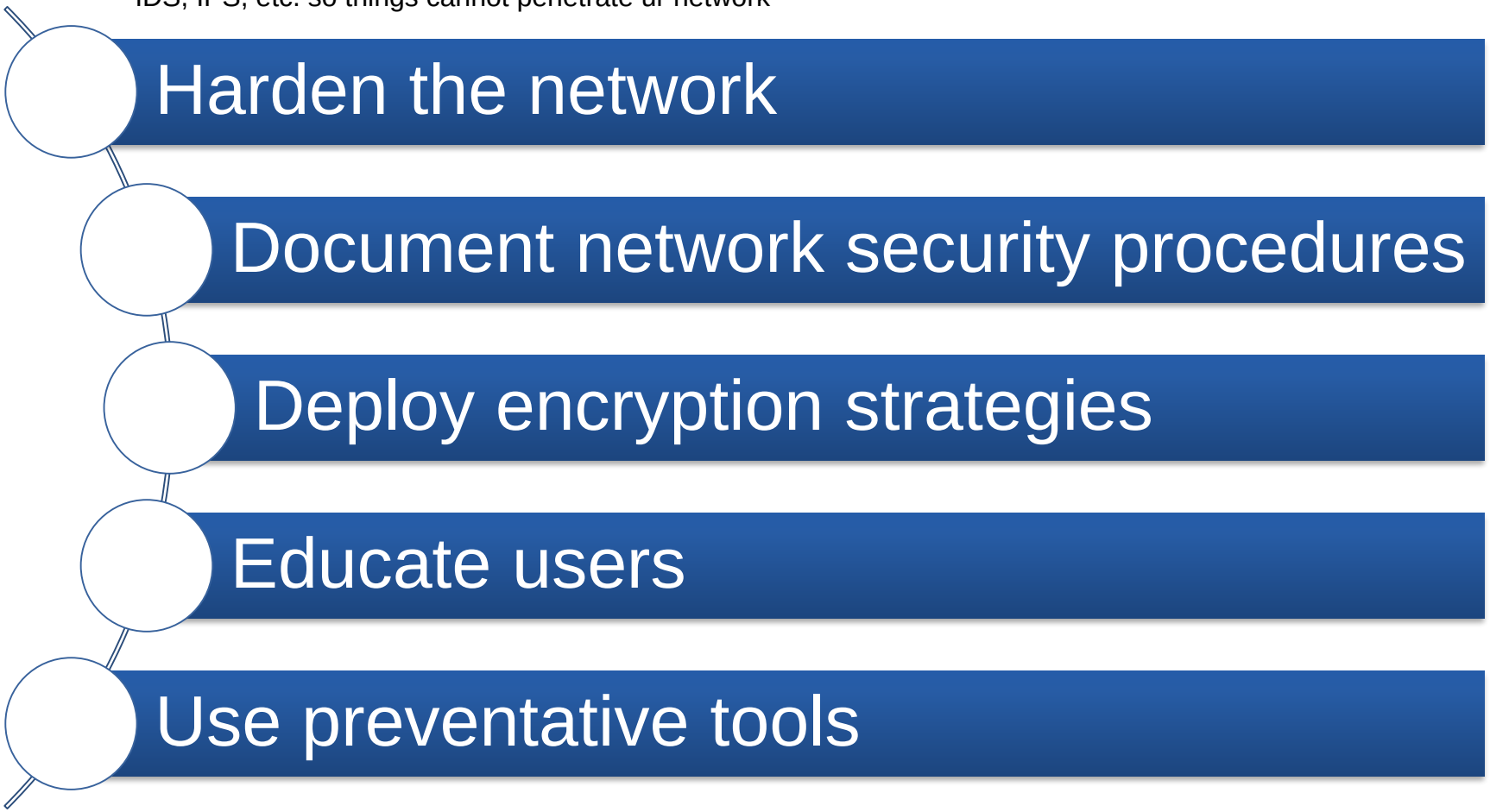


Best Practices for Mitigation

- When you start a Web application design, it is essential to apply threat risk modeling; otherwise you will squander resources, time, and money on useless controls that fail to focus on the real risks.

Best Practices for Mitigation (Cont.)

IDS, IPS, etc. so things cannot penetrate ur network



OWASP Overview

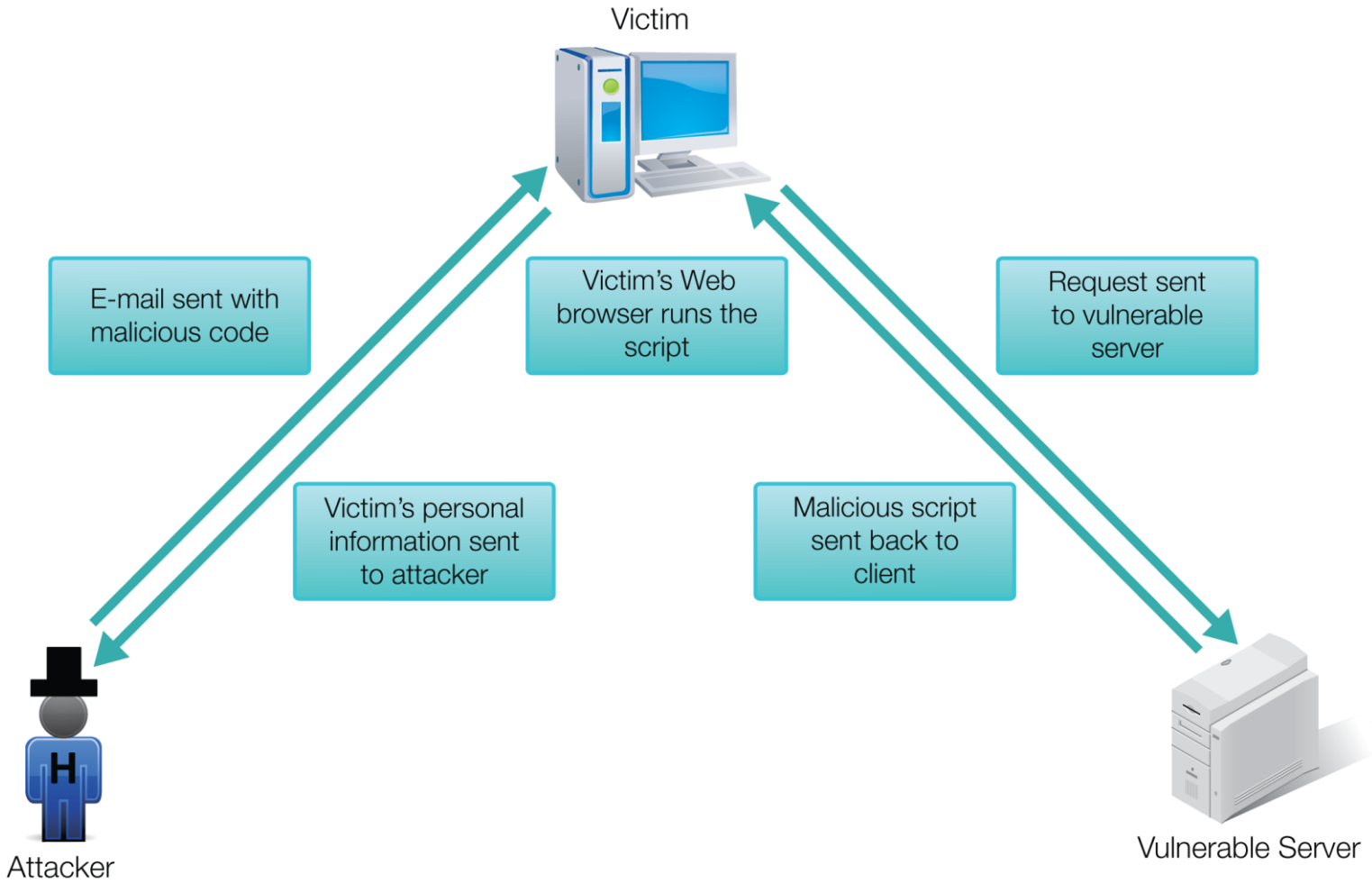
- The Open Web Application Security Project (OWASP) is a 501(c) 3 not-for-profit worldwide charitable organization focused on improving the security of application software.

OWASP Top 10

need to explain these and give examples on test2

1. XSS
   - A type of computer security vulnerability typically found in the Web applications that enables malicious attackers to inject client-side script into the Web pages viewed by other users.
2. Injection Flaws
   - Injection flaws allow attackers to relay malicious code through the Web application to another system.

Reflected XSS Attack



OWASP Top 10 (Continued)

3. Malicious File Execution

- Malicious file execution vulnerabilities are found in many applications. Developers will often directly use or concatenate potentially hostile input with file or stream functions, or improperly trust input files.

4. 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 Uniform Resource Locator (URL) or form parameter.

OWASP Top 10 (Continued)

5. Cross-Site Request Forgery
   - Also known as a one-click attack or session riding, it is a type of malicious exploit of the Web site whereby unauthorized commands are transmitted from a user that the Web site trusts.
6. Information Leakage and Improper Error Handling

no longer in top 10 now...
e.g. showing too much information when error occurs and displaying error page

   - Applications can unintentionally leak information about their configuration, internal workings, or violate privacy through a variety of application problems.

e.g. before web would say Username is wrong or password is wrong. so it gives u half of the info, u will know whether user or pw is right.
now say either username/password is wrong so they dont know. give less descriptive messages now to alleviate severity of this attack

OWASP Top 10 (Continued)

log out user periodically so session doesnt get jacked.
authentication force user to make strong pw, change periodically, cannot use old pws

7. Broken Authentication and Session Management

- Authentication and session management includes all aspects of handling user authentication and managing active sessions.

8. Insecure Cryptographic Storage

- Protecting sensitive data with cryptography has become a key part of most of the Web applications.

don't use weak crytography algorithms easily broken from online searches
store the keys in proper places and separately

OWASP Top 10 (Continued)

9. Insecure Communications

- Applications frequently fail to encrypt network traffic when it is necessary to protect sensitive communications. Encryption (usually SSL) must be used for all authenticated connections, especially the Internet-accessible Web pages, but backend connections as well.

if website security in one area is well designed but poor in another area, will still get data stolen; do proper design throughout!

OWASP Top 10 (Continued)

10. Failure to Restrict URL Access

- Frequently, the only protection for a URL that links to that page are not presented to unauthorized users. However, a motivated, skilled, or just plain lucky attacker may be able to find and access these pages, invoke functions, and view data.

Accepting User Input

- Accepting user input is extremely important to Web sites
  - Users are your customers.
- What visitors have to say can make your Web site more user friendly, therefore generating more users.

Accepting User Input (Cont.)

- Forums
- Web site feedback forms
- Online surveys

Summary

- Different types of traffic to Web sites
- Common vulnerabilities and attacks impacting Web applications
- Best practices for mitigating known Web application risks, threats, and vulnerabilities