

But... We're Secure. We have a firewall!  
&  
Acme Art Hacked

## INTRODUCTION

---

---

---

---

---

---

---

*"Truth is one, but error proliferates. Man tracks it down and cuts it up into little pieces hoping to turn it into grains of truth. But the ultimate atom will always be an error, a miscalculation."*

René Daumal (1908-1944)  
French poet, critic

Slide 2

---

---

---

---

---

---

---

**"We're secure, we have a firewall..."**

- 80% of all reported attacks occur via TCP port 80 ([www.incidents.org](http://www.incidents.org))
- Worms spread like lightning: 40-90 minutes to propagate enough computers to have impact worldwide
- Simple tools, sophisticated attacks: laptop and a browser
- Security products that use signature-recognition can only protect against known threats

Slide 3

---

---

---

---

---

---

---

**“We’re secure, we have a firewall...”**

- 99% of all attacks exploit known vulnerabilities
- More than 19 million people have the skills to hack
- A Web (HTTP) server is - by design - a general purpose piece of software: HTTP servers "blindly" attempt to service **any** request from **any** client
- Applications are the weakest link

Slide 4

---

---

---

---

---

---

---

---

**To Err Is Human**

- If you don't already know it, nothing is truly secure
- Error is at the heart of every security breach
- No level of firewall, intrusion-detection system (IDS) or anti-virus software will make you secure
- This is reality! Accept it and move along!

Slide 5

---

---

---

---

---

---

---

---

**Writing on the Wall**

- There were warnings about “Web traffic through the firewall” back in 1999...  
<http://www.infoworld.com/articles/op/xml/99/08/09/990809opsecwatch.html>
- So, why all the fuss now?
  - People are understanding how a single vulnerability in a Web application can expose the entire company
  - Port 80 is a portal into your company

Slide 6

---

---

---

---

---

---

---

---

### Course Organization

- Book: "Web Hacking: Attacks & Defense" by Stuart McClure
- Taught from the perspective of the hacker, not the System Administrator
- 3 Main Parts:
  - Part 1: The E-Commerce Playground
  - Part 2: URL's Unraveled
  - Part 3: How Do They Do It?

---

---

---

---

---

---

---

---

### Case Study: Acme Art Hacked!

- Q: How is it that a hacker can intrude through a simple Web site?  
A: A little knowledge of URL's, Perl (or any other programming language) and UNIX provide the answer!
- Case study: How a hacker stole credit card numbers from Acme Art, Inc.
- The server logs reveal how the hacker did the deed. See "AcmeServerLog.txt"

---

---

---

---

---

---

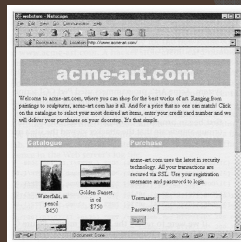
---

---

### Case Study: Acme Art Hacked!

- Let's replay the hackers moves:

- Hacker visits the Web site
  - Hacker sees Figure at right
  - Group A log generated




---

---

---

---

---

---

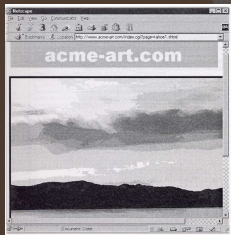
---

---

### Case Study: Acme Art Hacked!

- Let's replay the hackers moves (cont)

- Hacker clicks around a little
  - Figure at right shows an image click
  - Log shows some clicking around
  - Group B log generated



Slide 10

---

---

---

---

---

---

---

---

### Case Study: Acme Art Hacked!

- Let's replay the hackers moves (cont)

- Hacker tries to access /cgi-bin/
  - Gets an HTTP "403 Forbidden" error
  - Group C log generated
- Nothing yet... really...

Slide 11

---

---

---

---

---

---

---

---

### Case Study: Acme Art Hacked!

- Replay the hackers moves (cont):

- Hacker sees the first flaw: the URL
  - URL reveals there is a CGI script that loads pages: `index.cgi`
  - Hacker knows this by looking at the URL's in Group B
  - Asks `index.cgi` to reveal itself through `../index.cgi?page=index.cgi`
  - Since the browser simply thinks the response contains text, it prints the source for `index.cgi`!
  - Next figure shows what the hacker sees

Slide 12

---

---

---

---

---

---

---

---

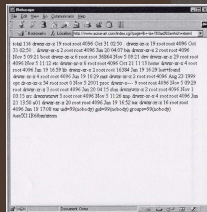


### Case Study: Acme Art Hacked!

● Replay the hackers moves (cont):

- Second flaw discovered by trial and error: what if I execute a UNIX shell command?

- Hacker uses the pipe character, "|", to pipe output from a shell command into a file that is opened by the Perl open()
- Commands are: `ls -la /`, `id`, which `xterm`
- The `%0a` is the LineFeed character ("Enter" in UNIX)
- Figure at right



See Group F Log entries

Slide 16

---

---

---

---

---

---

---

---

---

---

---

---

### Case Study: Acme Art Hacked!

● Replay the hackers moves (cont):

- What knowledge has the hacker gained from executing the 3 commands?
  - Listing of the files in / (`ls -al /`)
  - The user id of the process running `index.cgi` (`id`)
  - The path to an `xterm` (which `xterm`)
- Hacker now opens a window to the server over an `xterm` using the "nobody" account
  - See Log Group G
  - Next figure...

Slide 17

---

---

---

---

---

---

---

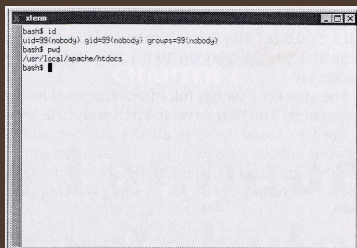
---

---

---

---

---



● Replay the hackers moves (cont):

- The hacker can now execute arbitrary commands on the Web server!
- YOU'VE BEEN HACKED!!!

Slide 18

---

---

---

---

---

---

---

---

---

---

---

---

### Case Study: Acme Art Hacked!

- ◉ Despite all the security audits, firewalls, strong password policies and whatever else, the hacker still gained access.
- ◉ How? Through poorly written applications!
  - Careless (lazy?) mistakes by programmers
  - What did the attacker need?
  - A little knowledge of HTTP, URL's, UNIX and Perl

Slide 18

---

---

---

---

---

---

---

---