# Data Breaches and Web Servers:
# The Giant Sucking Sound

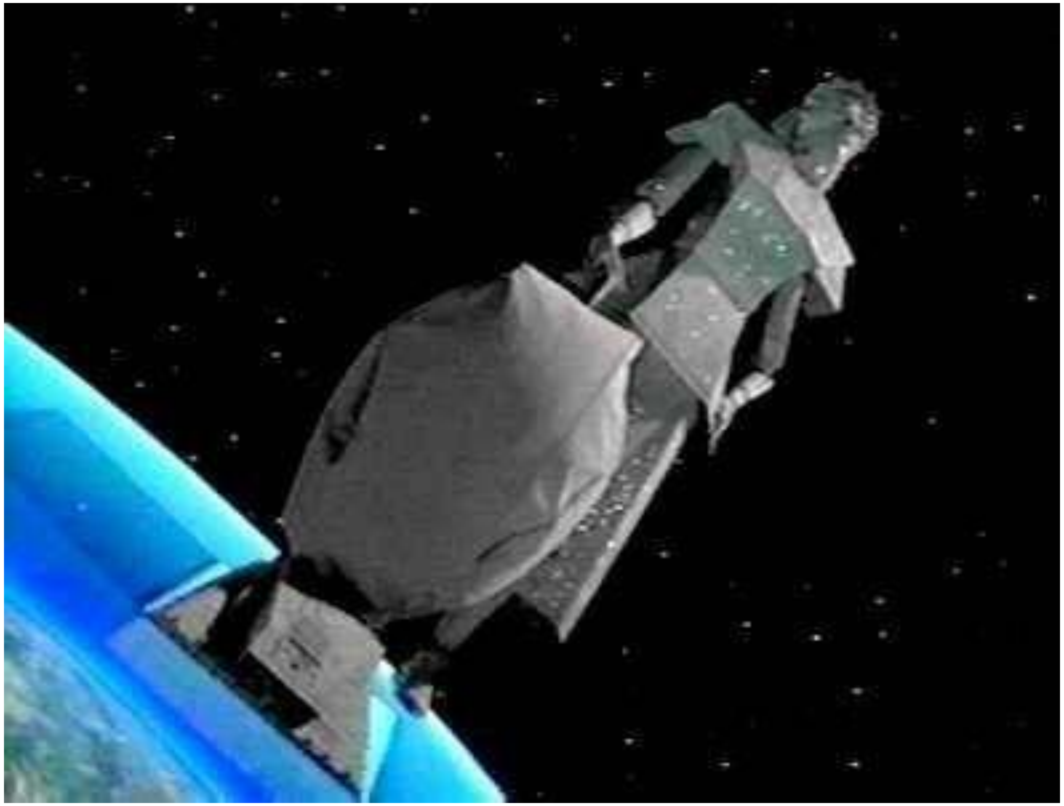
**Guy Helmer**
**CTO, Palisade Systems, Inc.**
**Lecturer, Iowa State University**
**@ghelmer**

Session ID: DAS-204

Session Classification: Intermediate

RSACONFERENCE2012

# The Giant Sucking Sound!



# Why Worry?

“You think you’re secure until you’re breached, then you find out what it costs”

- 2010 Ponemon Study: $7.2 million average breach cost
- Average cost per individual record: $214
  - Potential cost to org: # customers * $214
- Sony reported $171 million cost for breaches
  - Plus indirect costs

# Objectives

- Sense the urgency of protecting web servers
- Recognize how web server data loss happens
- Recognize what can be done to protect web servers
- Prioritize and implement protections

# Attack Vectors

Quick check: How many have seen successful web attacks recently? What sort of attacks?

# Common Attack Vectors

- Major problems:
  - SQL injection
  - Parameter tampering
- Other issues:
  - Password guessing / authentication attacks
  - Cross-site scripting
  - Cross-site request forgery
  - Insecure data storage

# SQL Injection

- Web server often ties into database
- String data in web forms passed in DB requests
- Think about the common lost password lookup web form:
    - Query code in application:
        - SELECT * FROM users WHERE username = ‘$USER’;
    - What if $USER is “joe’ OR ‘x’=‘x”? After substitution:
        - SELECT * FROM users WHERE username = ‘joe’ OR ‘x’=‘x’;
    - Returns all records in the users table!

# Parameter Tampering

- Modifying a form, query parameter, or cookie
  - Such as changing the account ID in a form to access an account other than that owned by the authenticated user
    - URL in page:

      http://www.bank.com/action.asp?account=1001&debit=1000
    - Modified URL:

      http://www.bank.com/action.asp?account=2002&debit=1000
  - Firefox Tamper Data plugin
  - Internet Explorer TamperIE extension

# Authentication Attacks

- Poor passwords – easily guessed
  - letmein
  - qwerty
  - 12345678
- Passwords stored in plaintext
- Passwords stored in unsalted hashes

# Cross-site Scripting (XSS)

- Inject malicious Javascript/Java/Flash/ActiveX
  - Such as through comment forms
    - PHP injection a personal site
  - Usually used to execute arbitrary code in a victim’s browser
  - Can occur through advertisements

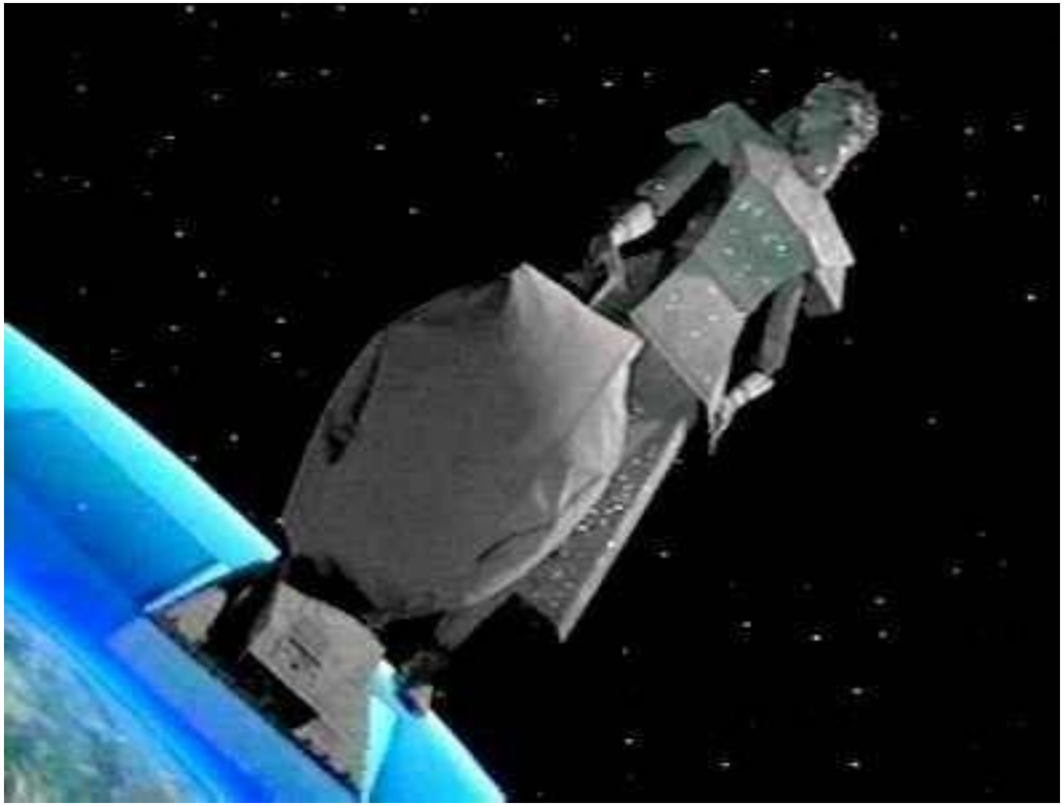
# Cross-site Request Forgery

- Malicious web request executed using identity of victim user
  - Such as IMG, IFRAME, or FORM tag with malicious URL/query parameters, e.g.
    - <img src="http://foo.com/logout">
  - Exploits victim’s browser state or form content:
    - Cookies
    - Form or query parameters
    - Cached authentication credentials

# Insecure Data Storage

- Sensitive data
  - Cryptographically obscured or
  - Appropriately encrypted
- Applies to
  - Online and
  - Offline

# Stop the Sucking!



# Securing Web Servers

Quick check: How many have active web application firewalls in place? Data loss prevention on web servers? Are they working?

# Defenses

- What helps immediately:
  - Web application firewall
  - Data loss monitoring and prevention
  - Log monitoring
  - Patching / updating
- Plus good hygiene:
  - Architecture
  - Hardening
  - Penetration testing
  - Auditing

# Web Application Firewall

- Protect web applications
  - Cross-Site Scripting
  - SQL Injection
- Can require rule customization
- Can affect web site performance

# Web Application Firewall

- Quick and simple filters
  - Joomla
    - http://docs.joomla.org/Htaccess_examples_(security)
  - Wordpress Firewall Plugin
    - http://www.seoegghead.com/software/wordpress-firewall.seo
- More involved filters
  - mod_security
    - http://www.modsecurity.org/
  - MS IIS URLScan
    - http://www.iis.net/download/urlscan
- Numerous commercial offerings

# Data Loss Monitoring / Prevention

- Data loss prevention
  - Data-at-rest
    - Monitor static content
    - Find data that should not be publicly accessible
  - Data-in-motion
    - Monitor outbound data
    - Unexpected volumes of privileged data
- Database activity monitor
  - Monitor database requests
    - Unexpected privileged access
    - Unexpected data access
    - Attack protection

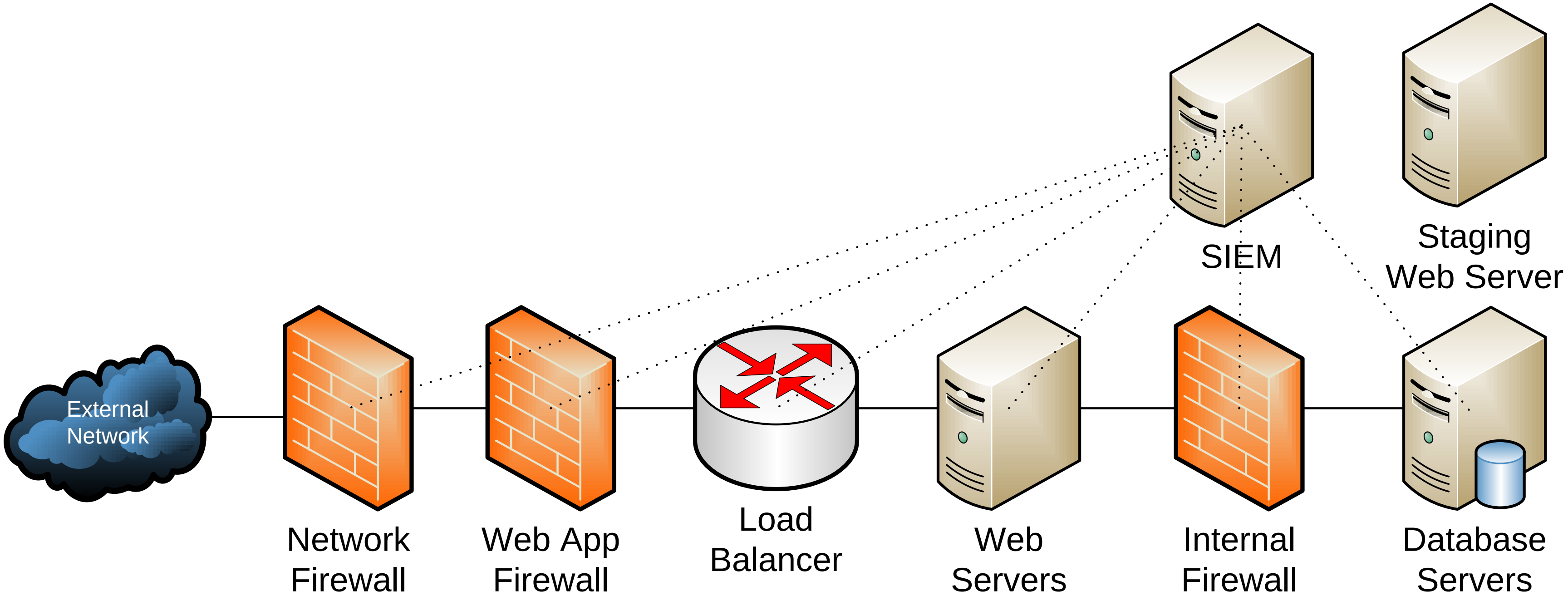
# Log Monitoring

- Trend monitoring
  - Web log analyzers
- Anomaly monitoring
  - Check web error logs
- Security integration and event monitor
  - Tie events from all components of web application
    - Firewall, web app firewall, load balancer, web server, internal firewall, database server

# Patching / Updating

- Define patch strategy
  - Policy
  - Process
- Process
  - Assess
  - Obtain
  - Test
  - Deploy
  - Validate

# Architecture



# Hardening

- Host firewall
- Management access authentication and auditing
- No unnecessary software installed
- Disable unnecessary services
- Software up-to-date
- Access controls on files, directories, and registry
- No unnecessary web server scripts or modules

# Penetration Testing

- Check
  - Common misconfigurations
  - Path traversal
  - Out-of-date software
  - Parameter and cookie tampering
  - Missing authentication checks
  - Cross-Site Scripting
  - Script injection
  - SQL injection

# Auditing

- Verify the website software
  - Server, add-on components, web apps
- Verify the website content
- Web searches to validate available content
  - Example: Outdated documents available via HTTPS
- Verify firewall functionality
- Verify admin accounts and access history
- Verify file and directory permissions

# Apply

- In the first three months following this presentation you should:
  - Evaluate and apply web application firewall and/or data loss prevention to web server transactions
  - Harden, patch, and verify web server software components
  - Implement log monitoring
- Within six months you should:
  - Determine what additional protections are warranted
  - Analyze and pen-test web apps for issues

# Summary

- Sense the urgency of protecting web servers
- How web server data loss happens
- What can be done to protect web servers
- Prioritize and implement protections

# Online Resources

- Open Web Application Security Project
  - https://www.owasp.org/
- National Institute of Standards and Technology
  - Secure Web Servers: Protecting Web Sites That Are Accessed By The Public
    - http://csrc.nist.gov/publications/nistbul/b-January-2008.pdf
  - Guide to Secure Web Services (Spec. Publ. 800-95)
    - http://csrc.nist.gov/publications/nistpubs/800-95/SP800-95.pdf

Contact: guy.helmer@palisadesystems.com

**?**