Web Server-Side Security

CS @CU

Protecting the Server

- Servers are very tempting targets
- Defacement
- Steal data (i.e., credit card numbers)
- Distribute malware to unsuspecting clients

Standard Defenses

- Check all inputs
- Remember that *nothing* the client sends can be trusted
- Scrub your site

Server-Side Scripts

- Most interesting web sites use server-side scripts: CGI, ASP, PHP, server-side include, etc.
- Each such script is a separate network service
- For a web site to be secure, *all* of its scripts must be secure
- What security context do scripts run in? The web server's? How does the server protect its sensitive files against malfunctioing scripts?
- This latter is a particular problem with server plug-ins, such as PHP
- Partial defense: use things like `suexec`

Injection Attacks

- Often, user-supplied input is used to construct a file name or SQL query
- Bad guys can send bogus data
- Example: a script that sends email collects a username and executes `/usr/bin/sendmail username`
- The bad guy supplies `foo; rm -rf /` as the username
- The actual code executed is `/usr/bin/sendmail foo; rm -rf /`
- Oops...

Example: Webmail Server

- Assumption: general-purpose machine has a webmail server
- User mailboxes are under `/home/*/mail`. Folders are separate directories under that; each mail message is a separate file in a subdirectory. (This is very close to the CU-CS setup.)
- What needs tight filtering?

Filtering Webmail Requests

- The usename may not need too much filtering (except for SQL issues), becase it's authenticated against a list of valid users
- Folder names do need checking — what if a user specifies folder `../../../etc`?
- What if a user specifies `../../../etc/passwd` for a message file?

File Permissions

- What UID does the webmailer run as?
- It needs some privileges, to read everyone's mail folders
- But running as root is dangerous, because then it can be tricked into reading protected files:
  `ln /some/secret/file mail/inbox/42`

Scrubbing Your Site

- What is *really* being served?
- Web servers often come with default scripts — some of these are insecure
- Example: `nph-test-cgi` that used to come with Apache
- Example: proprietary documents; Google for them:

```
filetype:pdf "company confidential"
```

- (By the way, many documents have other, hidden data)
- Can Google for some other vulnerabilities, too

Users

- If your site permits user web pages — this deparment? — you have serious threats
- Are the user CGI scripts secure?
- Can users run PHP scripts in the browser's security context?
- Are all of these secure?

Email Security

The Usual Questions

- What are we trying to protect?
- Against whom?

Assets

- Confidentiality — people often discuss sensitive things via email
- Authenticity — who really sent the email?
- Anti-spam?
- Phishing?
- Authenticity has many motivations here

Secure Email

General Strategy

- Basic scheme is pretty straight-forward
- Encrypt the message body with a symmetric cipher, using a randomly-generated traffic key
- Use public key cryptography to encrypt the traffic key to all recipients
- Digitally sign a hash of the message
- But there are many details

Some Details

- Obvious ones: which symmetric, public key, and hash algorithms to use?
- More subtle: which algorithms do the recipients understand?
- Where do certificates come from?
- Do you sign the plaintext or the ciphertext?
- How do you handle BCC?
- Will the ciphertext survive transit intact?
- How are header lines protected?
- What about attachments?
- Many possible answers to all of these questions

Transit

- Not all mail systems accept all characters
- Cryptographic transforms won't survive even minor changes
- Very few are 8-bit clean
- EBCDIC vs. ASCII? Unicode? Tabs versus blanks?
- Solution: encode all email in *base 64*, using characters all systems accept: A-Za-z0-9+/
- Use 4 bytes to represent 3; overhead is 33%
- Only those characters matter; everything else is deleted on receipt, including white space

Signing

- If you sign the plaintext and then encrypt, the sender's identity is hidden from all except the proper recipients
- If you sign the ciphertext, a gateway can verify signatures and present mail accordingly — perhaps better for anti-spam and anti-phishing

Headers

- Headers change in transit
- Obvious example: `Received:` lines are added
- Less-obvious example: Email addresses are often rewritten to hide internal machines, and present clearer addresses to the outside: `smb@att.com` $\longrightarrow$ `Steven.Bellovin@att.com`
- Consequence: headers are *not* protected by secure email schemes

General Flow

- Collect input message
- Put in canonical form
- Encrypt and sign, or sign and encrypt
- Add metadata: encrypted traffic key, your certificate, algorithm identifiers, etc.
- Convert to transit form
- Embed in email message

Threats

CS@CU

Eavesdropping

- Most obvious way to read email: eavesdropping
- The bad guy "simply" listens to the network
- Harder than it sounds, except for some wireless nets
- Frequently used by police and intelligence agencies, i.e., the FBI's *Carnivore* device

Password Theft

- Most email is retrieved by login and password
- Anyone who gets your password can read your email
- It's much easier for an eavesdropper to pick those up — passwords are usually sent each time someone polls for new email

Hacking

- The real threat to email is while it's in storage
- This can be temporary storage, waiting for you to pick it up
- It can also be your personal machine, for email you've sent or received
- What if your laptop is stolen? Does it have plaintext copies of all the secure email you've sent and received?

Screen Dumps

- Connect via X11
- Use some other Trojan horse software to dump user's screen periodically
- Reflection off the back wall...

Subpoena Attacks

- What if you records are subpoenaed?
- This is a legal issue; technical wiggling won't help!

Rubber Hose Cryptanalysis

- What if the local secret police want to know what some intercepted email says?
- Protecting human rights workers was one of the original goals for PGP!
- It's public key-encrypted — you *can't* read it
- If the signature is encrypted, they can't even prove you sent it
- Of course, people like that don't care much about proof, and they don't like to take "no" for an answer...

Spoofing

- Ordinary email is trivial to spoof
- On timesharing machines and web mailers, the systems can tack on the userid
- On PCs, individuals set their own addresses
- *No* security — if you need to authenticate email, you have to use crypto

Systems Issues

- Only read email on secure machines
- Only connect to them securely
- Watch out for buggy mailers and systems
- But if the process of reading secure email is too cumbersome, your email will be insecure, because you'll never use the secure version
- Finding the right tradeoff is a difficult engineering choice

PGP and S/MIME

CS@CU

Approaches

- Two major standards, PGP and S/MIME
- Many minor syntactic differences
- Major split by audience: computer scientists like PGP; mainstream users use S/MIME
- Biggest technical difference: how certificates are signed

Certificate Style

- S/MIME uses standard X.509 certificate format
- More importantly, X.509 certificates form a traditional PKI, with a root and a hierarchical structure
- Works well within an organization
- Between organizations, can work if it's easy to find that organization's root
- CU has no PKI — what is the PKI under which you'd find my cert? Why should you trust its root?

Web of Trust

- PGP use a "web of trust"
- *Anyone* can sign a certificate
- Most people have more than one signature — I have 65 signatures on my primary PGP key
- Do you know and trust any of my signers?
- See my key at

  `http://www.cs.columbia.edu/~smb/smbpgp.txt`

Does the Web of Trust Work?

- Number of signatures alone is meaningless; I can create lots of identities if I want
- I can even forge names — is the "Angelos Keromytis" who signed my key the same one who's a professor here? How do you know?
- There are at least six PGP keys purporting to belong to "George W. Bush". One is signed by "Yes, it's really Bush!"
- You have to define your own set of trust anchors, as well as policies on how long a signature chain is too long

Finding Public Keys

- Many mailers cache received certificates
- Some organizations list people's certificates in an LDAP database
- Some people have them on their web site
- For PGP, there are public key servers — anyone can upload keys
- Is that safe? Sure — the security of a certificate derives from the signature, not from where you found it

Which Style is Better?

- PGP was easier to start — it doesn't need an infrastructure
- Many security and network conferences have "PGP key-signing parties"
- S/MIME is better for official use — it makes it clearer when someone is speaking in an organizational role, since the organization issued the certificate.
- Both have usability issues, though PGP is probably worse

Phishing

What is Phishing?

- Spoofed emails, purportedly from a financial institution
- Ask you to login to "reset" or "revalidate" your account
- Often claim that your account has been suspended

A Phish


```
From: no-reply@flagstarbanking2.com
To: undisclosed-recipients:;
Subject: YOUR ACCOUNT HAS BEEN SUSPENDED !!!
Date: Fri, 29 Sep 2006 09:29:25 -0500

...

If you fail to provide information about your
account you'll discover that your account has been
automatically deleted from Flagstar Bank database.

Please click on the link below to start the update
process:

https://www.flagstar.com/Signon.cgi?update
Flagstar Bank
```

What's Wrong?

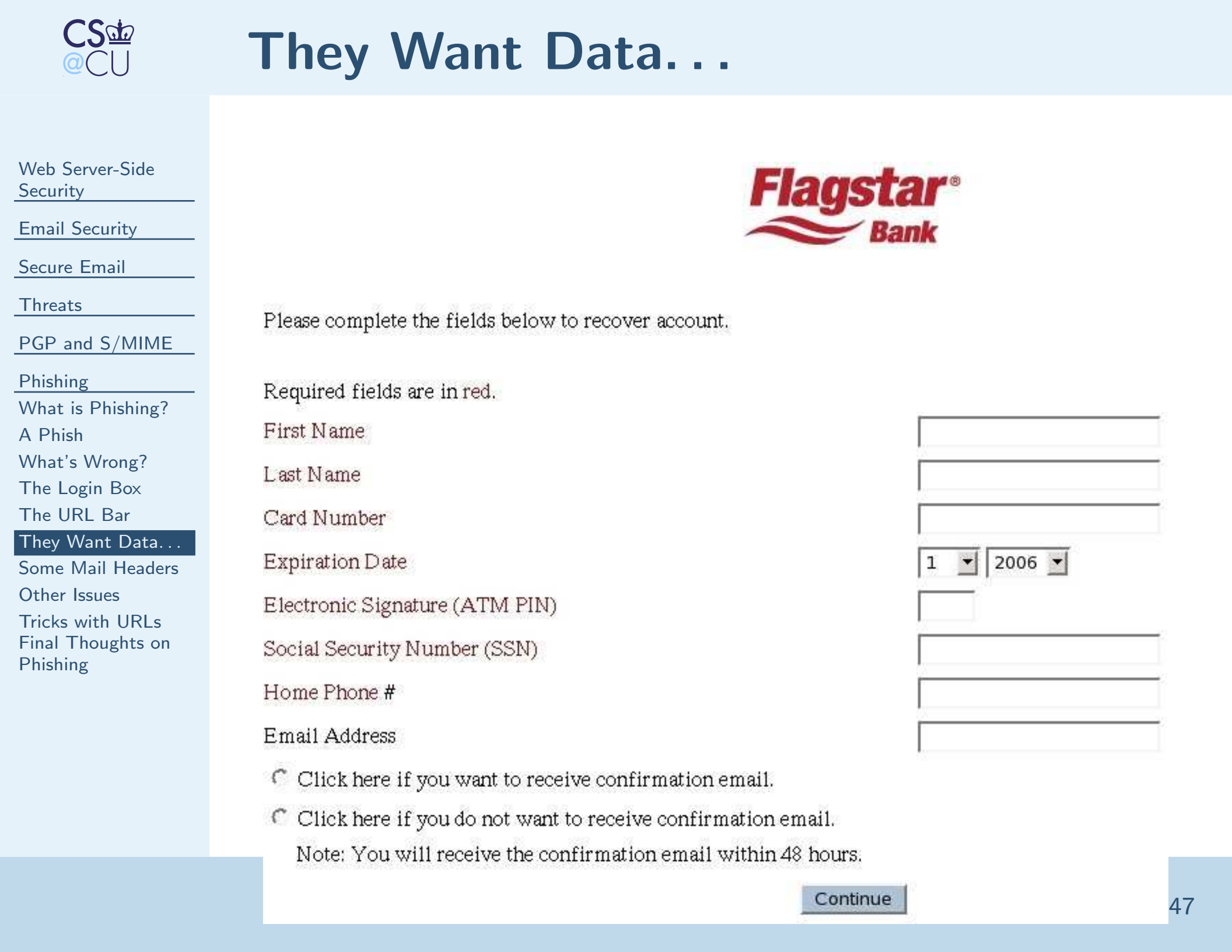
- The URL is a booby trap:

- When I clicked on it, I was actually redirected to a site in Colombia, via yet another indirection...
- The login page appears identical to the real one
- (One of the web sites I visited seemed to have several variant "bank" pages)

The Login Box

The URL Bar

They Want Data...



Some Mail Headers


```
Received: from plesk.salesforcefoundation.org
   ([198.87.81.9])
   by cs.columbia.edu (8.12.10/8.12.10)
   (version=TLSv1/SSLv3 cipher=DHE-RSA-AES256-SHA
   bits=256 verify=NOT) for <smb@cs.columbia.edu>
Received: from adsl-68-20-44-198.dsl.chcgil.amerite
  (68.20.44.198) by 198.87.81.11
```

Where does `plesk.salesforcefoundation.org` come from? It is *asserted* by the far side. The `198.87.81.9` is derived from the IP header, and is hard to forge (but stay tuned for routing attacks, in a few weeks). A DNS lookup on `198.87.81.9` isn't very helpful; the mapping is controlled by the address owner, not the name owner.

Other Issues

- Why is the email from `flagstarbanking2.com`?
- The domain for the bank is `flagstar.com` — no "ing" and no "2".
- *That's legit!* — the real web site for their online service is `flagstarbanking2.com`
- We have trained users to accept weird, seemingly gratuitous differences; it can make life easier for the phisher

Tricks with URLs

- `http://cnn.com@some.other.site/foo`
  `cnn.com` is a userid
- `http://2151288839/foo`
  2151288839 is 128.58.16.7, `cluster.cs.columbia.edu`
- `http://rds.yahoo.com/_ylt=A0g...http%3a/`
  So the search engine knows what you clicked on

Final Thoughts on Phishing

- We have the basic technical mechanisms to authenticate email and web sites
- Human interaction with these mechanisms remains a very challenging problem
- Security is a *systems problem*