

Web-Based Vulnerabilities

CSH5 Chapter 21

“Web-Based Vulnerabilities”

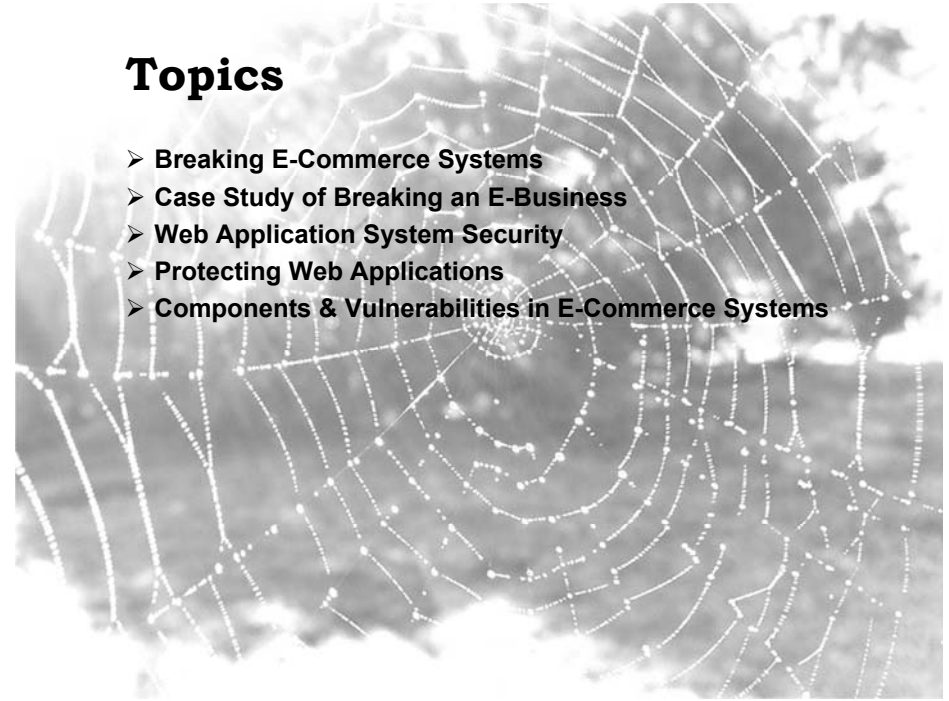
Anup K. Ghosh, Kurt Baumgarten,
Jennifer Hadley & Steven Lovaas

1

Copyright © 2011 M. E. Kabay. All rights reserved.

Topics

- Breaking E-Commerce Systems
- Case Study of Breaking an E-Business
- Web Application System Security
- Protecting Web Applications
- Components & Vulnerabilities in E-Commerce Systems



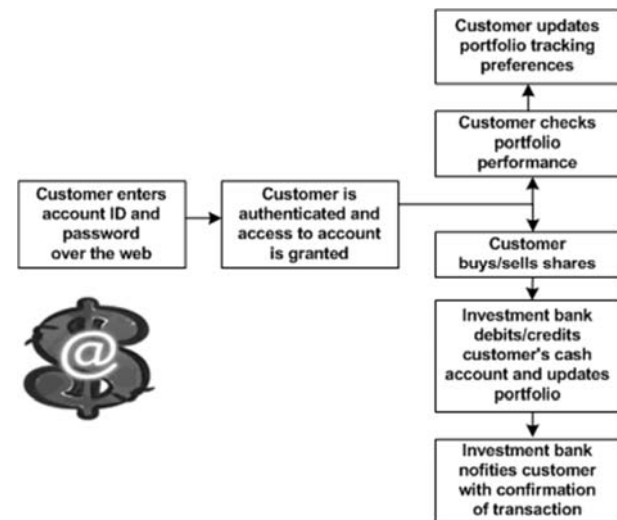
Breaking E-Commerce Systems

- Thinking about how criminal hackers think
 - ❑ Attack weakest link
 - ❑ Look for monetary gain
 - ❑ Low-hanging fruit
 - ❑ Attack servers when possible
- Must harden not only perimeter but also core
- Asymmetric attacks
 - ❑ Defense harder & more costly than offense
 - ❑ Script kiddies have caused \$M damage
 - ✓ E.g., MafiaBoy 2000 vs eBay, Amazon, Schwab....

3

Copyright © 2011 M. E. Kabay. All rights reserved.

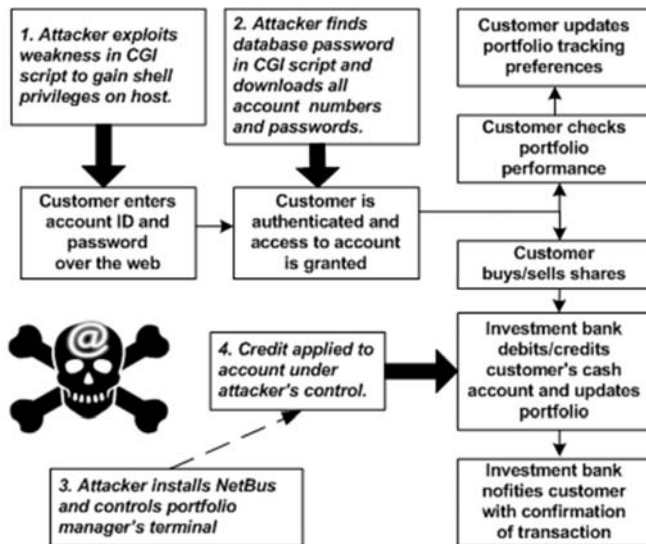
Case Study of Breaking an E-Business (1)



4

Copyright © 2011 M. E. Kabay. All rights reserved.

Case Study of Breaking an E-Business (2)



Copyright © 2011 M. E. Kabay. All rights reserved.

5

Web Application System Security

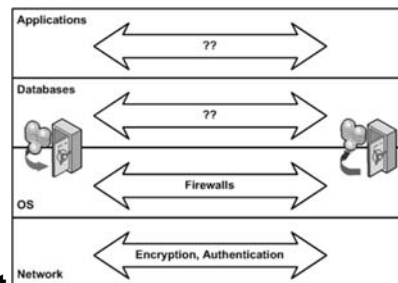
- Absolutely require corporate security policy
 - ❑ Informs decisions on specific security configurations
 - ❑ Inconsistencies can doom security
- Security systems should be independently evaluated
 - ❑ System audits (do measures conform with policy?)
 - ❑ Vulnerability analysis (can we locate obvious gaps in security?)
 - ❑ Penetration testing (can we break through the barriers using criminal hacker methods?)

Copyright © 2011 M. E. Kabay. All rights reserved.

6

Protecting Web Applications

- Layered view of systems
- Network, OS flaws usually documented
 - ❑ Alerts
 - ❑ National Vulnerability Database <http://nvd.nist.gov/>
- Vulnerability scanners available (see CSH5 Ch 46)
- Firewalls critical element
- Application servers (Java etc) must be secured
- Application security = function of how programs are configured & used (not just of patches)



Copyright © 2011 M. E. Kabay. All rights reserved.

7

Components & Vulnerabilities in E-Commerce Systems

- Client-Side Risks
- Network Protocol Risks
- Business Application Logic
- CGI Script Vulnerabilities
- Application Subversion
- Web Server Exploits
- Database Security
- Platform Security



Copyright © 2011 M. E. Kabay. All rights reserved.

8

Client-Side Risks

- Most e-commerce uses browsers
 - ❑ Also extending to hand-held devices
- Threats from malicious mobile code (*CSH5* Ch 16 & 17); e.g., Web scripts, Java applets, ActiveX controls, Trojan horse programs
- Serious risk from loss of privacy
 - ❑ Identity theft against *data subjects*
 - ❑ Business & legal consequences for corporate victims
 - ❑ Browsers typically convey much private info
 - ❑ Spyware tracks computer usage

Network Protocol Risks

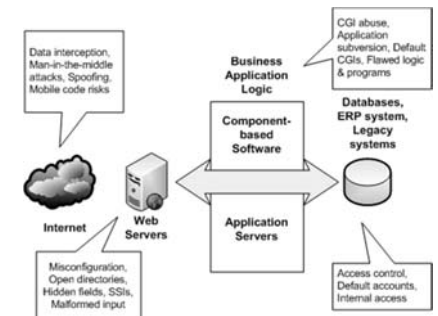
- Primarily result from sending unencrypted data over the 'Net
- Several protocols preserve confidentiality by using encryption
 - ❑ SET (Secure Electronic Transaction)
 - ❑ SSL (Secure Sockets Layer)
 - ❑ S/HTTP (Secure HTTP)(superseded)
 - ❑ S/MIME (Secure Multipurpose Internet Mail Extensions)
 - ❑ CyberCash (proprietary credit-card system)(bankrupt 2001, bought by VeriSign & First Data Merchant Services Corp.)
- See *CSH5* Ch 30

Network Protocol Attacks

- Man-in-the-middle (intercepting, inserting)
- DNS attacks (altering tables to misdirect users)
- War dialing (scanning all phone numbers in block for modems)
- Exploiting software holes (FTP, Bind, SMTP, HTTP)
- Internal access (unauthorized behavior by authorized personnel)
- Leveraging trusted hosts (attack from linked system)
- Brute-force decryption (test all possible keys)

Business Application Logic

- Key area of vulnerability
 - ❑ Usually custom SW
 - ❑ Complex
 - ❑ May not be tested as thoroughly as COTS
- Critical elements include
 - ❑ Common Gateway Interface (CGI)
 - ❑ Hypertext Processor (PHP)
 - ❑ Component-based software (CBS)
 - ✓ Enterprise JavaBeans (EJB)
 - ✓ Java 2 Enterprise Edition (J2EE)
 - ✓ Common Object Request Broker Architecture (CORBA)
 - ✓ Common Object Model (COM & DCOM)



CGI Script Vulnerabilities

- Frequent object of attack
- Inputs not under control of programmer
- Misconfiguration common problem
 - ❑ Individuals can add CGI to Web pages
 - ❑ Can go out of control – introduce holes
 - ❑ Best to limit execution of CGI to central directory under control of admin
- Protect cgi-script directories (*cgi-bin*)
- Languages create weaknesses
 - ❑ Perl, JavaScript, Python
 - ❑ Don't include Perl interpreter in *cgi-bin*
 - ✓ Could allow unauthorized execution of commands

Application Subversion

- Program misuse
- Exploit program logic
 - ❑ Raise user privileges
 - ❑ Gain unauthorized data access
- Attacker may discover unauthorized ways of using system
- Send malformed input including commands
- Redirect program output
- Beware of amateurs
- Apply strict software quality assurance to production code

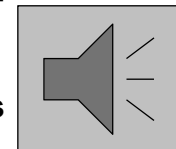
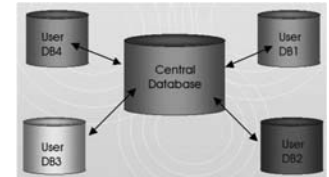
Web Server Exploits

- Configuration
 - ❑ Default = max function, min security
- HTML Coding & Server-Side Includes
 - ❑ Disallow SSI to prevent insertion of unauthorized commands
- Private Documents in Public Directories
 - ❑ Disallow *directory browsing*
- Cookies & Other Client-Side Risks
 - ❑ Users can alter cookies created by Web site
 - ❑ Cookie poisoning can exploit authentication tokens
 - ❑ E.g., alteration of discount codes → losses



Database Security

- Web interfaces too often added to formerly closed systems without proper analysis
- Most users do not encrypt their databases
- Buffer-overflow attacks can grant root access to intruder
- Some programmers *hard-code* passwords into programs (!! NO NO NO!
- Default DB settings often weak
- Audit DB log files for anomalies



Platform Security

- Operating system security essential
- See *CSH5* Ch 24
- Must not count solely on perimeter security
 - ❑ Harden OS configuration to resist attack even if perimeter is breached
 - ❑ Maintain up-to-date patches (see *CSH5* Ch 40)
 - ❑ Vulnerability assessments
 - ❑ Penetration testing



APWG Web Vulnerabilities Survey (2011)

- Anti-Phishing Working Group (APWG)

❑ <http://apwg.org/>

- APWG Web Vulnerabilities Survey:

❑ Principal Investigator and Correspondent Author:

❑ Dave Piscitello

❑ dave.piscitello@icann.org

❑ Contributing Researchers:

❑ John LaCour, Russ McRee, Robert W. Capps II, Rod Rasmussen,

❑ Ebrima Ceesay, Thomas J. Holt and Gary Warner

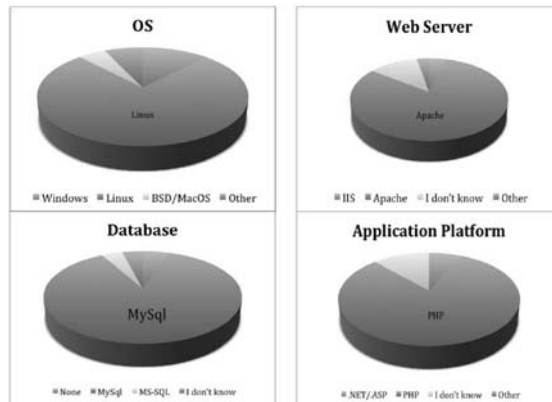
❑ *Published June 3, 2011*



http://www.antiphishing.org/reports/apwg_web_vulnerabilities_survey_june_2011.pdf

APWG Web Vulnerabilities Survey (2011)

- What hosting environments attract attackers?



http://www.antiphishing.org/reports/apwg_web_vulnerabilities_survey_june_2011.pdf

APWG Web Vulnerabilities Survey (2011)

- What are the attackers after?

➤ 7%: e-merchant

➤ 17%: customer data

➤ 4%: theft of customer data

➤ 74%: 1st attack resulting in phishing/spoof site (suggests goal is future use)

➤ 84%: uploaded phishing/spoof pages/scripts

➤ 24%: installed malicious software

http://www.antiphishing.org/reports/apwg_web_vulnerabilities_survey_june_2011.pdf

APWG Web Vulnerabilities Survey (2011)



Discovery, Response, & Remediation

- 52%: phishing detection company reports attack
- 18%: Web hosting reports attack
- 18%: Company being phished reports attack
- 8%: Called law enforcement
- 40%: attacks discovered in <1 day
- 18%: attacks discovered in 2-3 days
- 25%: did not know elapsed time between attack and discovery
- 6%: discovered through review of server logs
- 16%: discovered through review of content changes
- 4%: discovered through IDS/IPS

http://www.antiphishing.org/reports/apwg_web_vulnerabilities_survey_june_2011.pdf

21

Copyright © 2011 M. E. Kabay. All rights reserved.

APWG Web Vulnerabilities Survey (2011)



➤ What actions did victims take to stop attack?

We removed phishing web pages	85%
We repaired altered web pages related to our site	33%
We changed passwords for web programs (e.g., content management system, blog, etc.)	52%
We changed passwords for access to web server (e.g., Unix accounts)	54%
Our hosting provider shut down web site entirely	14%
We shut down the web site entirely	15%
We patched or update the operating system	11%
We patched or updated the web server software (e.g., Apache, IIS)	9%
We patched or updated vulnerable software packages	21%
We had our developers fix our custom software	8%
Reviewed system and web server log files	34%
We redirected the phishing site to the APWG phishing education page	14%

http://www.antiphishing.org/reports/apwg_web_vulnerabilities_survey_june_2011.pdf

22

Copyright © 2011 M. E. Kabay. All rights reserved.



DISCUSSION

23

Copyright © 2011 M. E. Kabay. All rights reserved.