Integrating Web Application Security into the IT Curriculum

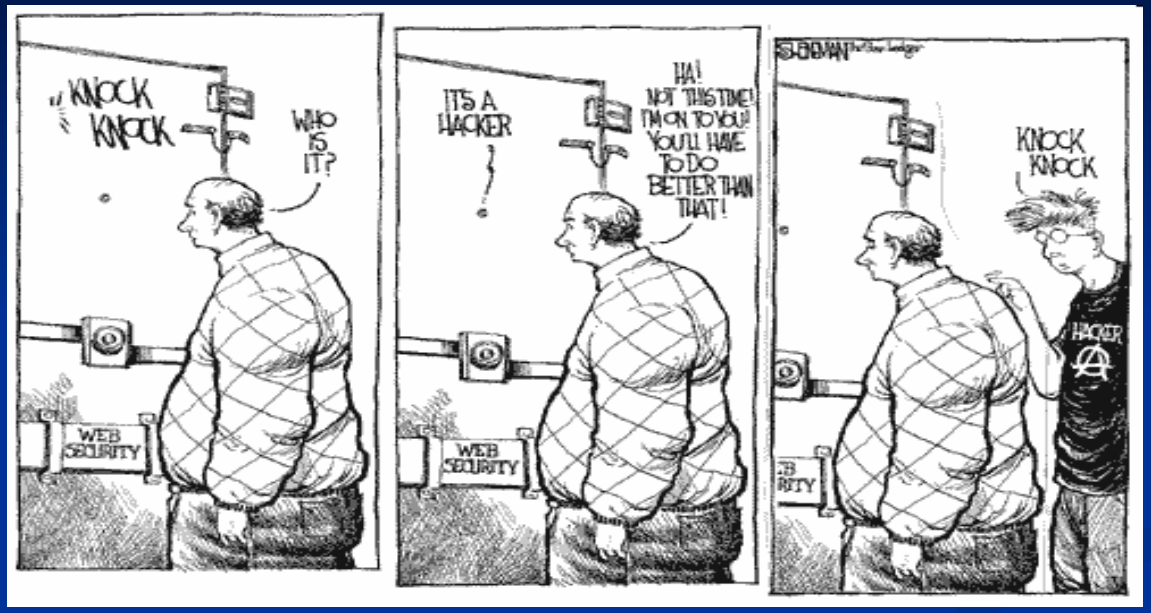
James Walden
Northern Kentucky University

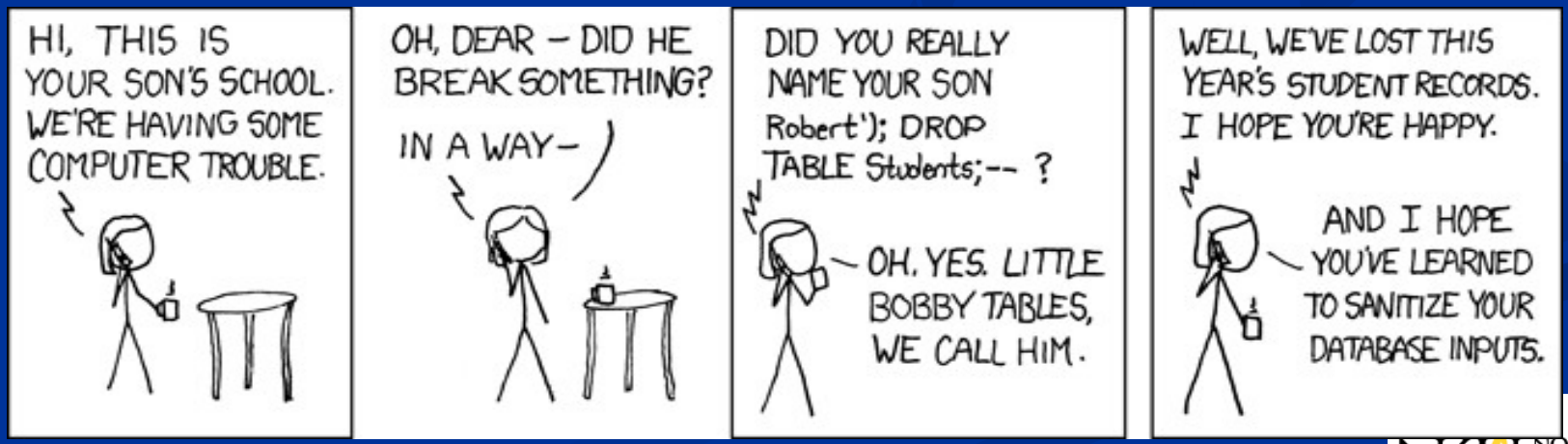
SIGITE 2008: 16-18 Oct

Topics

1. Why should we teach web application security?
2. What material do we need to cover?
3. How should we cover that material?
4. Where do we go from here?

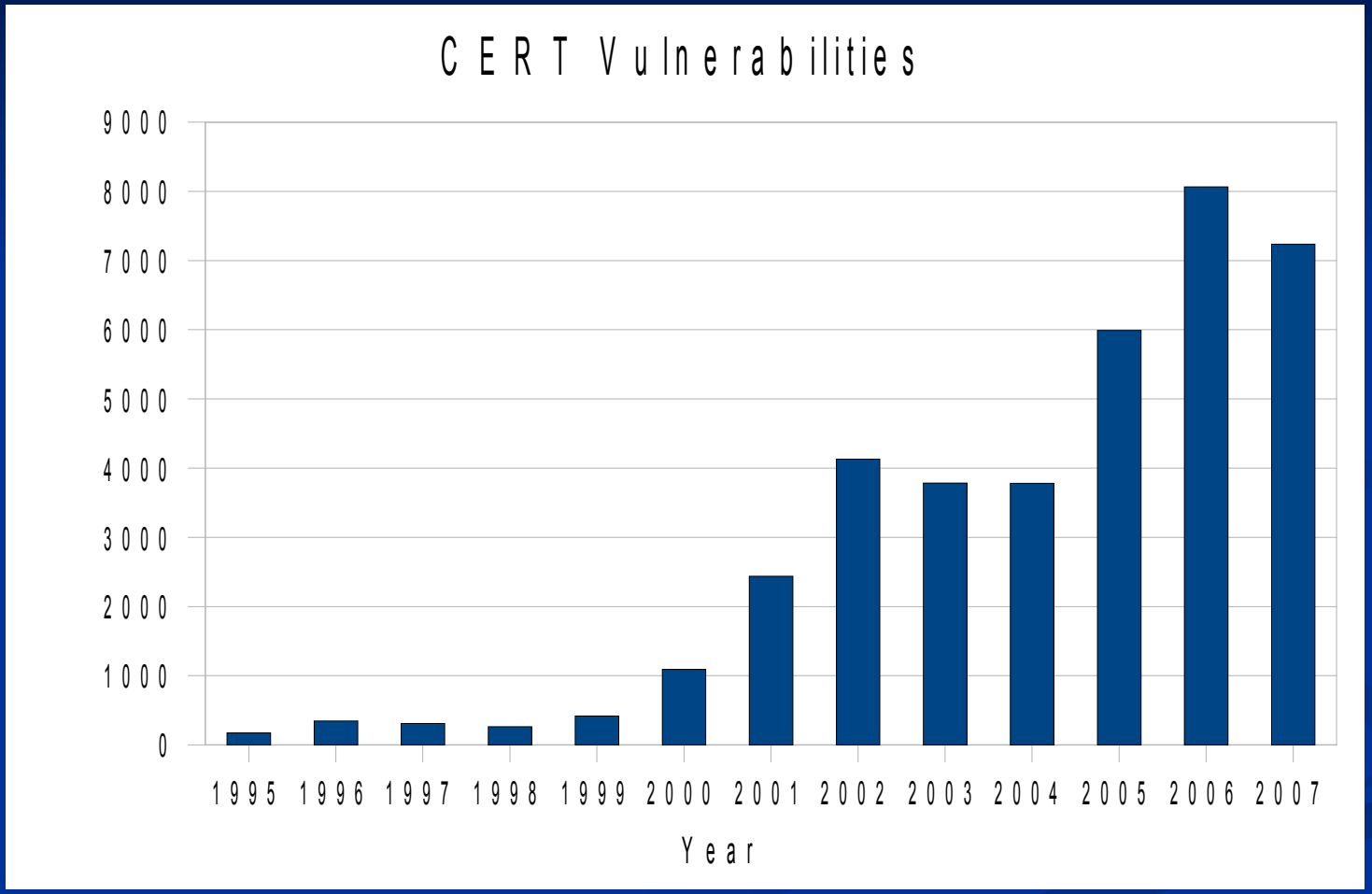
Is Web Hacking Really That Easy?



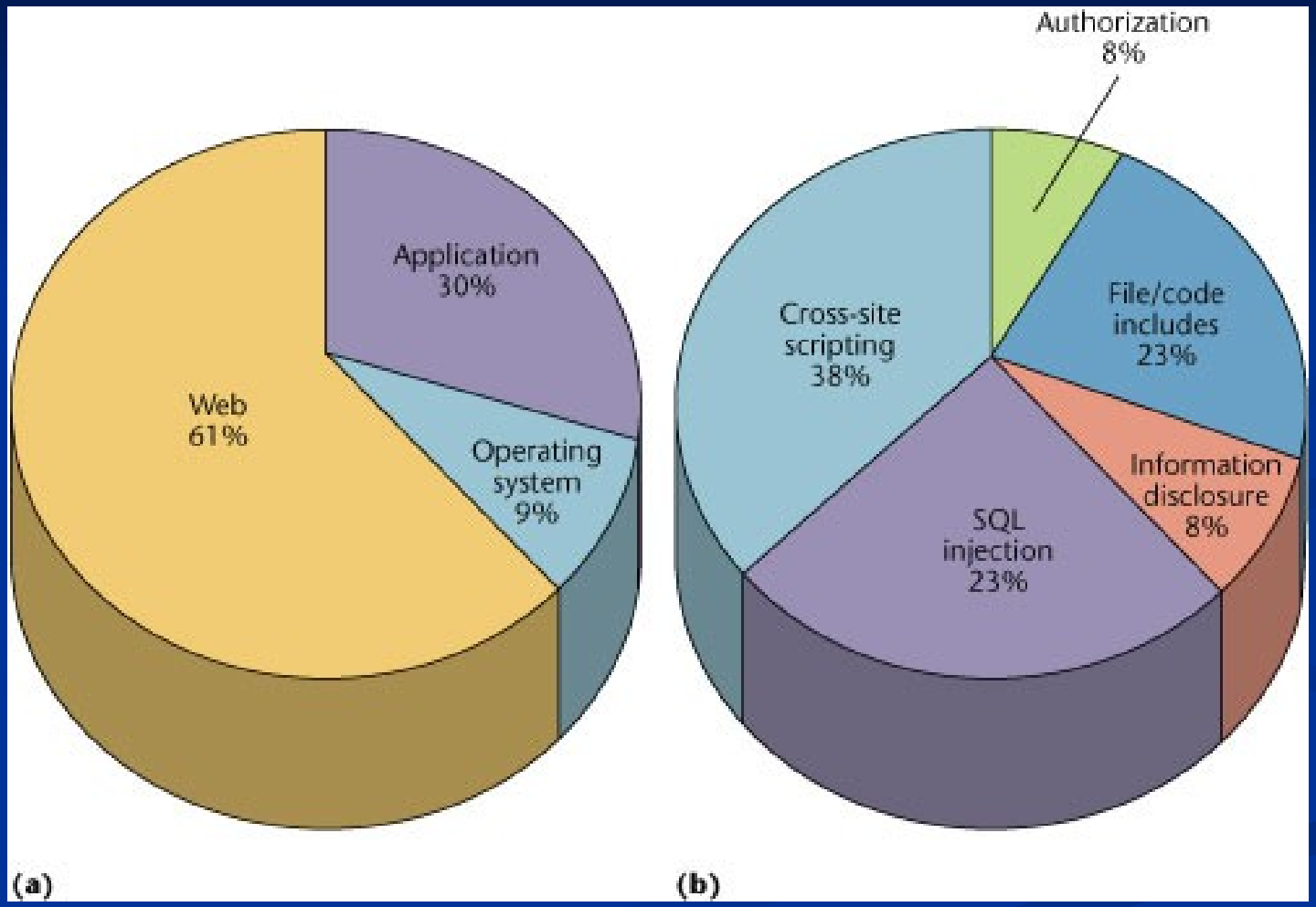


"Exploits of a Mom", XKCD

Vulnerability Growth

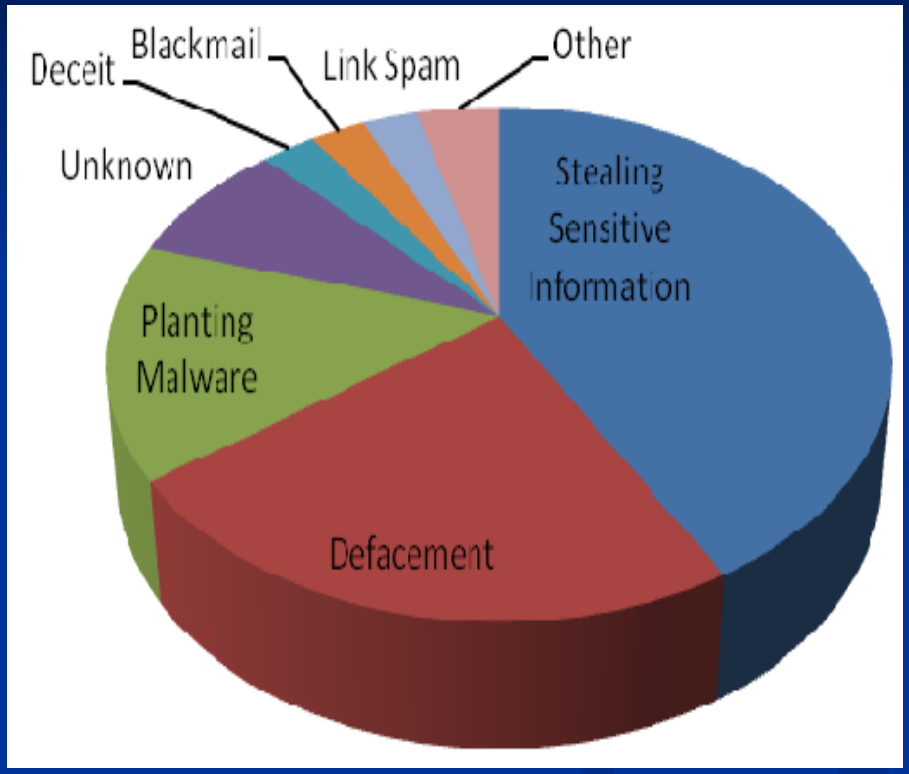


Web Vulnerabilities Dominate

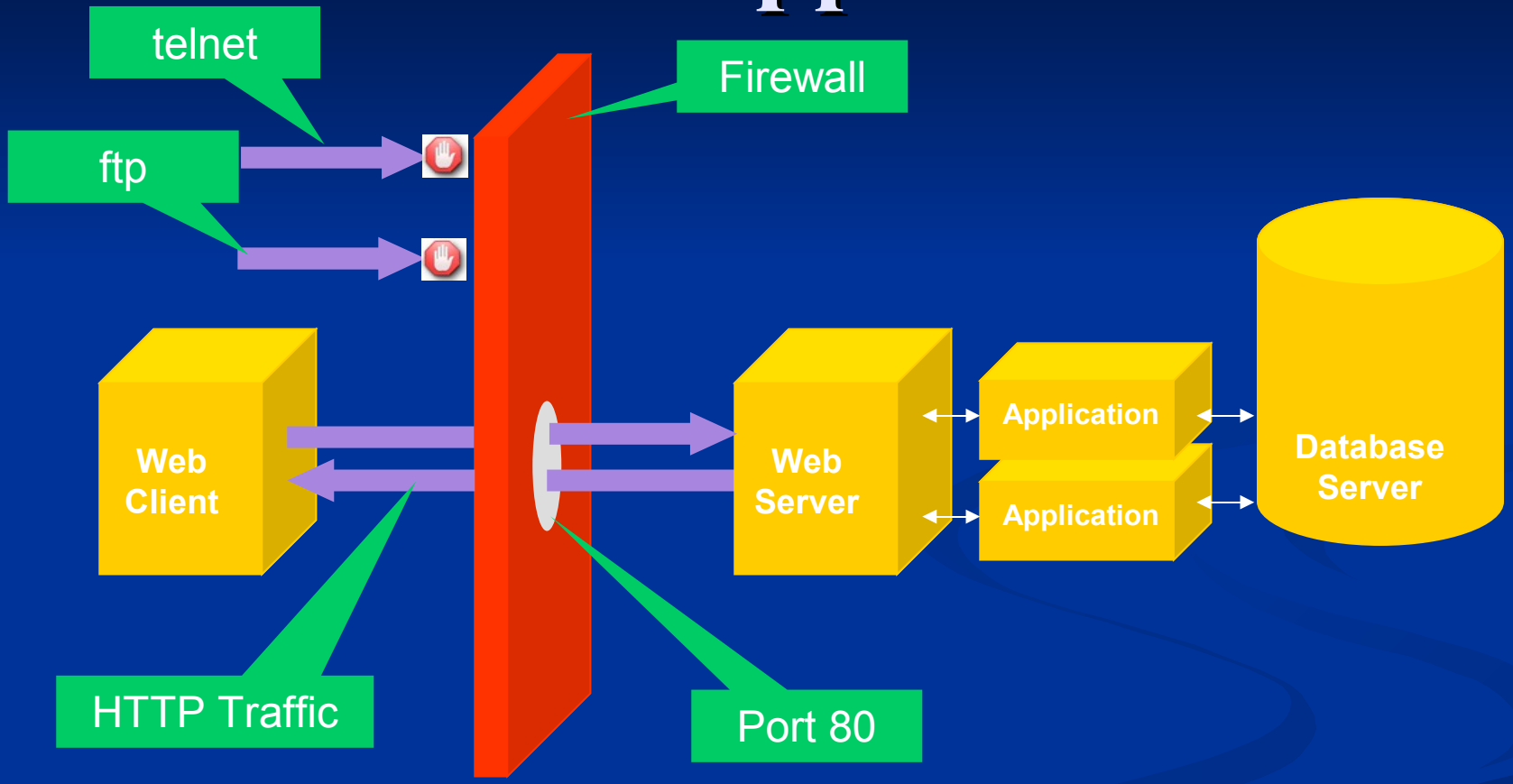


Reasons for Attacking Web Apps

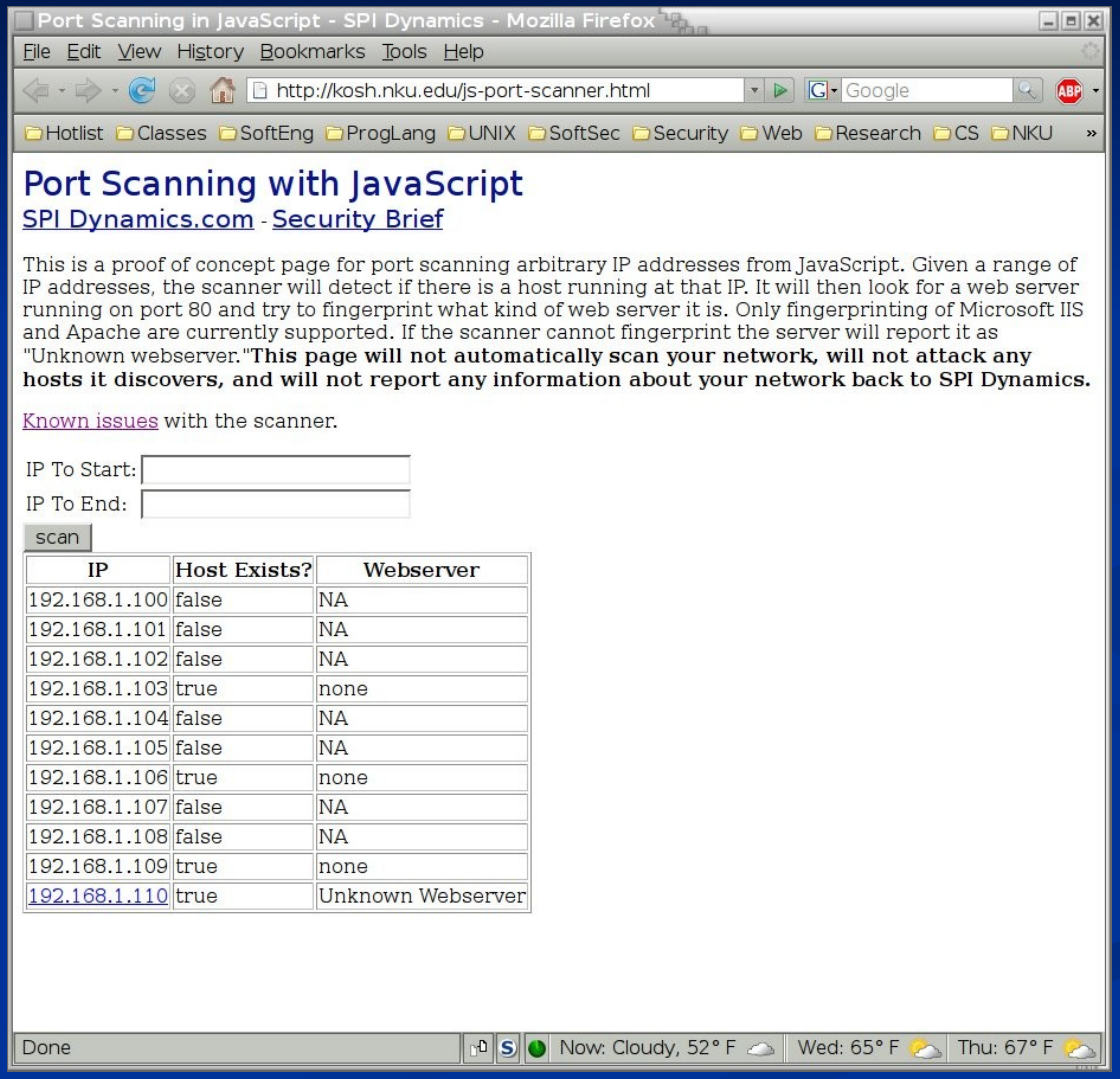
Attack Goal	%
Stealing Sensitive Information	42%
Defacement	23%
Planting Malware	15%
Unknown	8%
Deceit	3%
Blackmail	3%
Link Spam	3%
Worm	1%
Phishing	1%
Information Warfare	1%



Firewalls Don't Protect Web Apps



Browser Malware Bypasses Firewall



NKU NORTHERN KENTUCKY UNIVERSITY

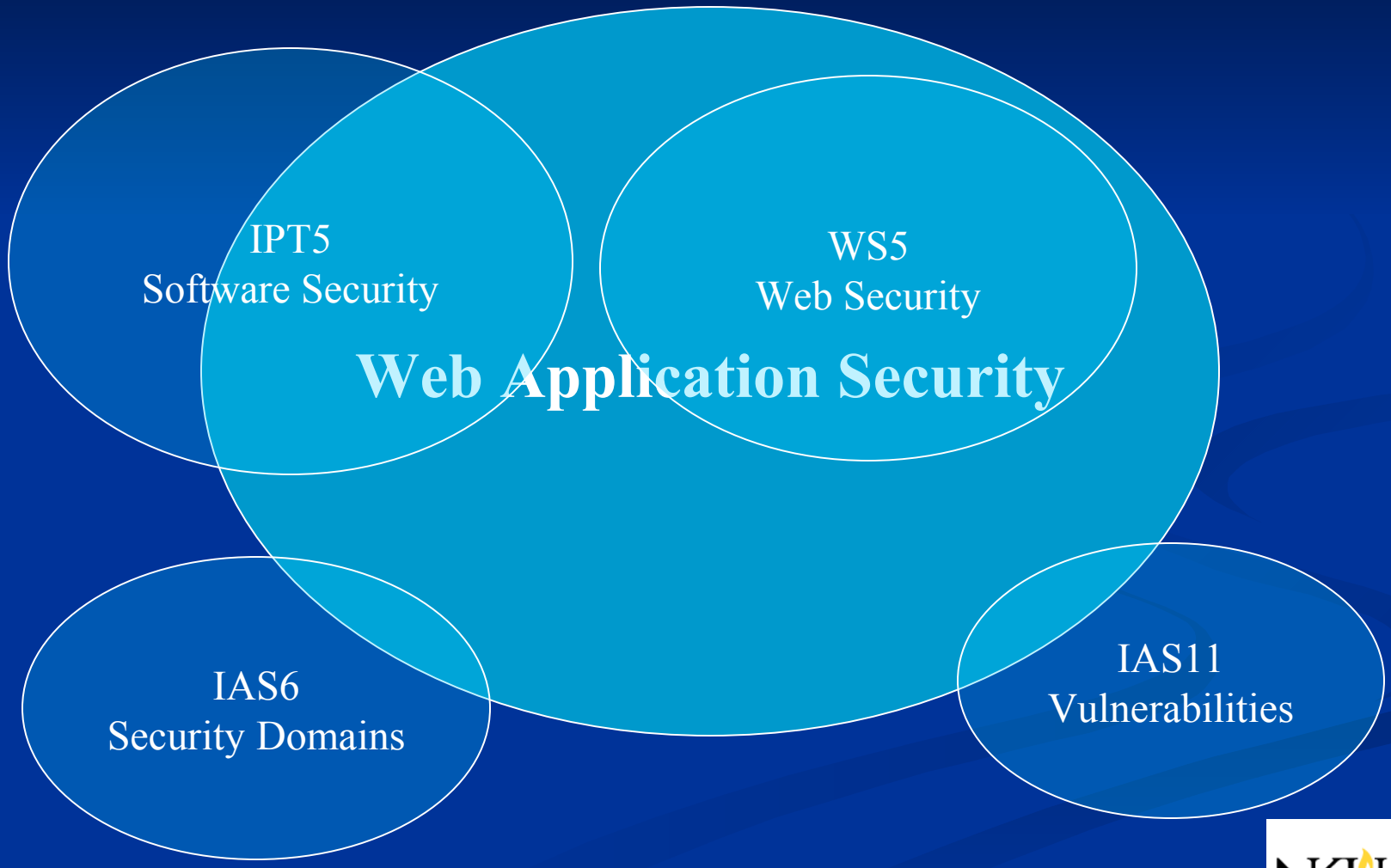
Goals

1. Identify and explain common vulnerabilities.
2. Explain security implications of client-side technologies like Javascript and ActiveX.
3. Detect security vulnerabilities in web applications using appropriate tools.
4. Design and implement web applications that do not contain common vulnerabilities.
5. Deploy and configure a web application in a secure manner.

Topic Outline

1. Web Application Input
2. Client-side Technologies
3. Input-based Attacks
4. Injection Attacks
5. Cross-site Attacks
6. Authentication
7. Secure Programming
8. Operational Security

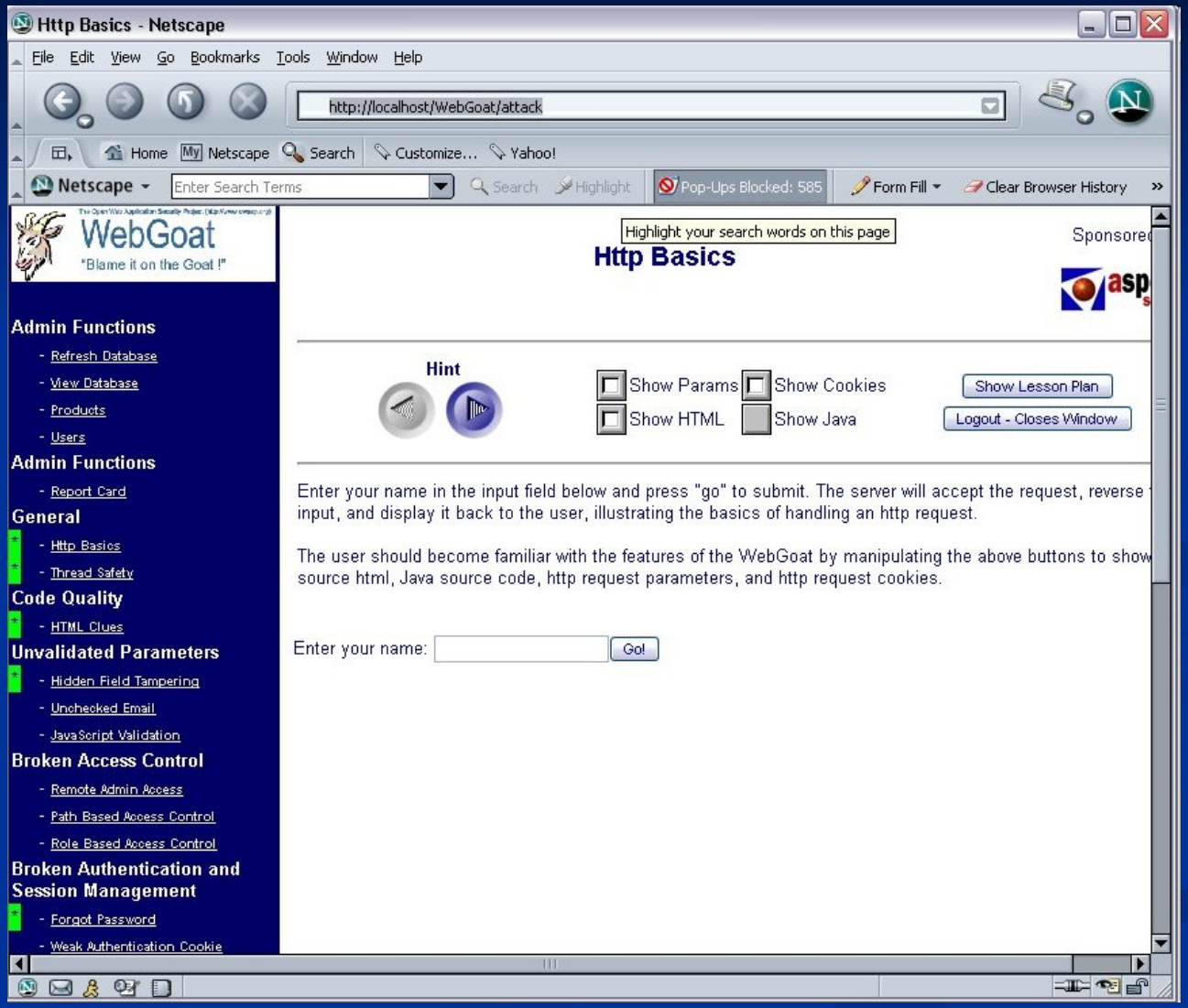
Web App Security in IT2005



Labs

1. WebGoat exercises on specific vulnerabilities.
2. Using a testing proxy to solve more advanced WebGoat exercises.
3. Assessing an application using a web vulnerability scanner.
4. Assessing a web application using a testing proxy.
5. Reviewing the code of an application using a static analysis tool.
6. Deploying a web application firewall.
7. Participating in the international CTF competition.

WebGoat



Tools

Web Proxies

MileSCAN
Web Security Auditor

PortSwigger.net
web application security

Web Application Firewalls

modsecurity
www.modsecurity.org

Vulnerability Scanners

Web Inspect ™

acunetix

Static Analysis

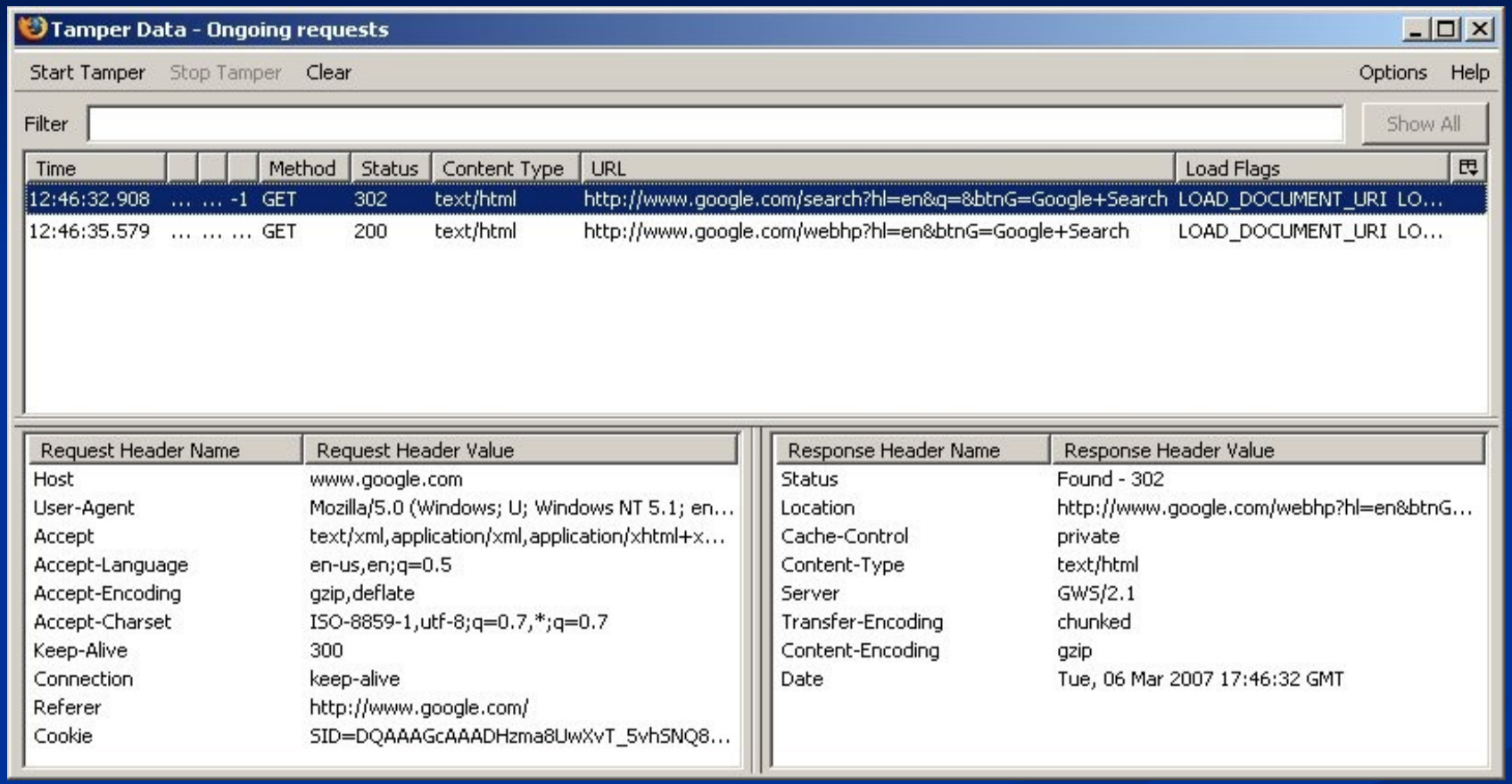
FORTIFY SOFTWARE

Klocwork

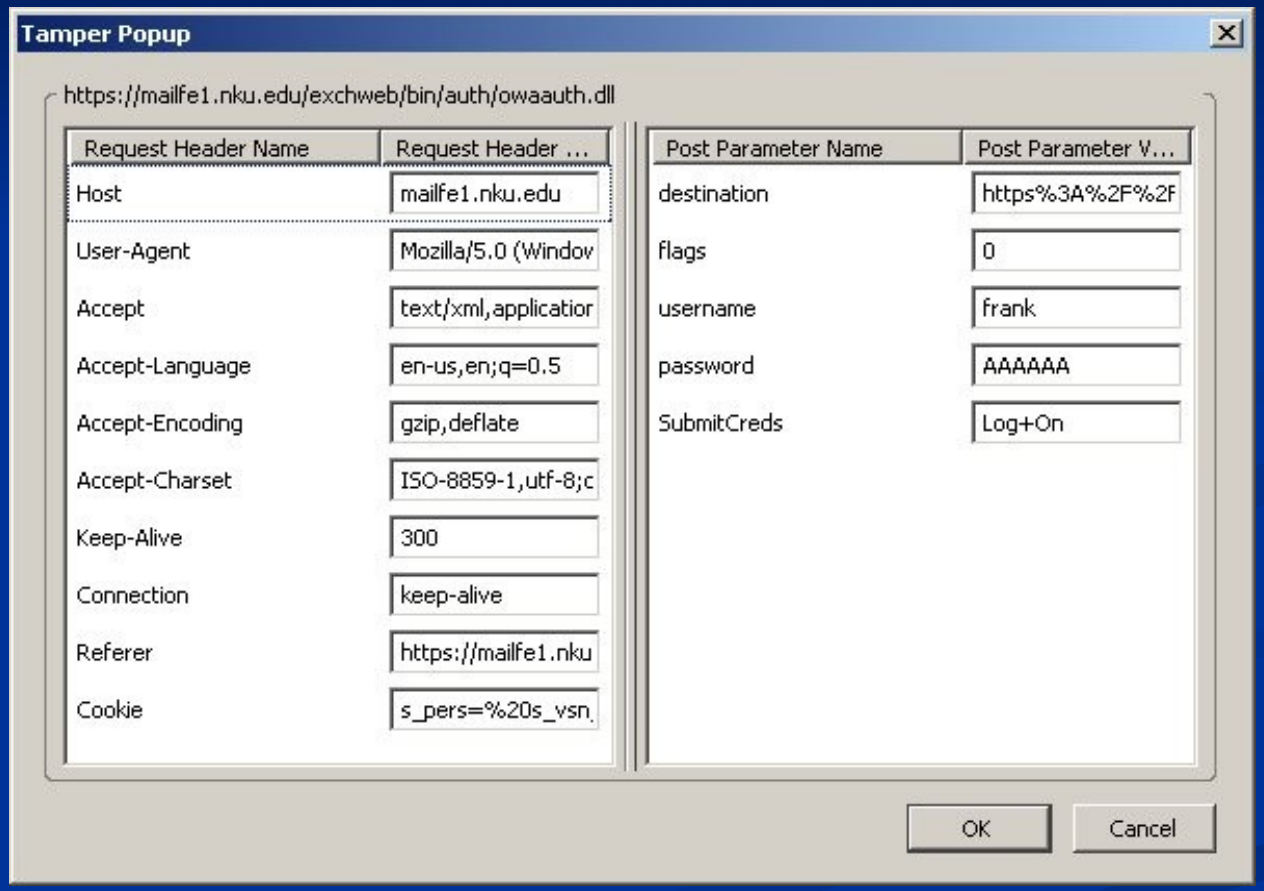
coverity

NKU NORTHERN KENTUCKY UNIVERSITY

Web Proxies


Tamper Data - Ongoing requests
Start Tamper
Stop Tamper
Clear
Options
Help
Filter
Show All
Time
Method
Status
Content Type
URL
Load Flags
12:46:32.908 ... ... -1 GET 302 text/html http://www.google.com/search?hl=en&q=&btnG=Google+Search LOAD_DOCUMENT_URI LO...
12:46:35.579 ... ... GET 200 text/html http://www.google.com/webhp?hl=en&btnG=Google+Search LOAD_DOCUMENT_URI LO...
Request Header Name
Request Header Value
Host www.google.com
User-Agent Mozilla/5.0 (Windows; U; Windows NT 5.1; en...
Accept text/xml,application/xml,application/xhtml+x...
Accept-Language en-us,en;q=0.5
Accept-Encoding gzip,deflate
Accept-Charset ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive 300
Connection keep-alive
Referer http://www.google.com/
Cookie SID=DQAAAGcAAADHzma8UwXvT_5vhSNQ8...
Response Header Name
Response Header Value
Status Found - 302
Location http://www.google.com/webhp?hl=en&btnG...
Cache-Control private
Content-Type text/html
Server GWS/2.1
Transfer-Encoding chunked
Content-Encoding gzip
Date Tue, 06 Mar 2007 17:46:32 GMT

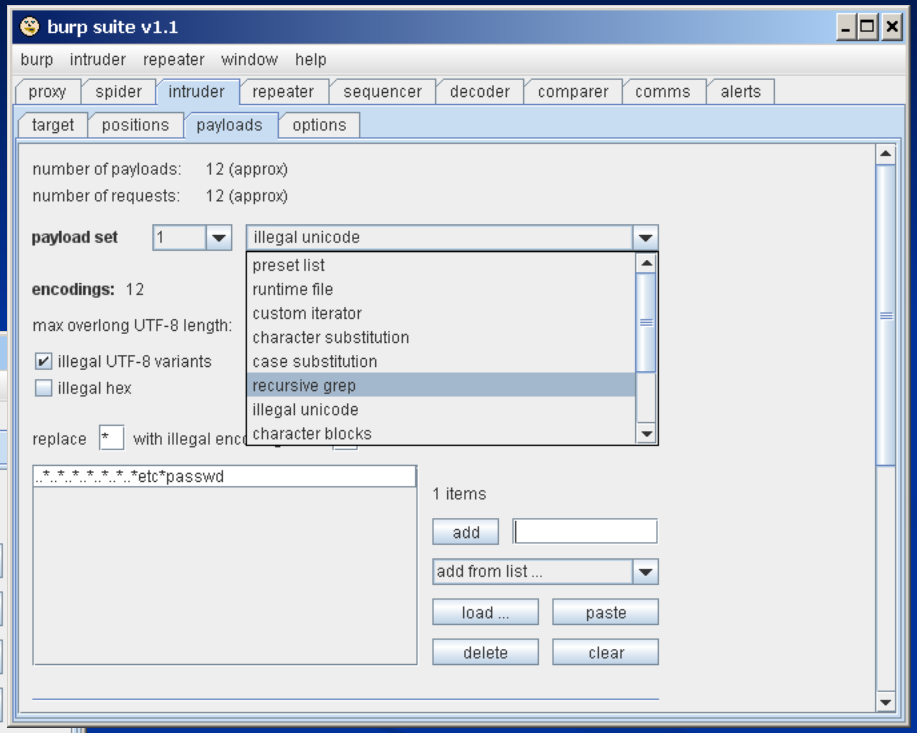
Altering Form Parameters

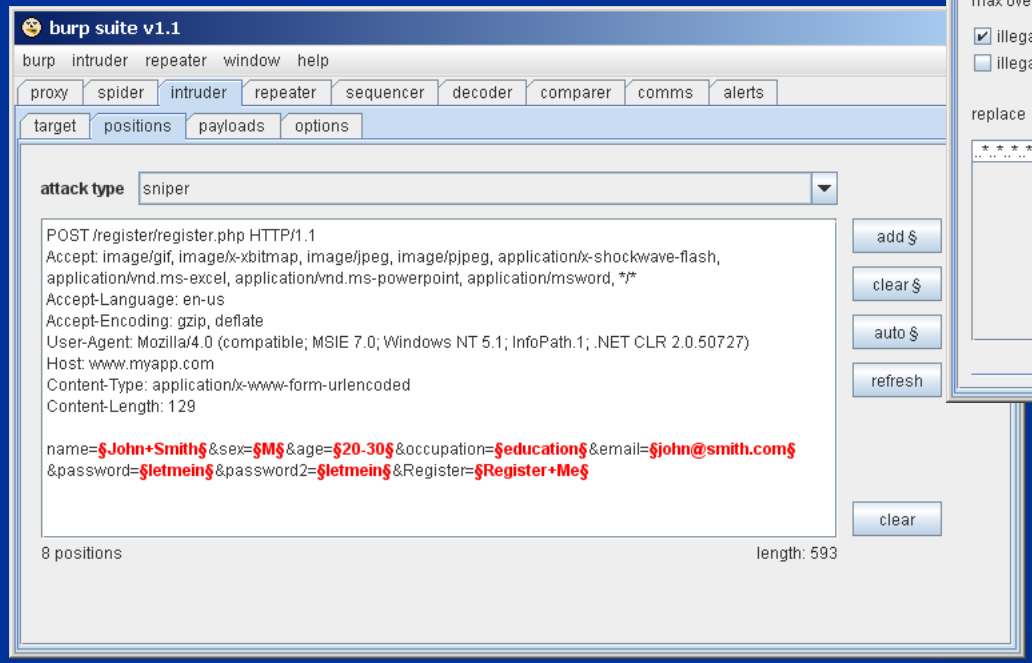


Fuzz Testing

Fuzz testing consists of

- Sending unexpected input.
- Monitoring for exceptions.





Web Application Firewalls

What is a WAF?

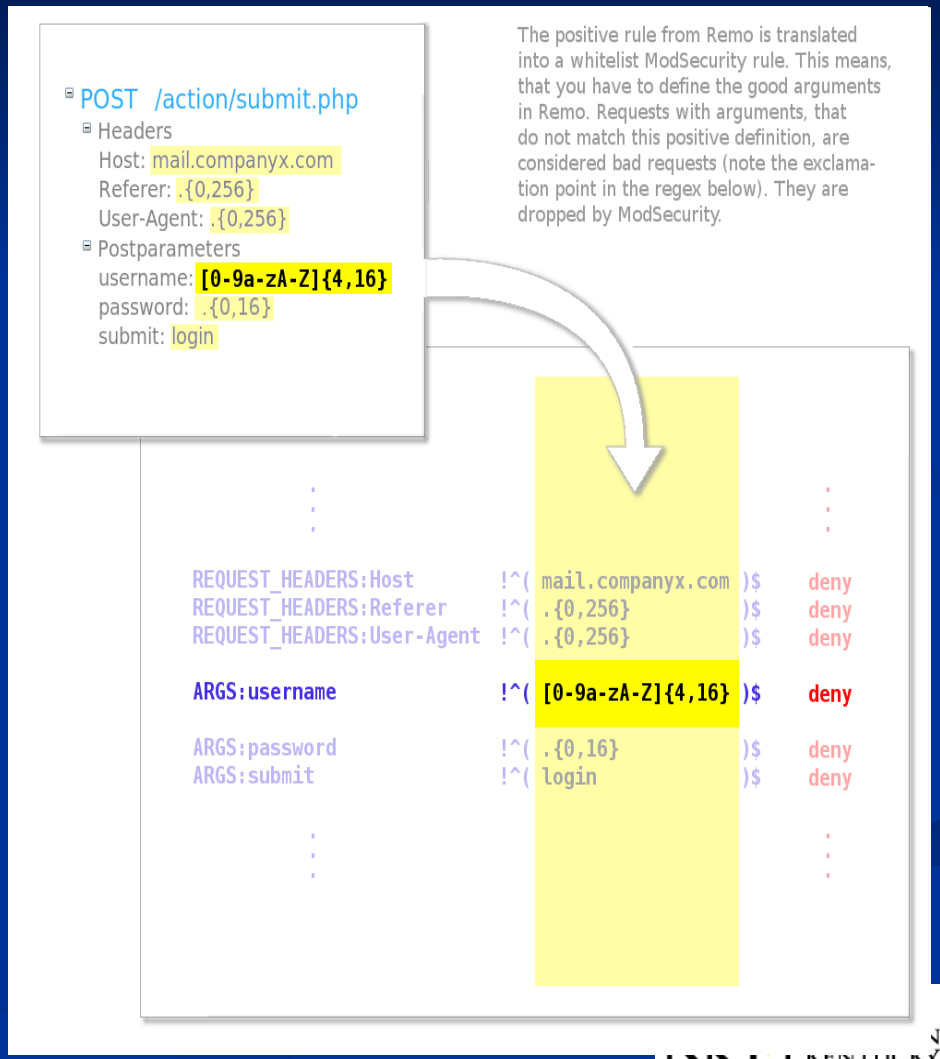
- Web monitoring.
- Access control.
- Behind SSL endpoint.

A/K/A

- Deep packet inspection.
- Web IDS/IPS.
- Web App Proxy/Shield.

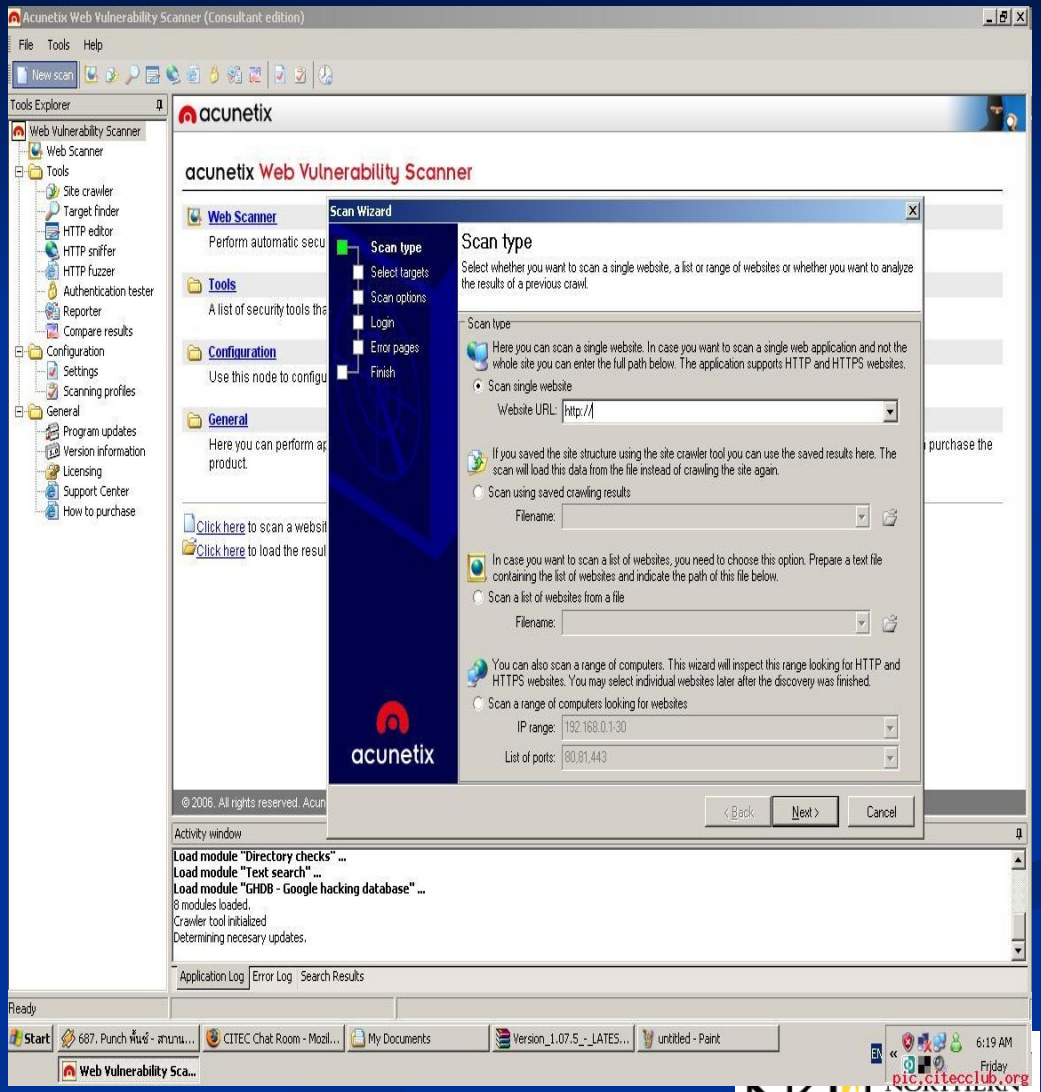
mod_security

- Open source.
- Embeds in Apache.
- Reverse proxy.



Vulnerability Scanners

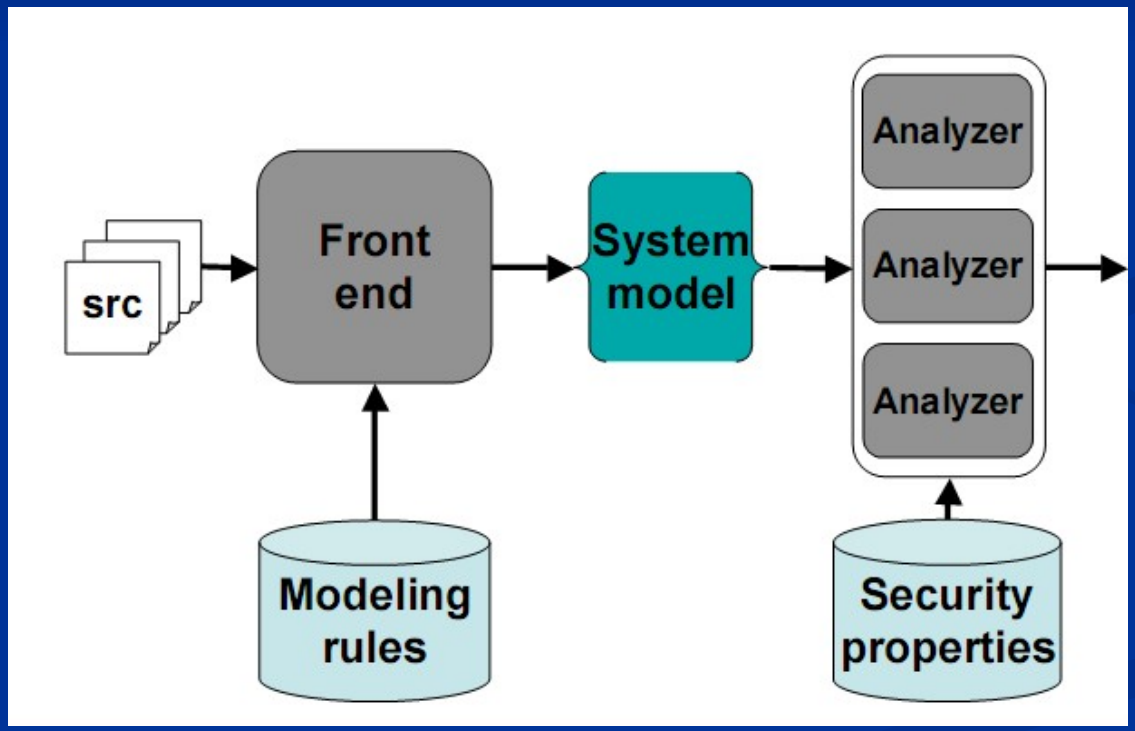
1. Spiders site.
2. Identifies inputs.
3. Sends list of malicious inputs to each input.
4. Monitors responses.



Static Analysis

Automated assistance for code auditing

- Speed: review code faster than humans can
- Accuracy: hundreds of secure coding rules



Tools

- Coverity
- FindBugs
- Fortify
- Klocwork
- Ounce Labs

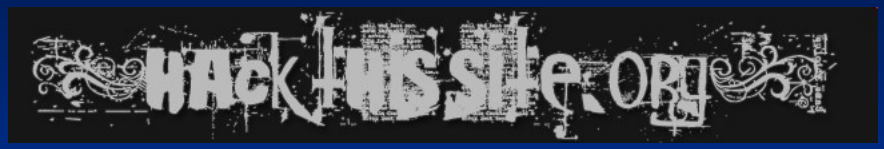
Labs

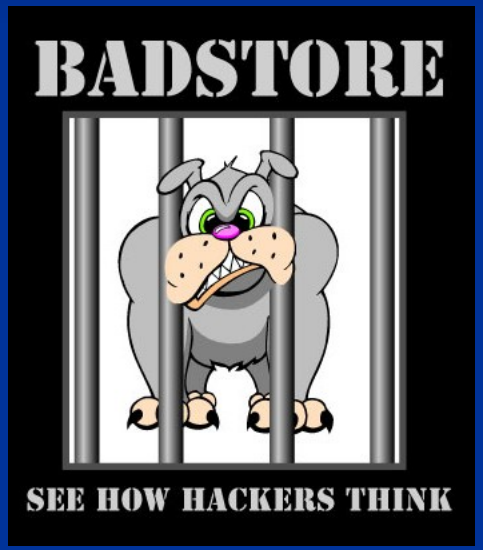
- WebGoat exercises on specific vulnerabilities.
- Using a **testing proxy** to solve more advanced WebGoat exercises.
- Assessing an application using a web **vulnerability scanner**.
- Assessing a web application using a **testing proxy**.
- Reviewing the code of an application using a **static analysis** tool.
- Deploying a **web application firewall**.
- Participating in the international CTF competition.

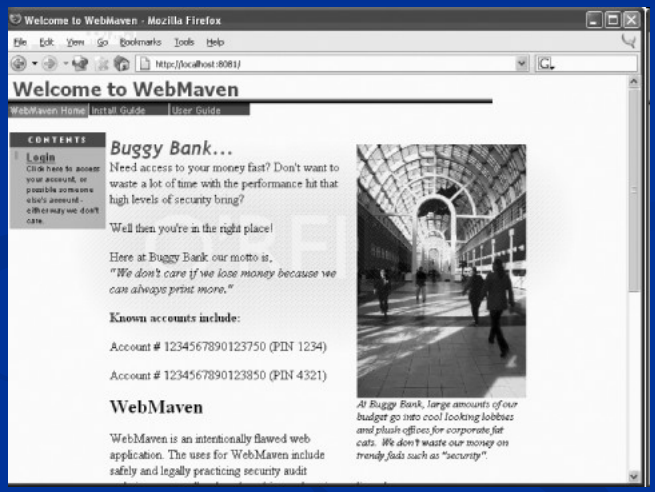
Approaches

1. Students evaluate and fix their own code.
   - Students learn about their own coding mistakes.
   - Scale of project limited to what students can write.
2. Students evaluate and fix your code.
   - Write a web application designed for teaching students.
3. Students evaluate and fix someone else's code.
   1. Use a web application designed for teaching.
   2. Analyze an open source web application with known vulnerabilities reported in NVD or other bug db.

Teaching Applications





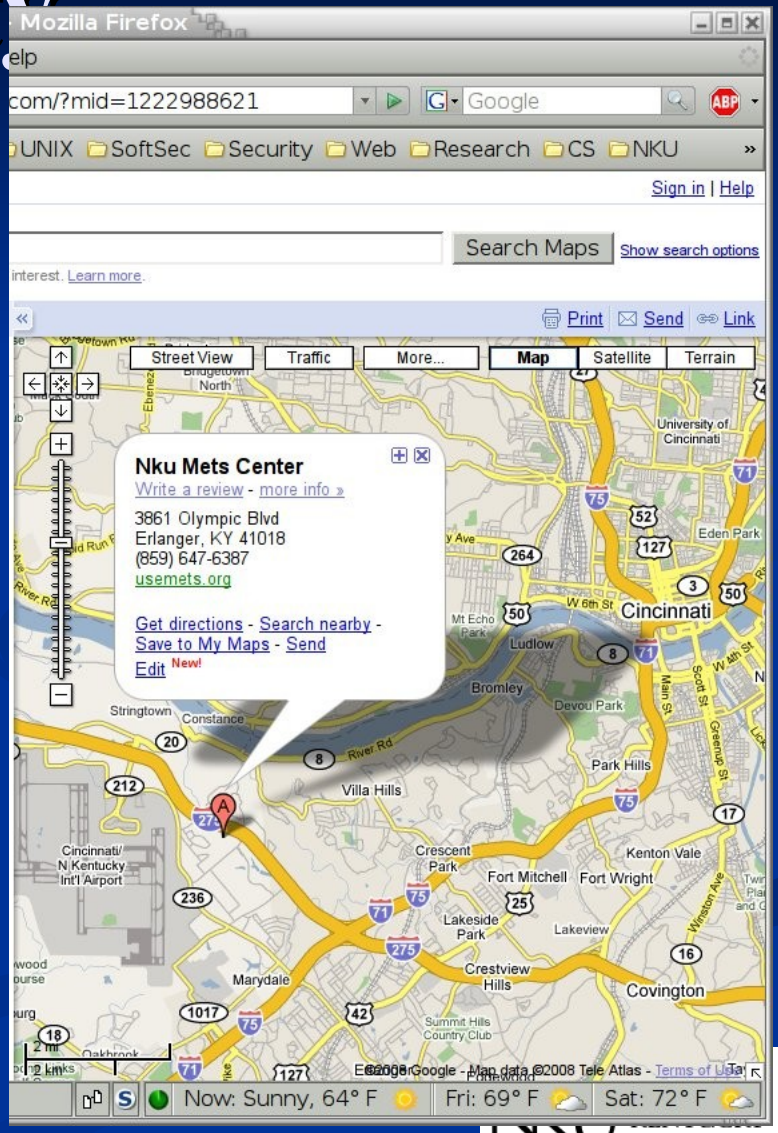


Hacme Bank, Books, Casino, Travel

Future Directions: AJAX Security

Asynchronous Javascript and XML

- Expanded server side API.
- Server API calls can be issued in any order by attacker; cannot assume calls issued in order by your client.
- Larger amount of client state.
- Client/server communication using data (XML/JSON) rather than presentation (HTML.)



Future Directions: Web Sec Class

1. Web Application Input
2. Client-side Technologies
3. Service Oriented Architectures
4. AJAX
5. Input-based Attacks
6. Injection Attacks
7. Race Conditions
8. Cross-site Attacks
9. Authentication
10. Secure Programming
11. Operational Security

Conclusions

1. Defense is shifting from network to application layer. →

   Firewalls, anti-virus, SSL input validation, WAF
2. Students need to learn to identify vulnerabilities.
   - Static analysis of source code.
   - Web proxies and scanners for testing.
3. Students need to learn to remediate vulnerabiliites.
   1. Web application firewalls for immediate short-term fixes.
   2. Repairing source code for long term fixes.