How were the Credit Card Numbers Published on the Web?

February 19, 2004

NETCONTINUUM

Agenda

- Security holes? … what holes?
- Should I worry?
- How can I asses my exposure?
- … and how can I fix that?
- Q & A
- Reference: Resources and Tools

NETCONTINUUM

But I already have "security" … don't I?

Network Security:

- Network firewall
- Network IDS
- SSL / TLS
- Network assessment

Server Security:

- Patch Warfare
- Host IDS
- Forensic log analysis
- Server assessment

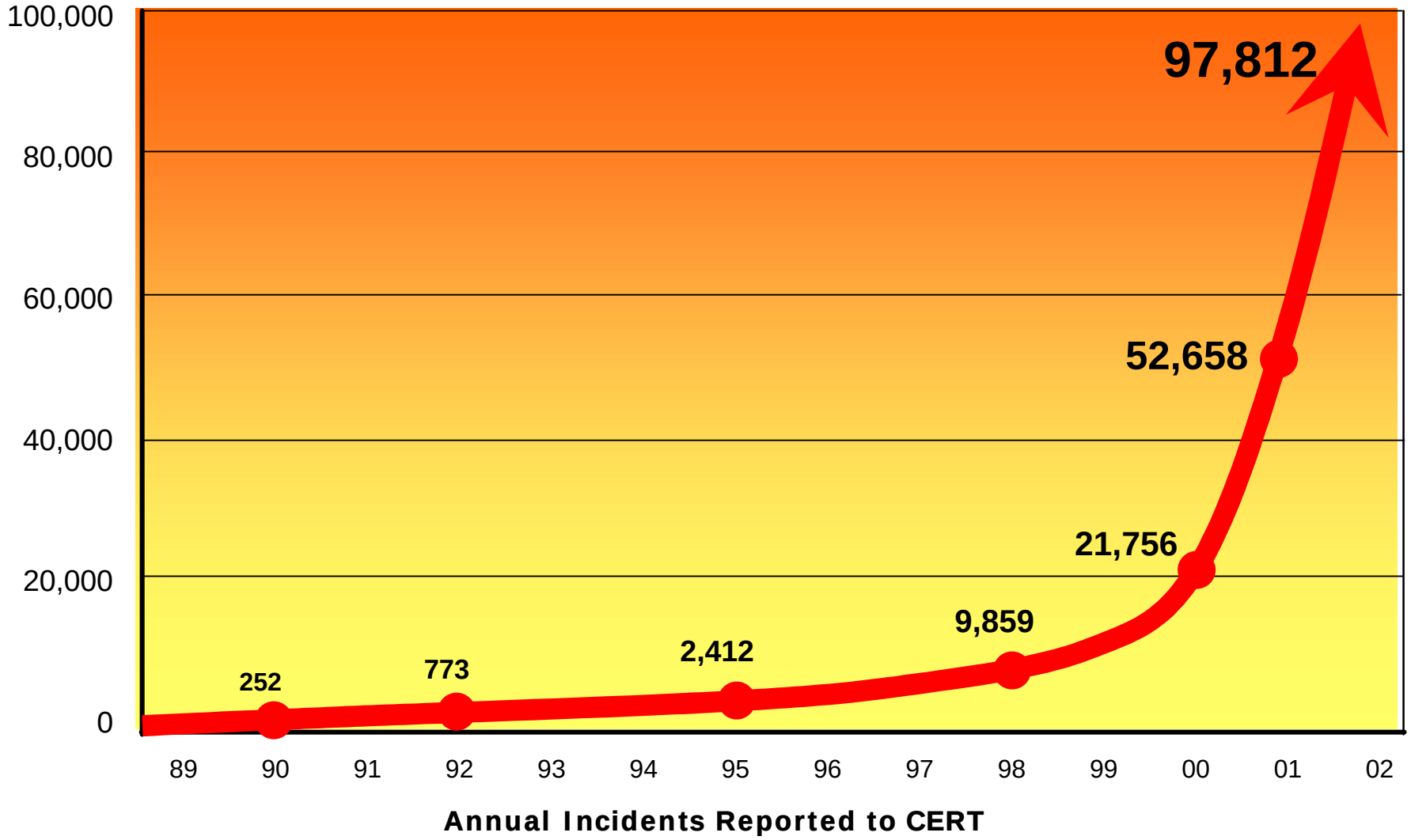
Application Security:

- AAA
- Web App Firewall
- Identity Mgmt.
- Assessments, audits

NETCONTINUUM

Rate of Security Incidents is Skyrocketing

Where is this massive increase coming from?



NETCONTINUUM

List of Application Attack Techniques Grows Every Day

Web Application Threats

1. Cross-Site Scripting
2. SQL Injection
3. Command Injection
4. Cookie/Session Poisoning
5. Parameter/Form Tampering
6. Buffer Overflow
7. Directory Traversal/Forceful Browsing
8. Cryptographic Interception
9. Cookie Snooping
10. Authentication Hijacking
11. Log Tampering
12. Error Message Interception
13. Attack Obfuscation
14. Application Platform Exploits
15. DMZ Protocol Exploits
16. Security Management Attacks
17. Zero Day Attacks
18. Network Access Attacks
19. TCP Fragmentation
20. Denial of Service
21. Distributed Denial of Service

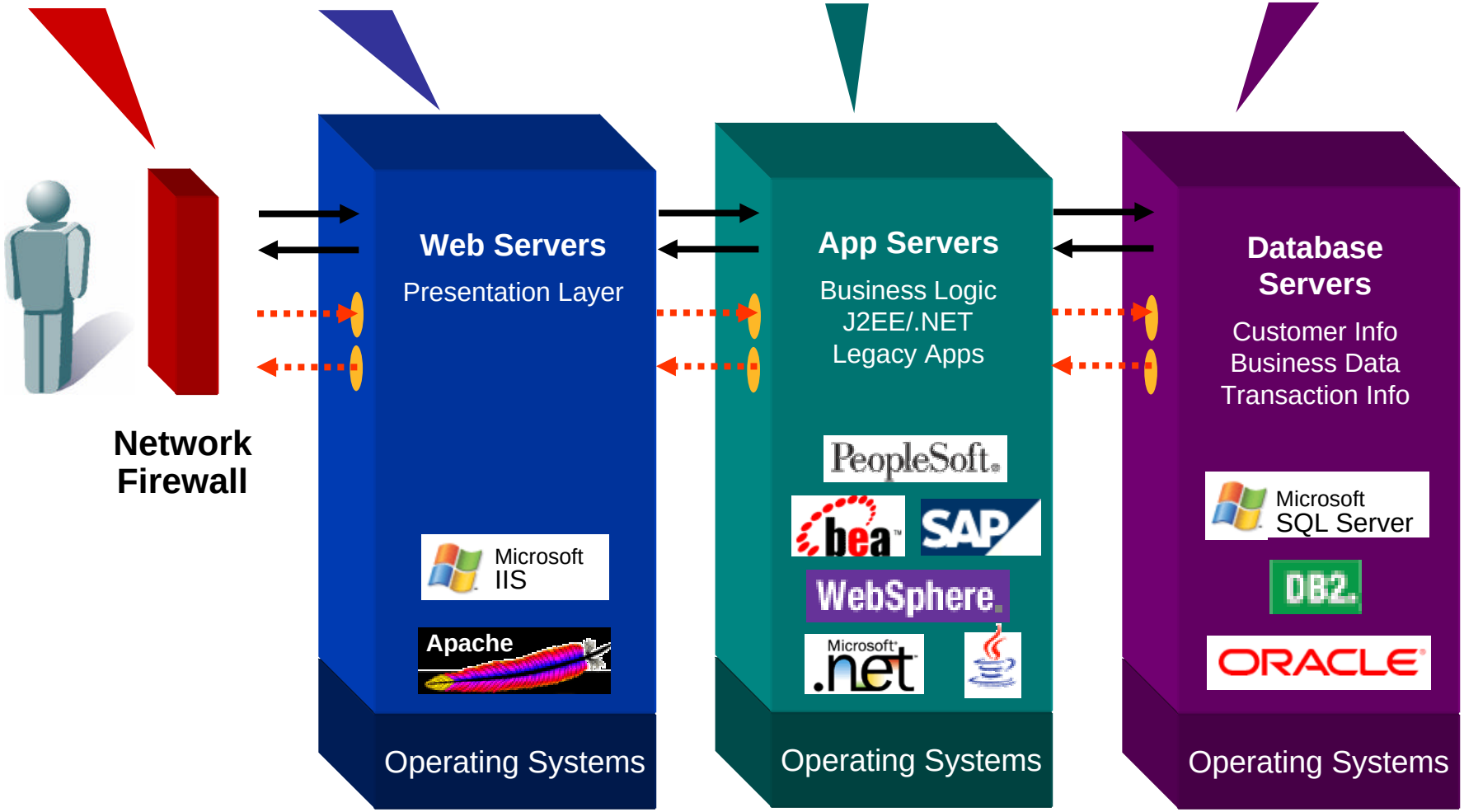
Most Common Impact

- Access to unpublished pages
- Unauthorized app access
- Password theft
- Identity theft
- Theft of customer data
- Modification of data
- Disruption of service
- Website defacement
- Recovery and cleanup

NETCONTINUUM

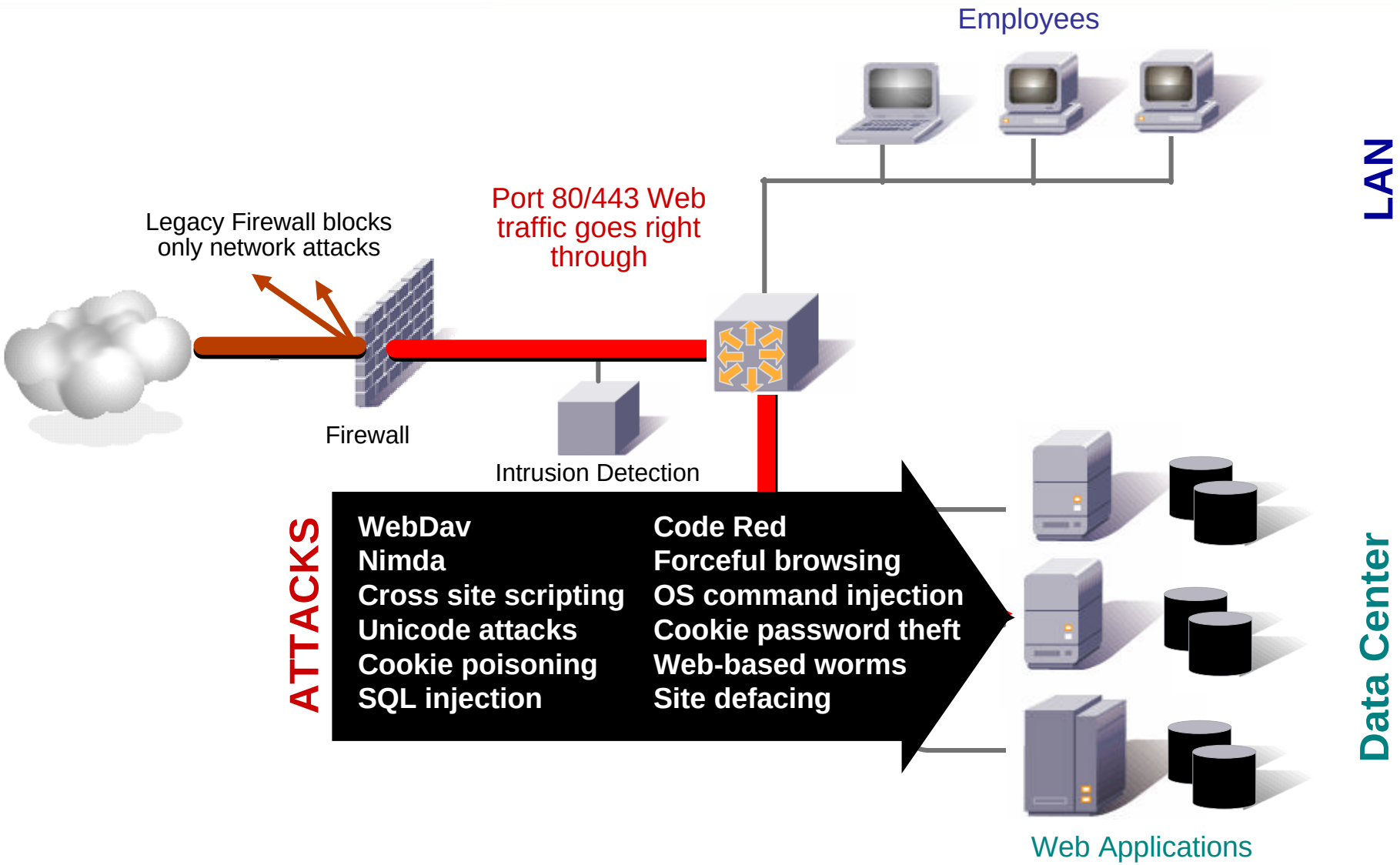
Web Apps are unbelievably complex

http://www.none.to/script?submenu=update&uid=1'+or+like'%25admin%25';--%00



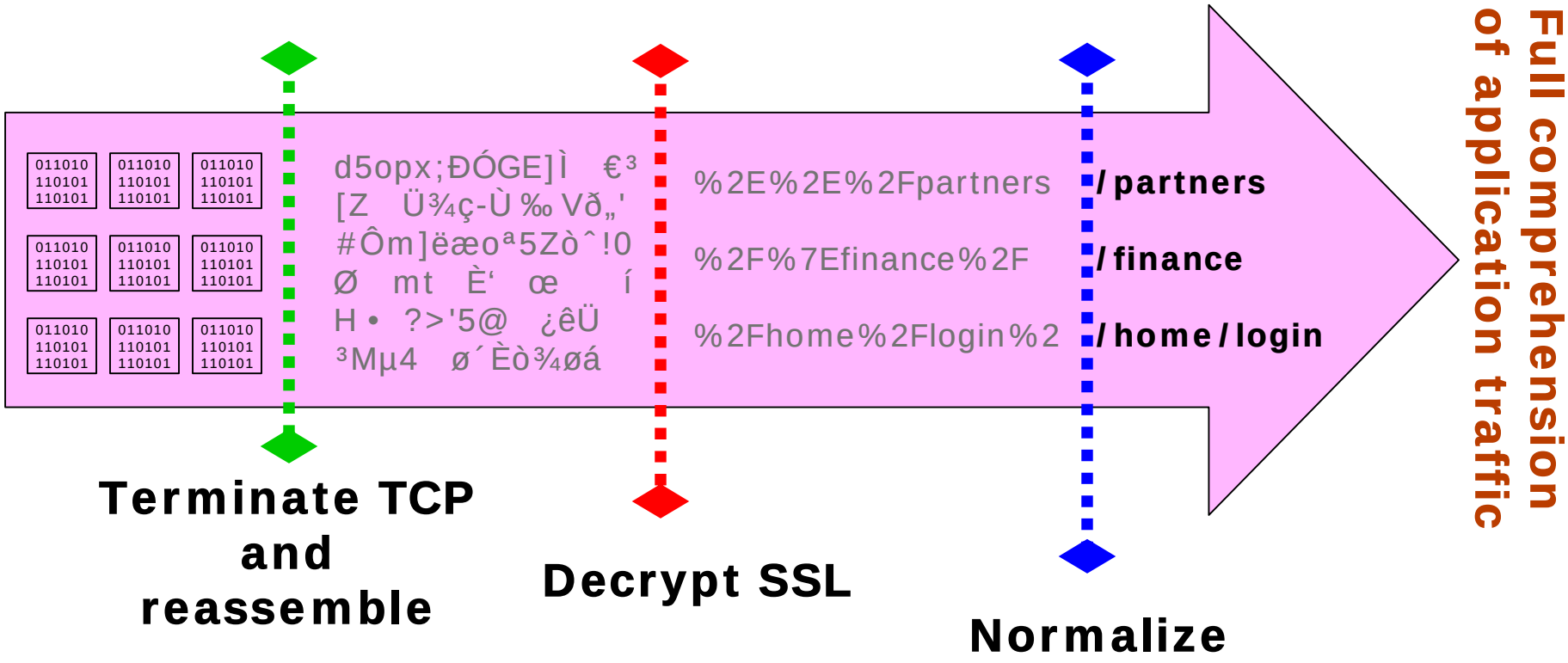
NETCONTINUUM

Web Attacks are Invisible to Firewall and IDS



NETCONTINUUM

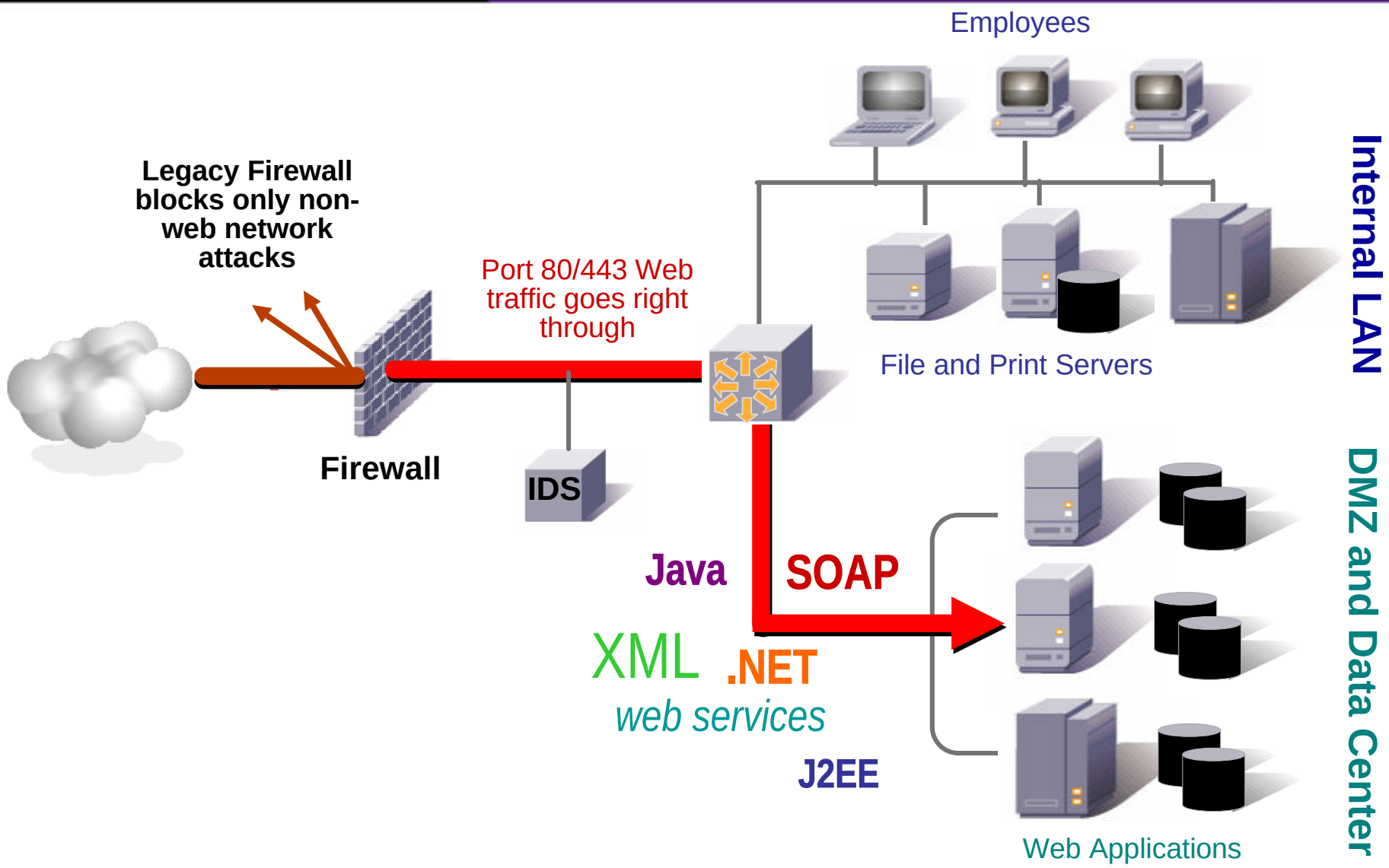
That's why!



- Normal site behavior is often hard to characterize
- There's no effective way to control browsers or apply policies to Internet users
- Web applications often run the business, can't easily be simulated, and can never be fully tested off-line

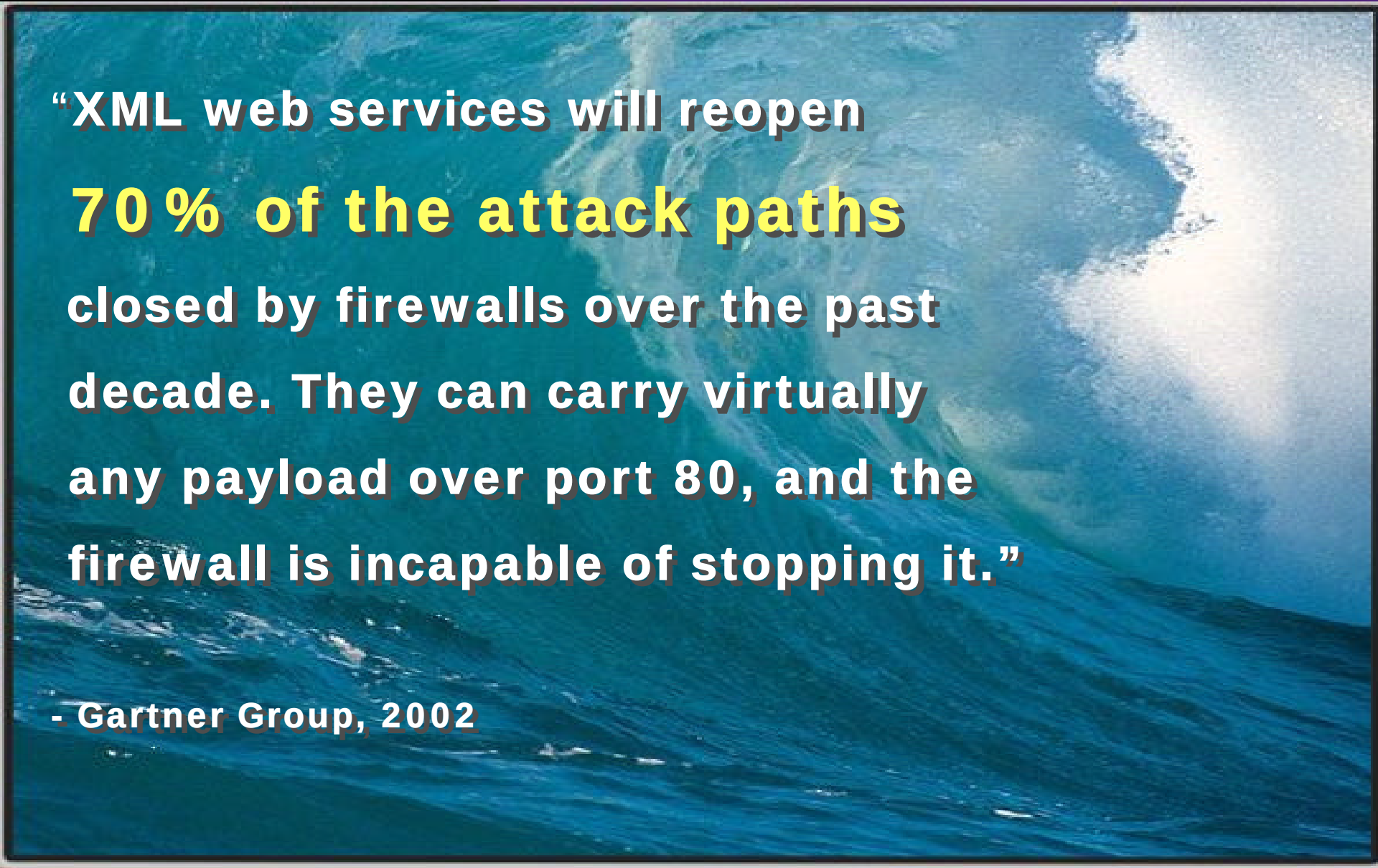
NETCONTINUUM

It's going to get worse



NETCONTINUUM

A lot worse…



"XML web services will reopen

70% of the attack paths

closed by firewalls over the past decade. They can carry virtually any payload over port 80, and the firewall is incapable of stopping it."

- Gartner Group, 2002

Detecting vulnerabilities in web applications

- Tools:
  - Learn what assessment tools are available, and test them
  - Use automated tools whenever possible
  - If an automated tool is not available, write or script one
  - Test the security of the network, servers, OS, webservers, middleware, business logic, databases, and browsers
- Techniques:
  - Think like an Attacker!!! Where do you want to go today?
  - Use de-compilation techniques to review source code
  - Be curious – try “strange” techniques and “fuzzing”
    - What can an unauthenticated user do?
    - What can an authenticated user do?
  - Document everything you do (and what you didn’t do)!
  - Become familiar with security bulletins

Get written permission from someone authorized to give it to you

Know thy Application!

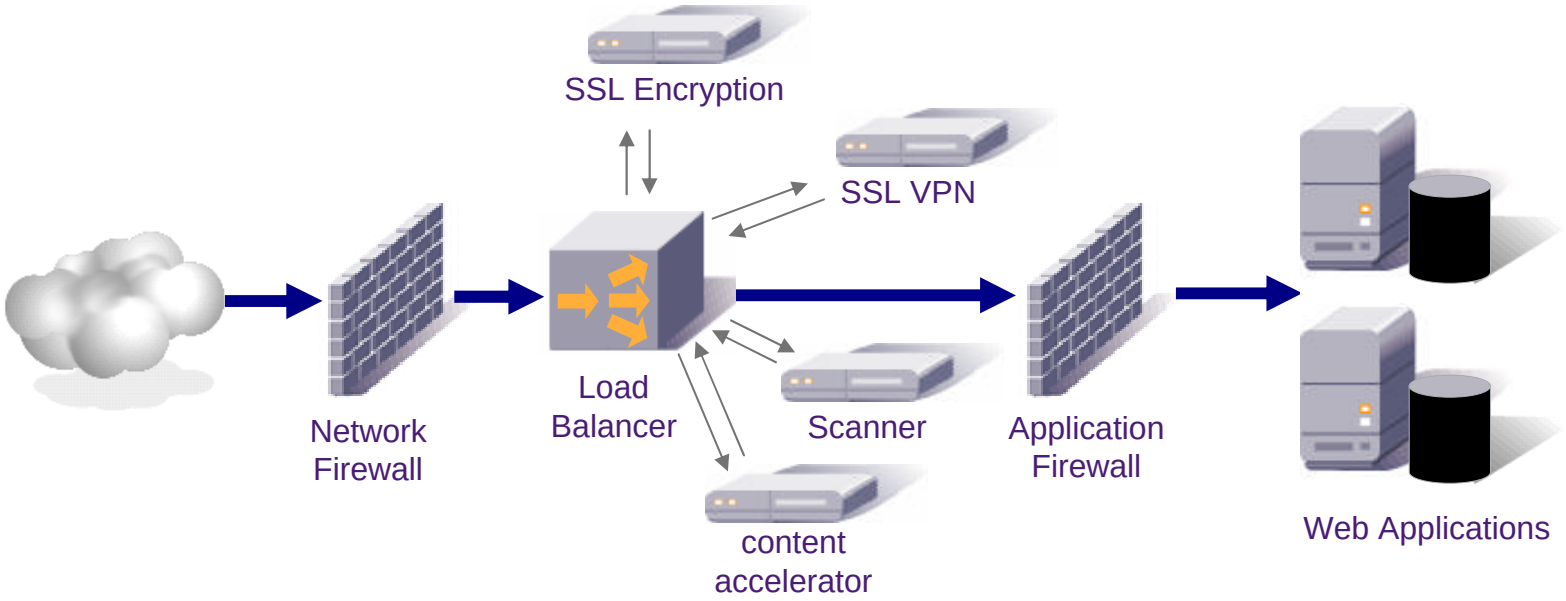
- Security alerts' bulletins – reading essential
- Abundance of open-source and commercial scanners:
  - nmap – primary lower-layers
  - nikto, nessus, brutus, spike
- Application vulnerability and assessment scanners perform automated application testing based on a database of known vulnerabilities
- Some (spike) are capable of brute-force testing and thus able to detect previously unknown problems
- Forensics and Intrusion Detection Systems

Protection Methods

- Firewalls – traditional network layer systems are usually inadequate to fully protect web applications
- Web Security Gateways – purpose-built systems to protect web applications through deep traffic inspection
- SSL – cryptographic protocol that provides privacy and authentication
- Restrictive access authorization policies
- Server and application patching

NETCONTINUUM

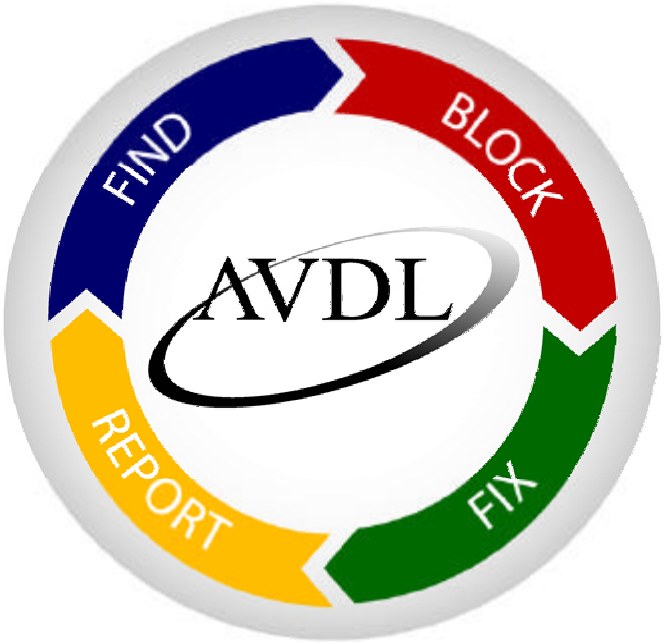
Secure Ecosystem



- Effective web app installation and deployment requires a set of techniques and systems that implement them
  - security
  - traffic management
  - monitoring

What is AVDL

Application Vulnerability Description Language (AVDL)



www.avdl.org

- Developed through the OASIS standards body
- XML-based standard proposed in April 2003
- Draft approved and now in public commenting period
- Multi-vendor effort in the application security led by NetContinuum, SPI Dynamics, and Citadel

Introduction to AVDL

- AVDL security data consists of “probes” representing application transactions
- “vulnerability” probes specify known defects, applicability domain, and detection signature as well as human-readable descriptions, tracking, etc…
- “traversal” probes specify normal, legitimate application usage and can specify parameters, attributes, valid ranges, etc…
- Probes can be batched together in “sessions” or used individually, either off-line or real-time

The AVDL Vision

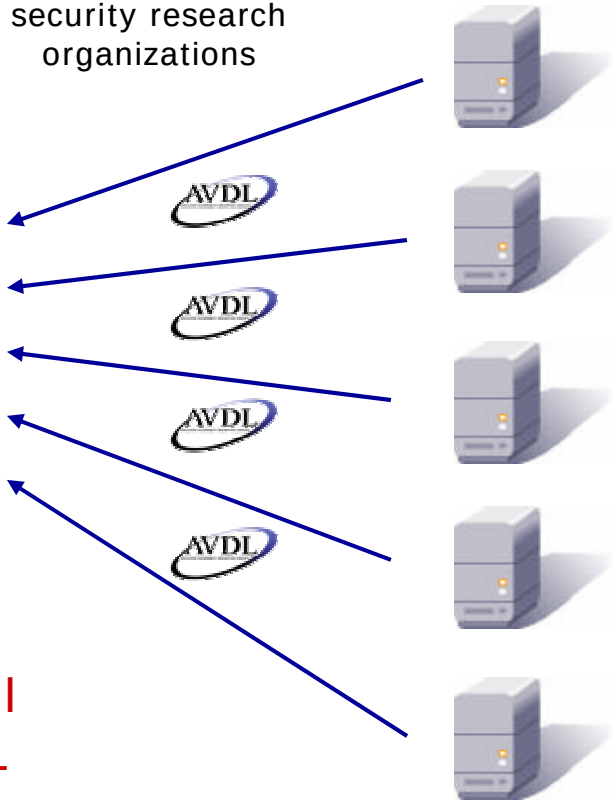
Security Alerts
CIAC subscribes to security alerts from vendors and security research organizations

DOE Sites
Custom subscription policy
Integrates with application firewalls and patching systems

System Admins
Custom subscription policy
Integrates with VA tools

CIAC Portal

US Government Security Incident Response Portal will debut this Spring with AVDL "listeners"



NETCONTINUUM

A comparison of mitigation strategies

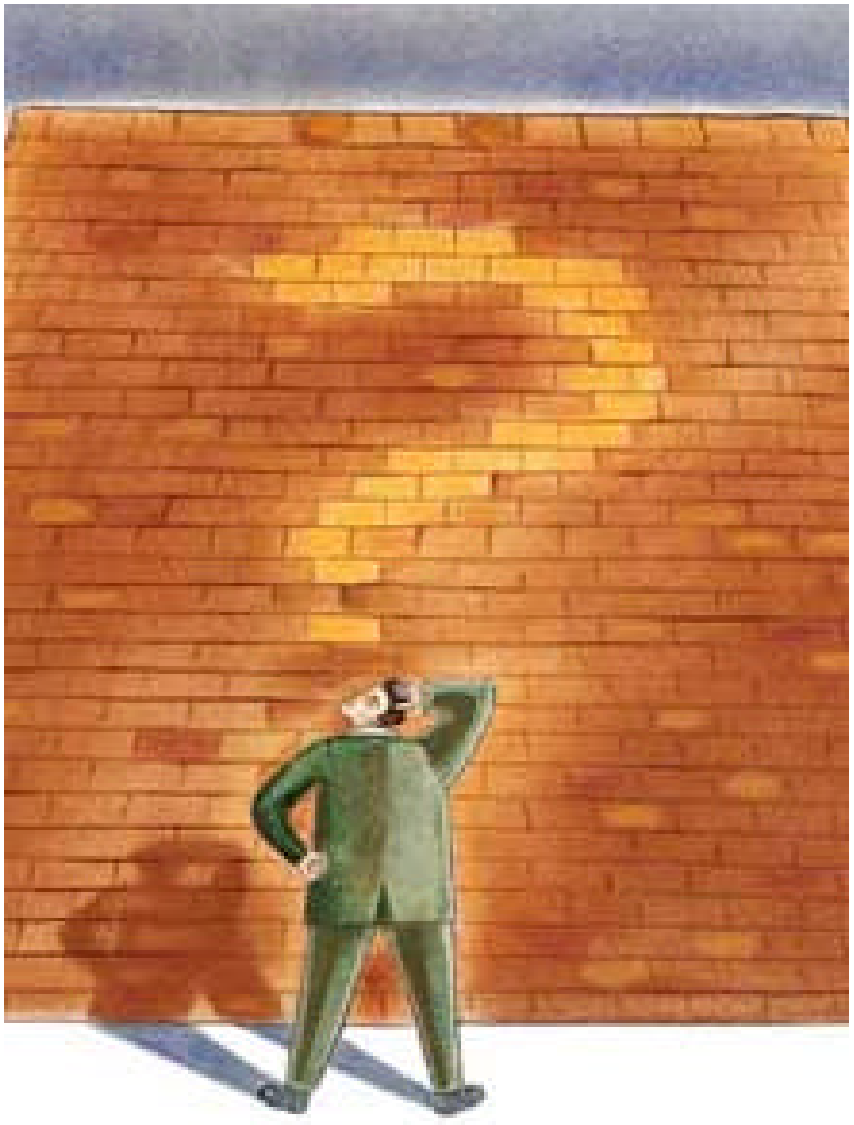
known network and application vulnerabilities	detect and disclose	vulnerability assessment tools, forensic log analysis, panic calls to Help Desk!
	mitigate	network firewall, application firewalls, patches and upgrades
unknown network and application vulnerabilities	detect and disclose	source code review, fuzzing, brute force attacks, penetration testing techniques
	mitigate	Personal firewalls, Host IDS, Web Application Firewalls

Action plan in 7 easy steps

To mitigate web application vulnerabilities:

1. Know the risk your organization is willing to accept, and clearly define “acceptable loss”
2. Implement a “Defense in Depth” protection architecture
3. Develop a deep understanding of the usage and features of your most critical web applications
4. Regularly test all layers of your web applications with automated and manual tools and techniques
5. Perform periodic forensic review of logs and error messages
6. Trust nobody – validate all user input
7. Think Like an Attacker!!! *(or befriend someone who does)*

Q & A



Thanks!

Jan Bialkowski
VP & CTO

NetContinuum
408.961.5603
jan@netcontinuum.com

NETCONTINUUM

Reference: Resources and Tools

Simulation and Training Environments	WebMaven – http://www.mavensecurity.com WebGoat – http://www.owasp.org/development/webgoat
Reading Materials	Open Source Security Testing Methodology Manual – http://www.isecom.org/projects/osstmm.htm OWASP Guide to Building Secure Web Apps http://www.owasp.org/documentation "Hacking Exposed" Series – http://www.hackingexposed.com "Web Hacking Attacks and Defense" – McClure, Shah
Assessment Tools	Nmap, Nikto, Nessus, Brutus, Spike, SPI WebInspect Top 75 Security Tools: http://www.insecure.org/tools.html
Proxy	Achilles, WebProxy, Paros, SPI Dynamics