

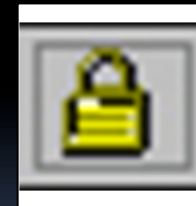
Michael Coates  
mcoates@mozilla.com  
michael-coates.blogspot.com

SSL SCREW UPS

# Who am I?

- Web Security Engineer @ Mozilla
- Contributor OWASP 2010 Top 10
- Author OWASP TLS Cheat Sheet
- Creator & Leader OWASP AppSensor
- Security Blogger  
<http://michael-coates.blogspot.com>

# SSL: Super Shiny Locks



# Padlock != Secure

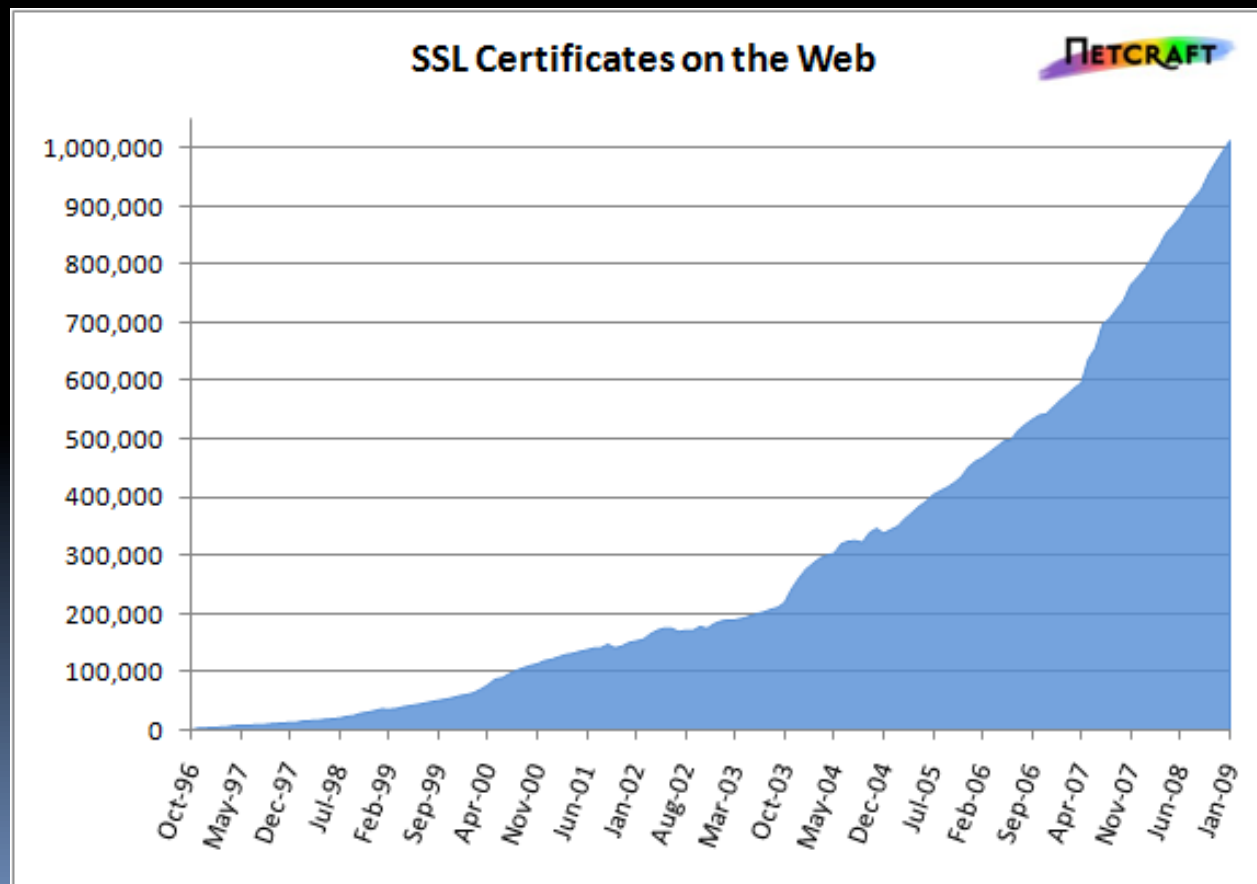
The image displays four different web login interfaces:

- Monster.com:** Features the logo "monster" with the tagline "Your calling is calling". Navigation links include "Home" and "Profile & Resume". A "Security Center" link is visible.
- Scottrade:** Shows a "Secure Login" window with fields for "Account #:", "Password:", "Language Preference:" (set to English), and "Start Page:" (set to Home Page). A "LOGIN" button and a "Scottrade Brokerage Account Agreement" link are present.
- Chase:** Prompts users to "Access your account online" and "Get a User ID" with a "GO" button. It also has a section for "Returning Users: Log On" with fields for "User ID:" and "Password:", and checkboxes for "Remember my Customer Number/Saver ID" and "Hide my typing".
- PayPal:** Shows a "PayPal, Inc. (US)" header and a "PayPal" logo. Navigation includes "Home", "Personal", and "Business". A "Get Started" section includes "Send Money". The "Account login" section has fields for "Email address" and "PayPal password", and a "Go to" dropdown menu set to "My account".

A large padlock icon is overlaid on the Scottrade page, and a "Customer Number or Saver ID" window is overlaid on the Chase page.

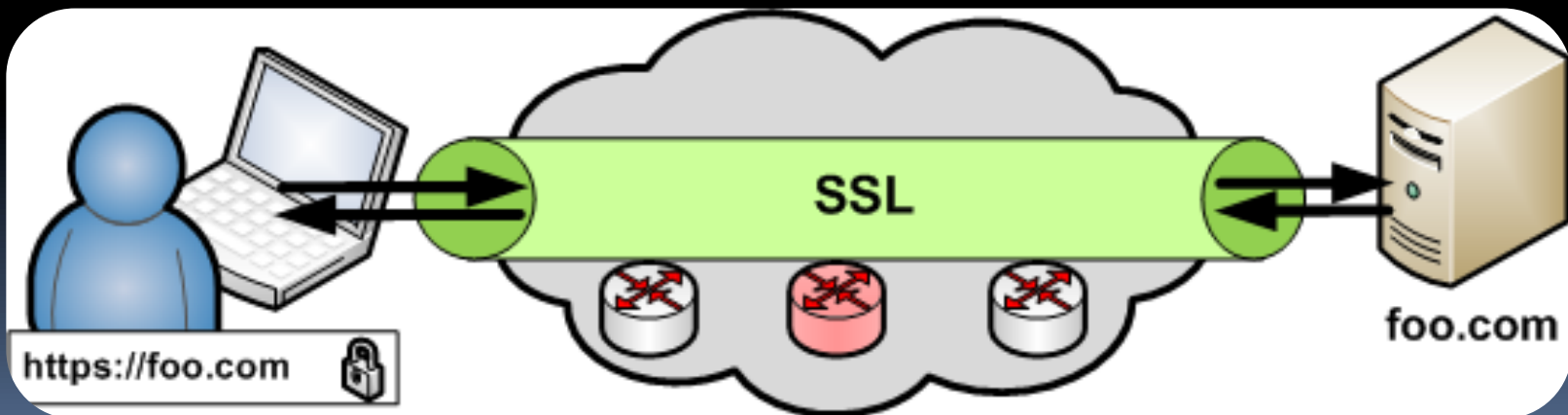
# SSL Growth

- > 1 Million SSL Certificates



# The Good

- Confidentiality
- Integrity
- Replay Protection
- End Point Authentication



# Problem: Usability



**The security certificate presented by this website was not issued by a trusted certificate authority**



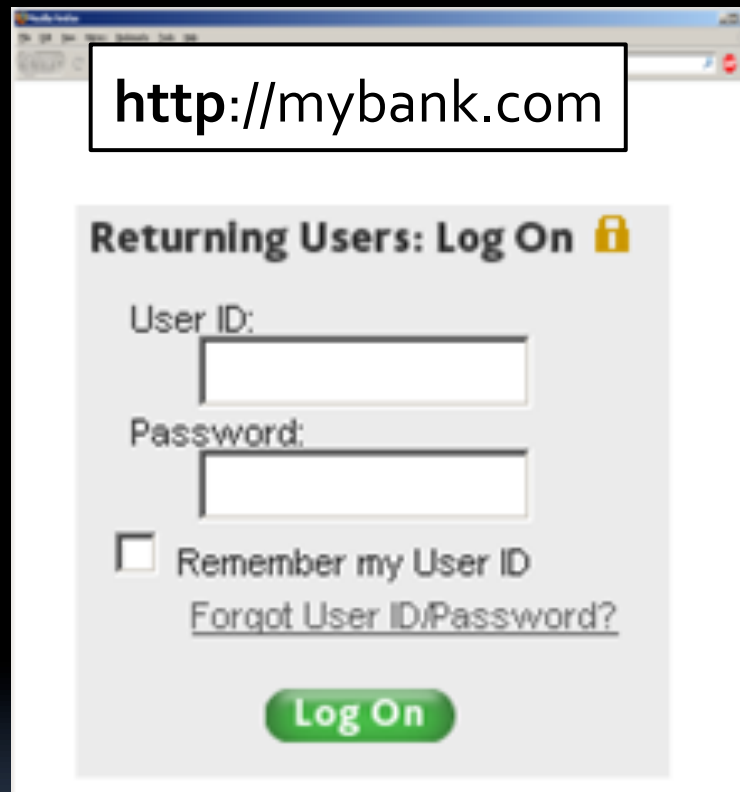
**The security certificate presented by this website was issued for a different website's address.**

# Problem: User Expectations

- How did you get to the site?
- Is HTTPS in the URL?
- Are those 0's or 0's?
- Did you get any browser warning messages?
- Did you click “ok” or “accept” to any popup boxes?



# Scenario: Insecure Landing Page



**http://mybank.com**

**Returning Users: Log On** 🔒

User ID:

Password:

Remember my User ID

[Forgot User ID/Password?](#)

**Log On**

```
<form method="POST" action="https://mybank.com/login" >  
  Username: <input type="text" name="user"> <br>  
  Password: <input type="password" name="pass"> <br>  
</form>
```

# Exploiting Insecure Landing Page



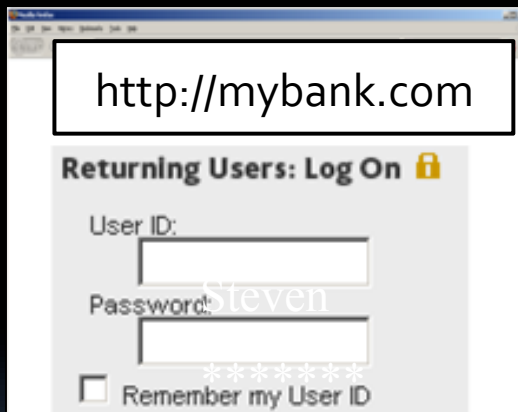
HTTP REQUEST  
GET http://mybank.com



mybank.com



HTTP Response



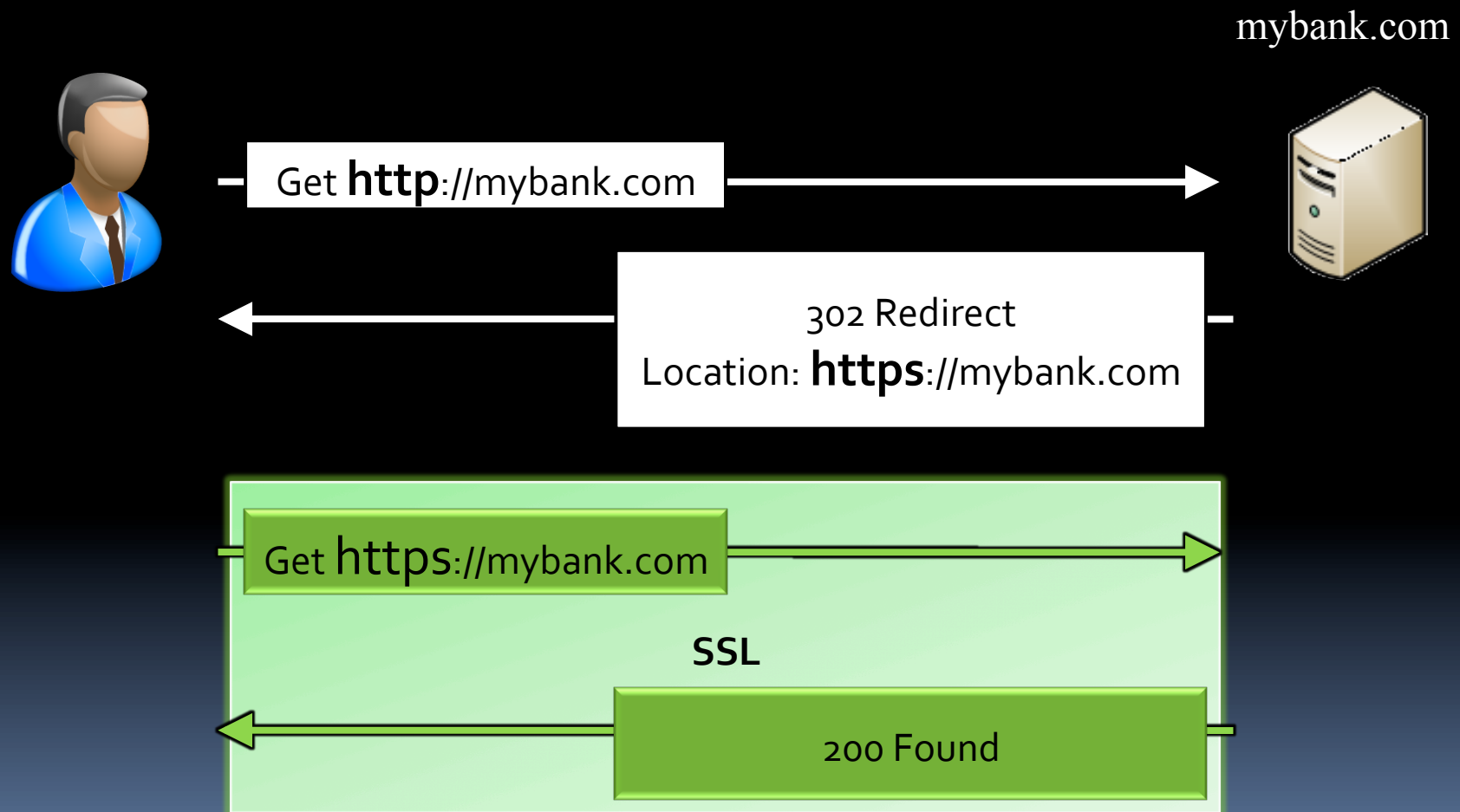
```
...  
<form method="POST"  
action="https://mybank.com/  
login" >  
...
```

POST **http**://mybank.com  
user:Steven&pass:JOSHUA

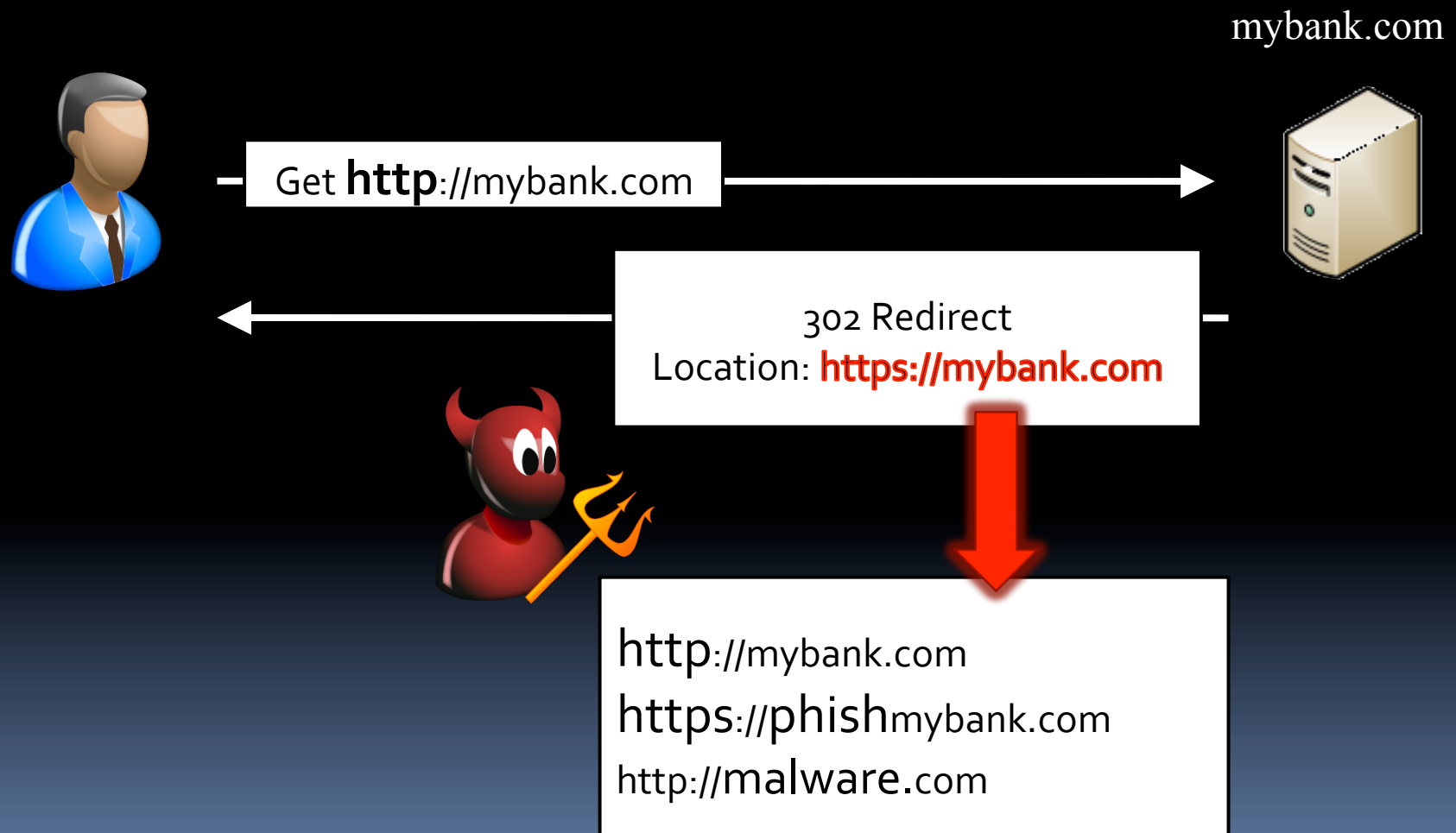
# Problem: Insecure Redirects



# Insecure Redirects – Behind The Scenes



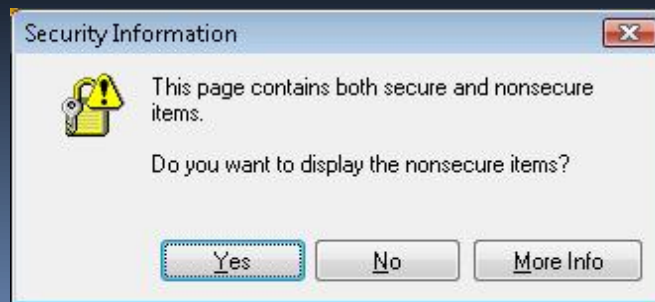
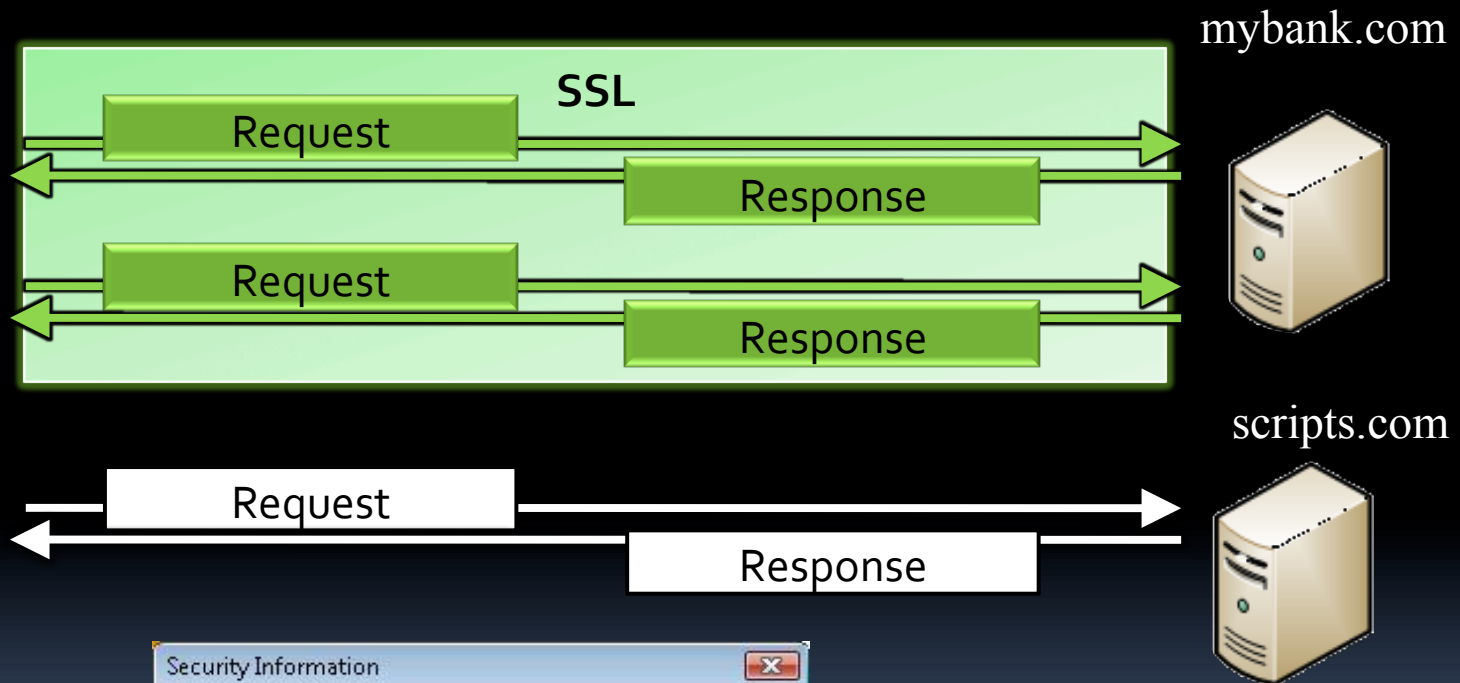
# Exploiting Insecure Redirects



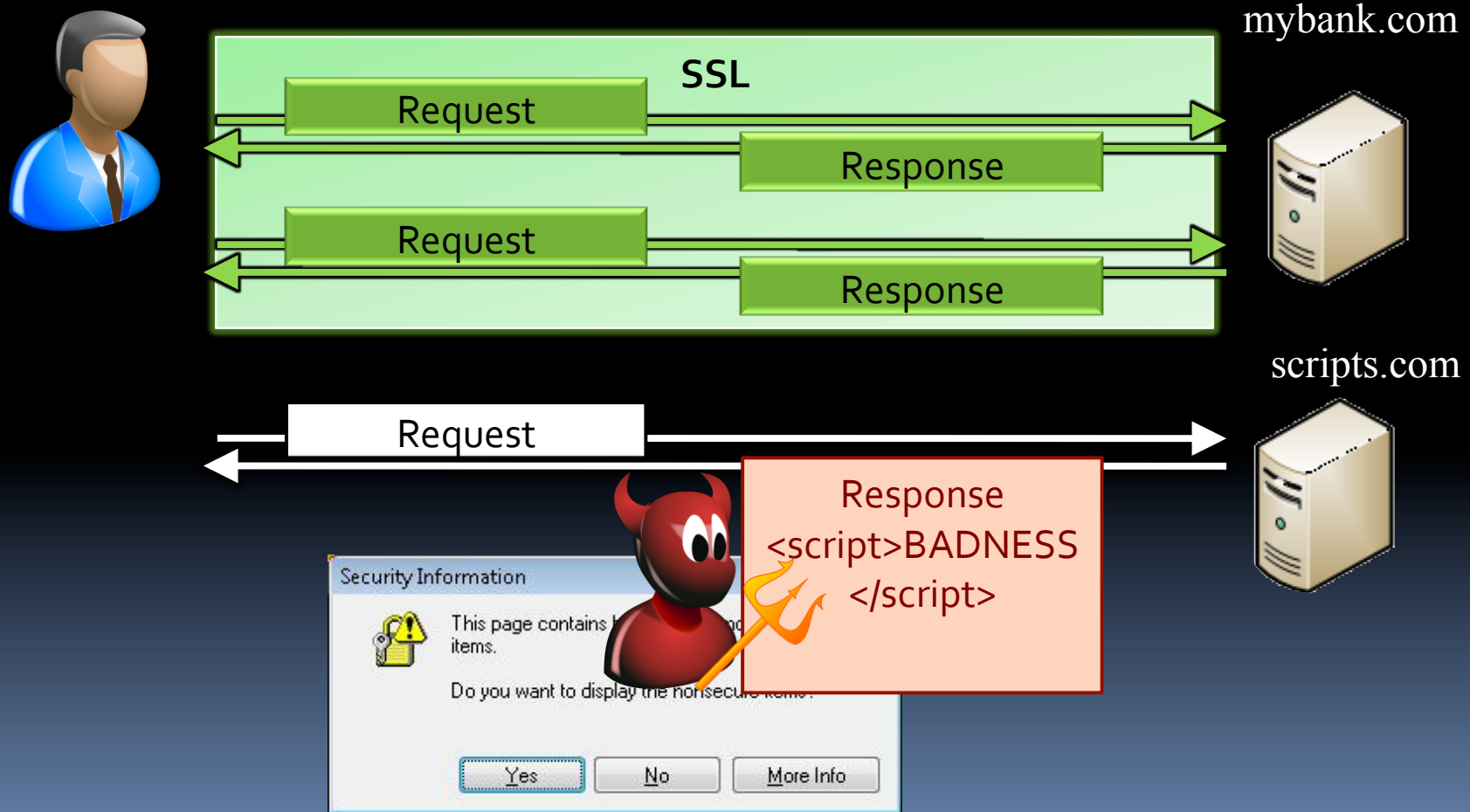
# Insecure Redirects via Google

- “Bank of America”
  - <http://www.bankofamerica.com/>
- “Chase”
  - <http://www.chase.com/>
- “Wachovia”
  - <http://www.wachovia.com>
  - Cookie set on HTTP response too!
- “Wells Fargo”
  - <http://www.wellsfargo.com/>

# Scenario: Insecure Content

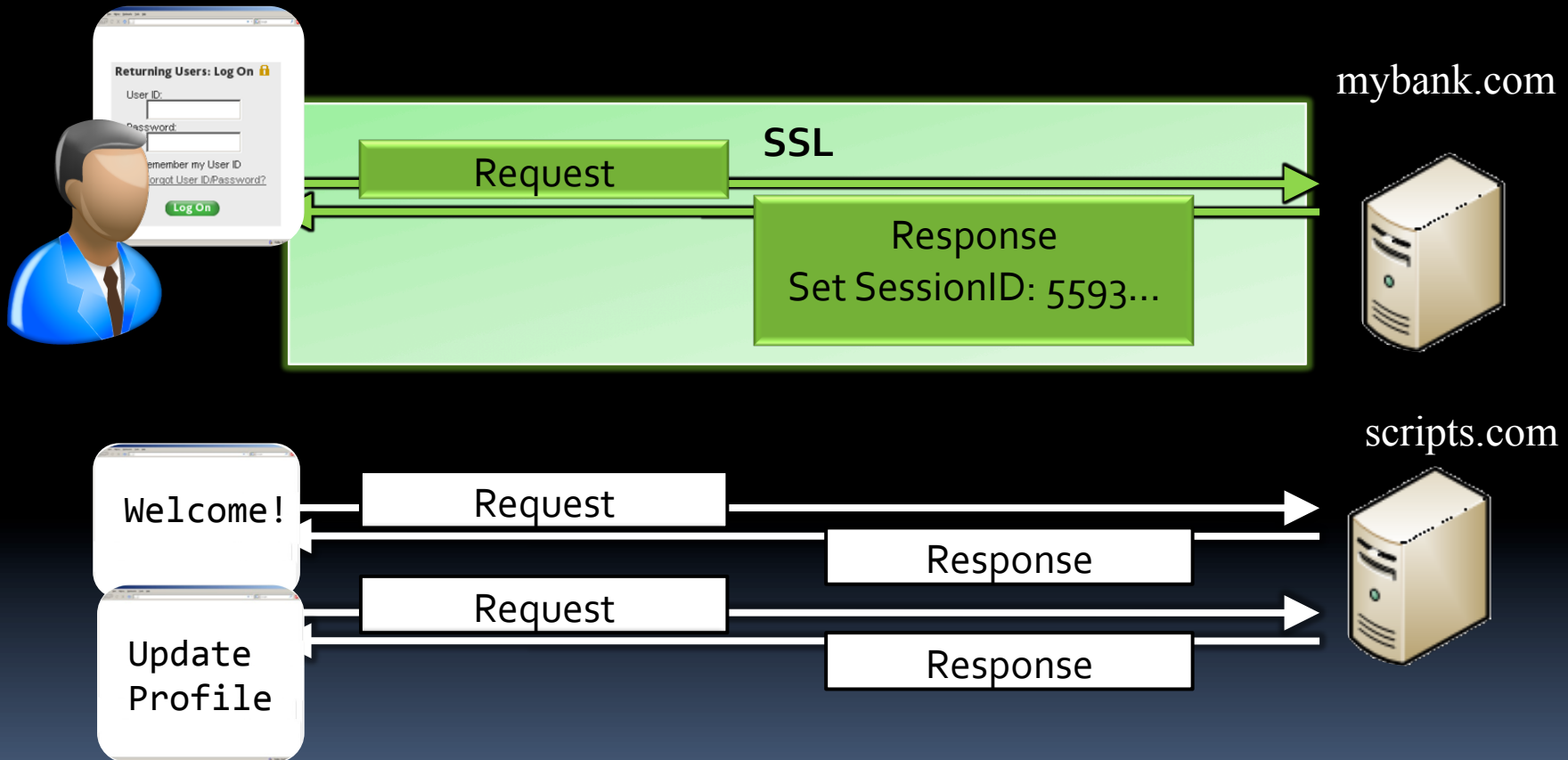


# Exploiting Insecure Content

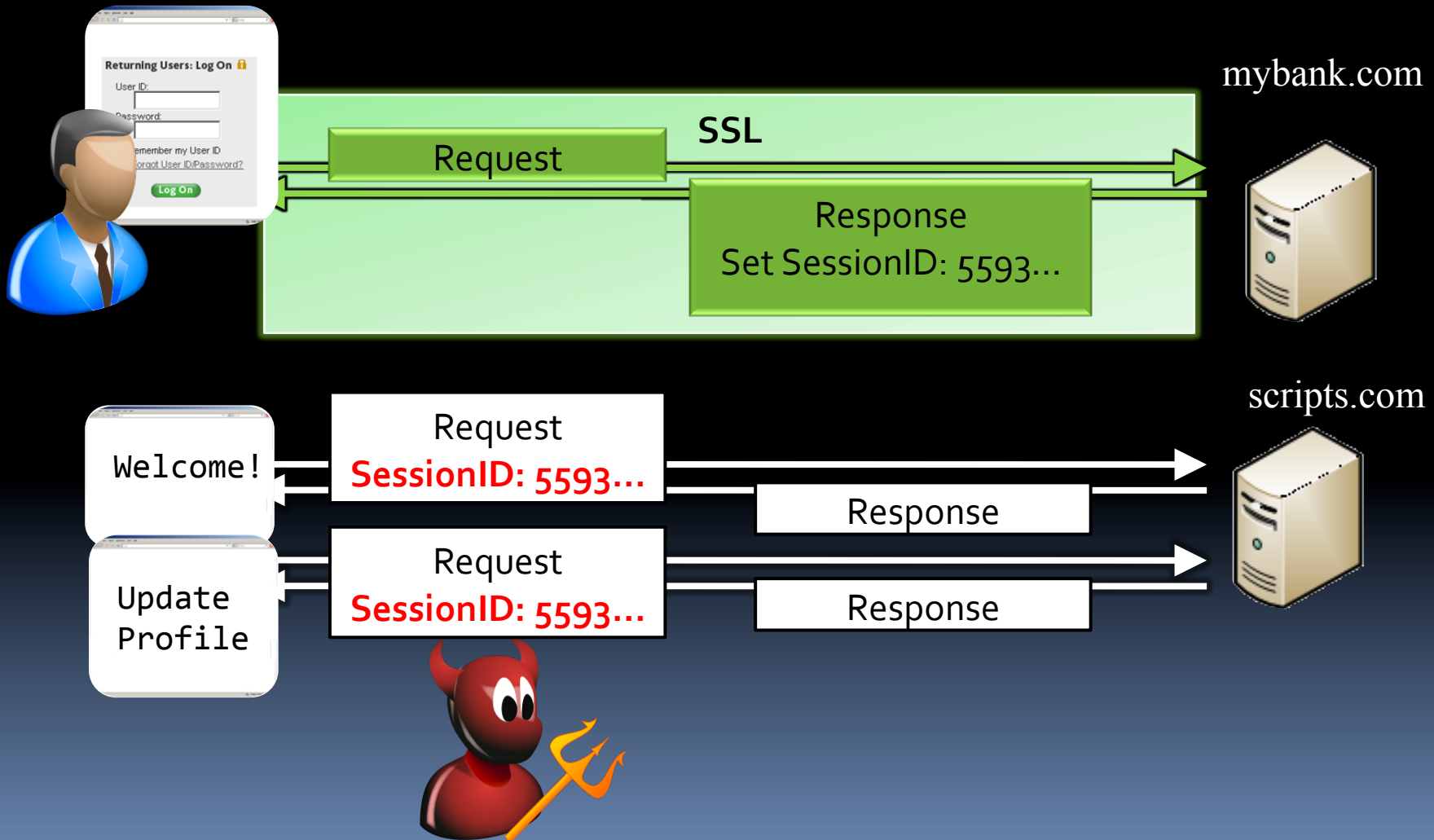




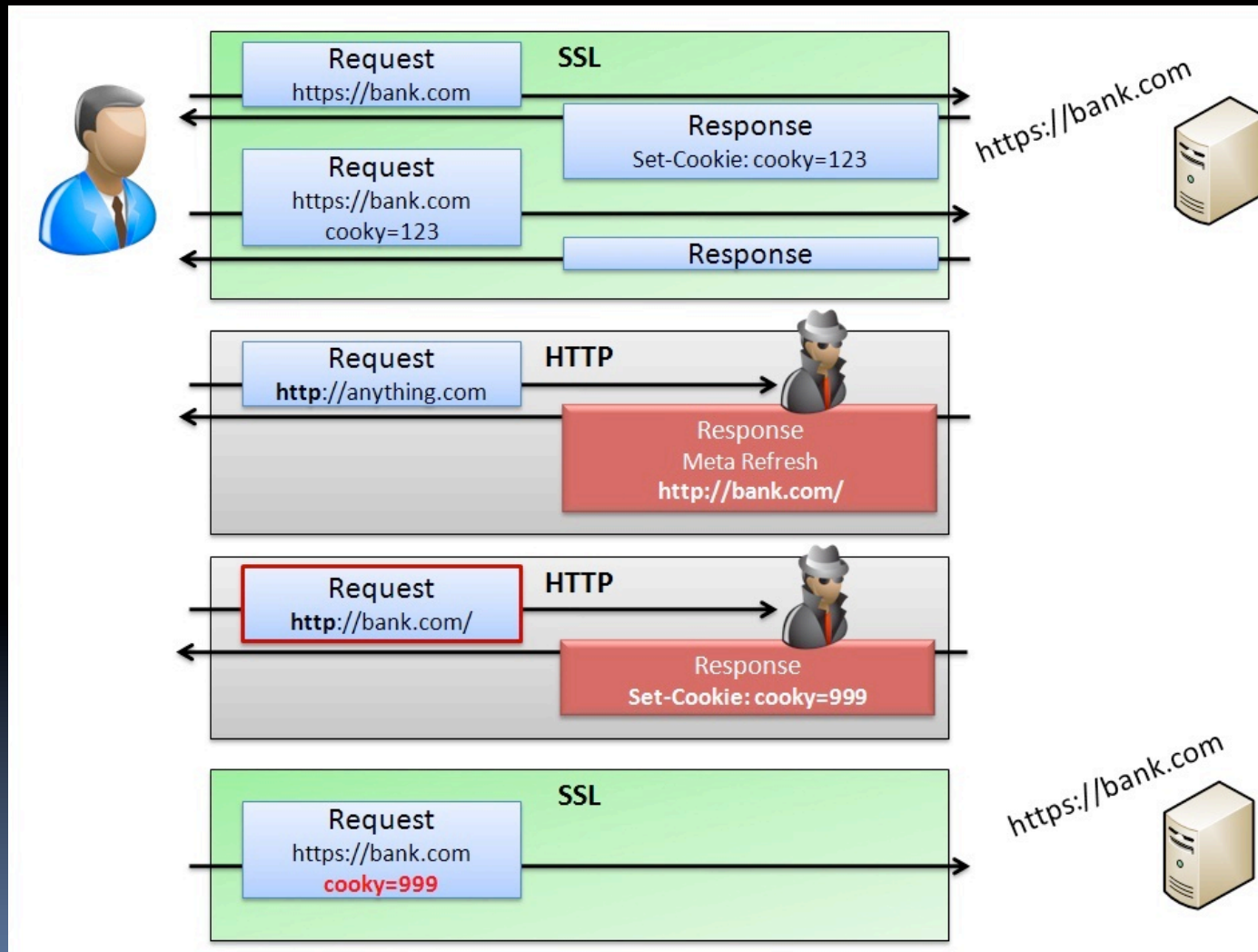
# Scenario: HTTP after Login



# Exploiting HTTP after Login



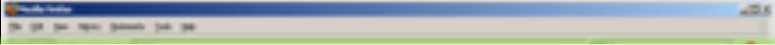
# Problem: Cookie Forcing



# Problem URL Leakage

Transition SiteA.com to SiteB.com	Expectation	Result
HTTP->HTTP	Referrer Leaked	Referrer Leaked
HTTP->HTTPS	Referrer Leaked	Referrer Leaked
HTTPS->HTTP	Referrer Secure	Referrer Secure
<b>HTTPS-&gt;HTTPS</b>	<b>Referrer Secure</b>	<b>Referrer Leaked</b>

# Exploiting URL Leakage

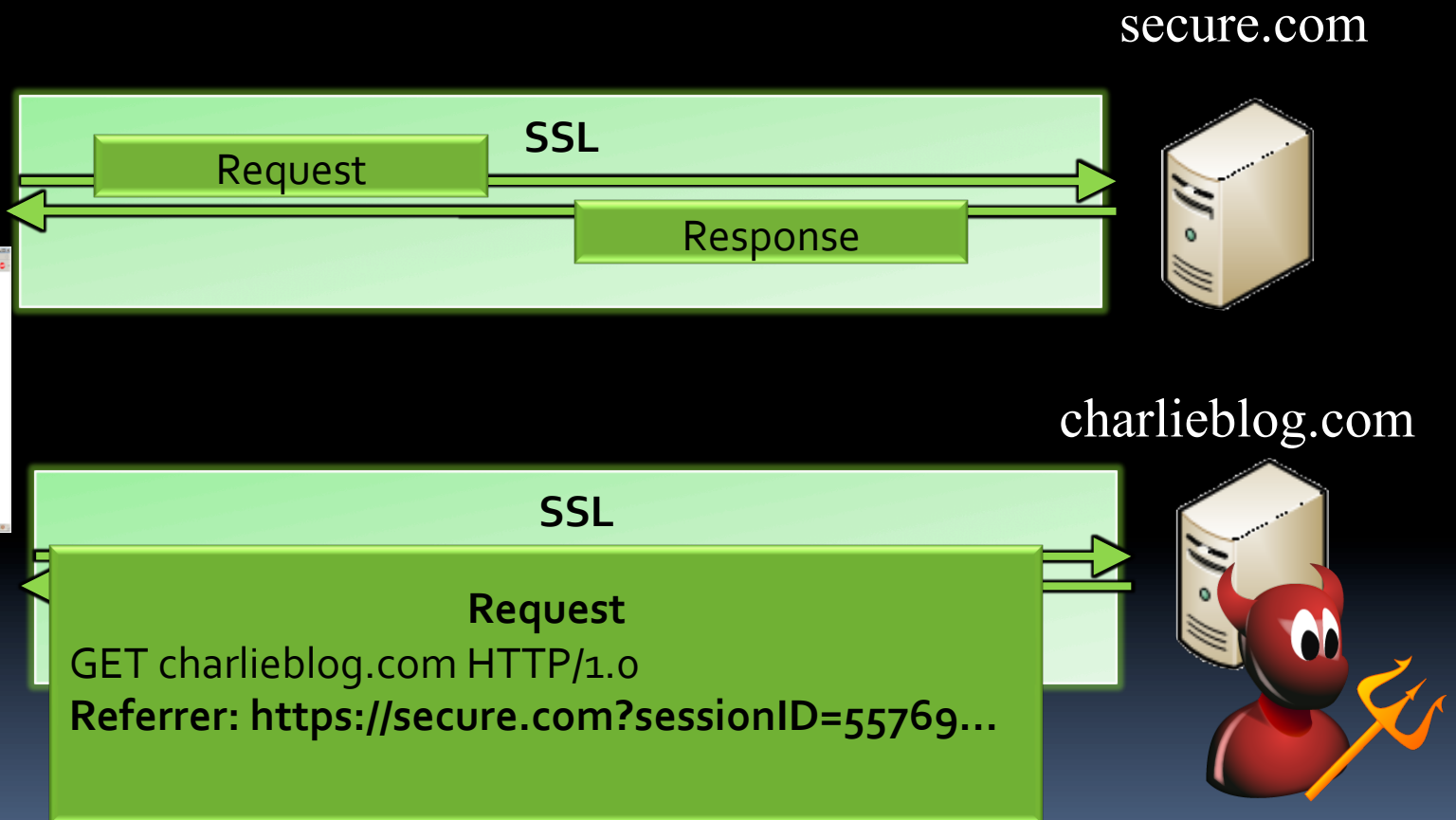
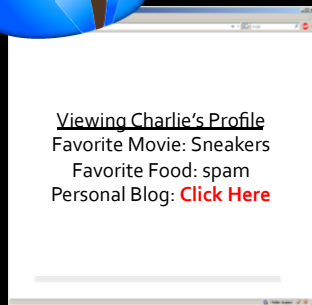


`https://secure.com?sessionID=55769...`

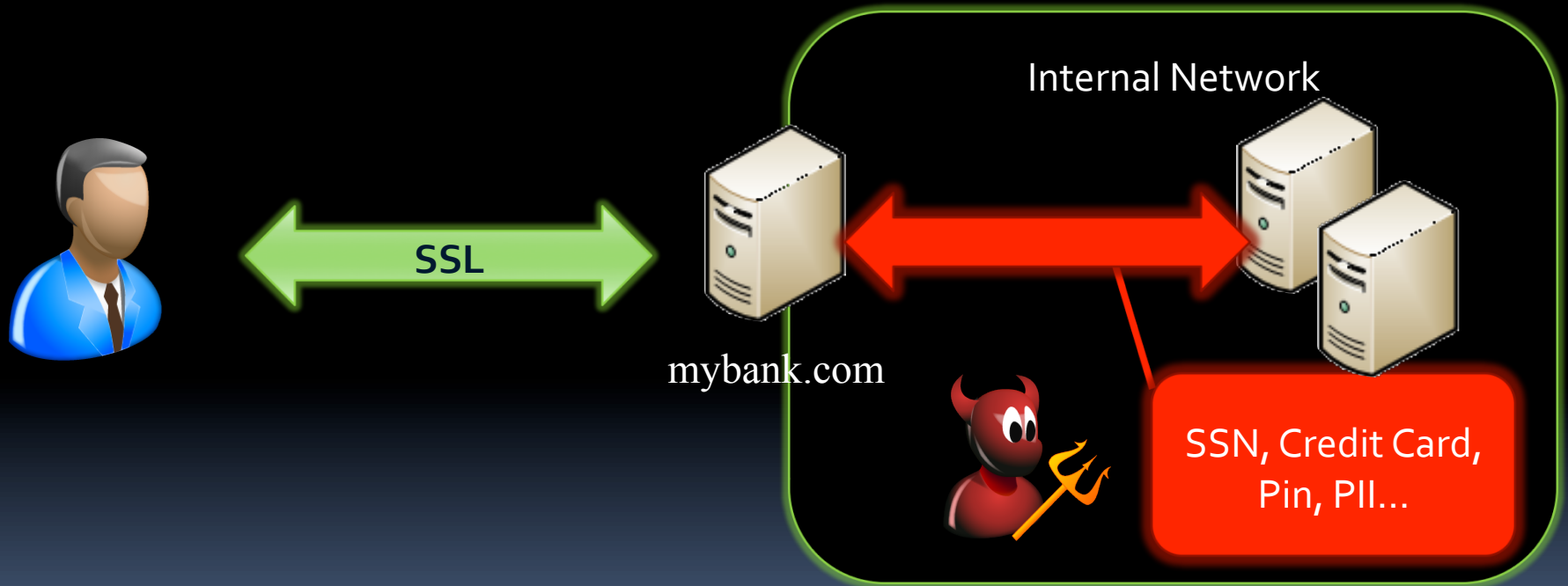
Viewing Charlie's Profile  
Favorite Movie: Sneakers  
Favorite Food: spam  
Personal Blog: **Click Here**

```
<a href="https://charlieblog.com">Click Here</a>
```

# Exploiting URL Leakage



# Problem: False Internal Trust



# Problem: Not all SSL is equal

- View Ciphers by Strength

```
openssl ciphers <strength> -v
```

- Test Server:

```
openssl s_client -connect site.com:443 -  
cipher <strength>
```

- Test Client:

```
openssl s_server -www -cert cacert.pem -  
key cakey.pem
```

<strength>=NULL|LOW|MEDIUM|HIGH|FIPS

## FIPS Approved Ciphers

ADH-AES256-SHA

DHE-RSA-AES256-SHA

DHE-DSS-AES256-SHA

AES256-SHA

ADH-AES128-SHA

DHE-RSA-AES128-SHA

DHE-DSS-AES128-SHA

AES128-SHA

ADH-DES-CBC3-SHA

EDH-RSA-DES-CBC3-SHA

EDH-DSS-DES-CBC3-SHA

DES-CBC3-SHA

## LOW Strength Ciphers

ADH-DES-CBC-SHA

EDH-RSA-DES-CBC-SHA

EDH-DSS-DES-CBC-SHA

DES-CBC-SHA

DES-CBC-MD5



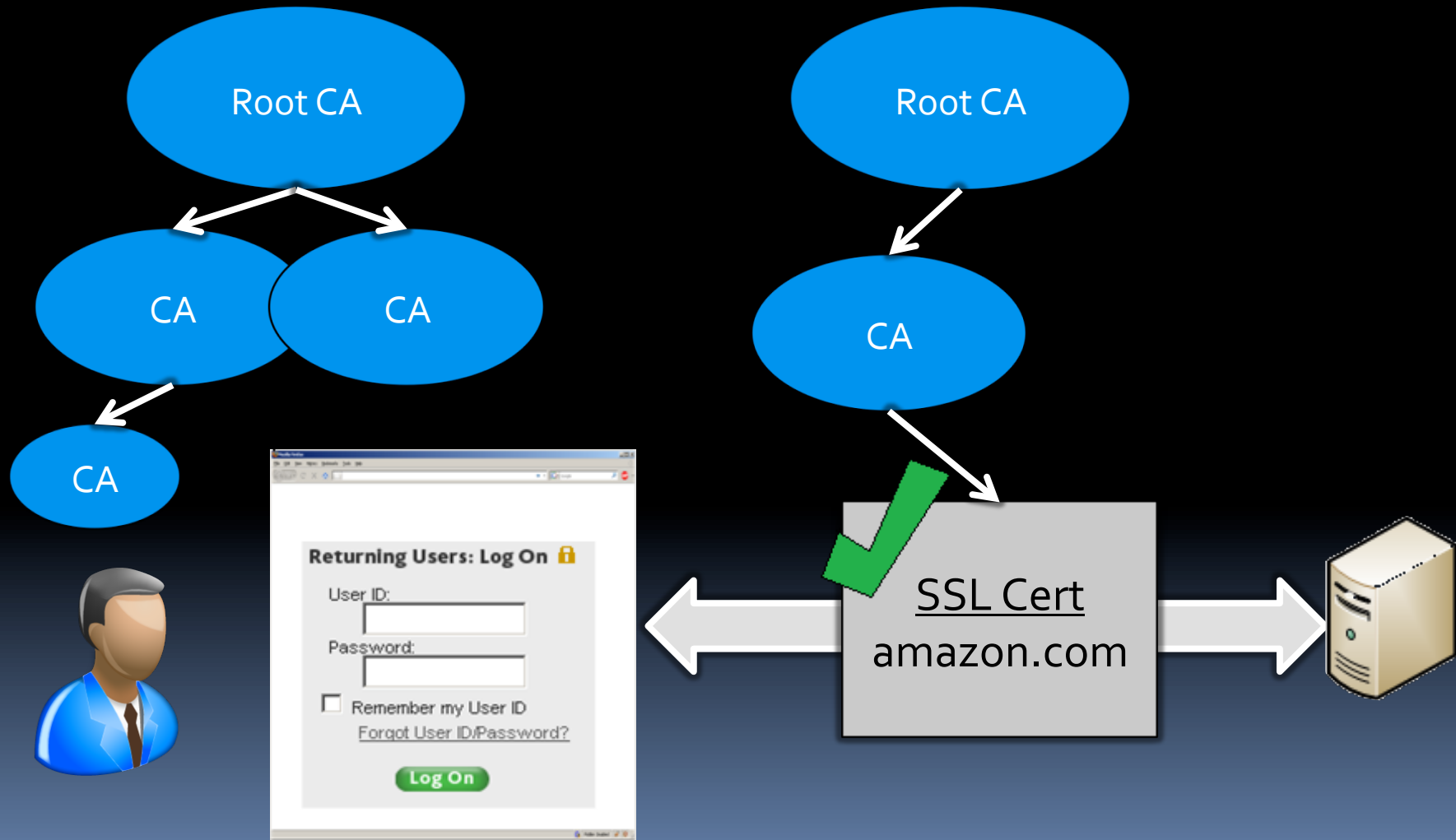
# More Problems

- MD5 Collision Rogue CA Creation
  - Alexander Sotirov, Marc Stevens, Jacob Appelbaum, Arjen Lenstra, David Molnar, Dag Arne Osvik, Benne de Weger
  - <http://www.win.tue.nl/hashclash/rogue-ca/>
- SSLstrip
- Null Prefix Attacks Against SSL/TLS Certificates
  - Moxie Marlinspike
  - <http://www.thoughtcrime.org/software/sslstrip/>
  - <http://www.thoughtcrime.org/papers/null-prefix-attacks.pdf>

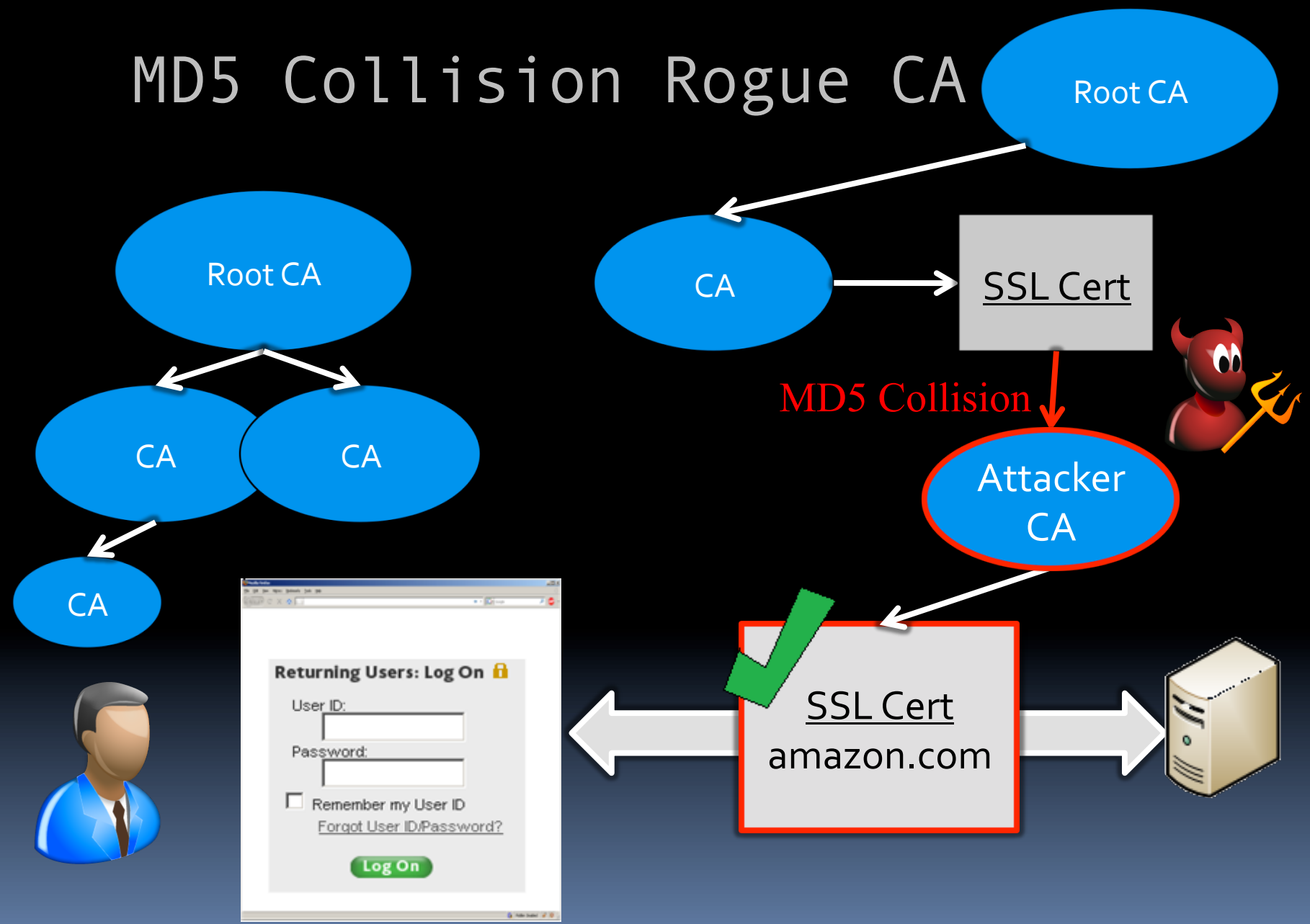
# MD5 Collision

- Attacker requests legitimate cert from CA
- Exploits MD5 Collision to create legitimate CA
- Issues legit certs from authorized CA

# MD5 Collision Rogue CA



# MD5 Collision Rogue CA



# Null Prefix Attack

CA Verifies Root Domain Ownership

www.foo.com

www.blah.foo.com

nonexistent.a.b.c.foo.com

amazon.com\0.foo.com

foo.com

Browser SSL Verification

▪ Microsoft CryptoAPI - \0 is eos

amazon.com == amazon.com\0.foo.com

# SSLstrip

- MitM SSL Connections
  - ARP Spoofing
  - IP Tables
- Auto Strip SSL -> HTTPS to HTTP
- Execute Null Prefix Attack
- Block Certificate Revocation Messages
  - OCSP Attacks

# Is There Hope?

- Average User == Not Technical
- Most Deployments Vulnerable
- Specialized Attack Tools Available

# Doing It Right...



## The Application

- SSL only
- No HTTP -> HTTPS redirects : HTTP shows "User Education" message
- No SSL errors or warnings

## The User

- Bookmark the HTTPS page
- Stop if any SSL warnings/errors presented

## The Browser

- Set realistic user expectations
- Support STS/ForceTLS



# Solution: Strict Transport Security

- Server Side Option
- Header tells browser to only send HTTPS requests for site
- Blocks Connection w/any Errors

HTTP/1.1 200 OK

Server: Apache

Cache-Control: private

Strict-Transport-Security : max-age=500; includesubdomains

# Resources – TLS Cheat Sheet

Transport Layer Protection Cheat Sheet - OWASP - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.owasp.org/index.php/Transport\_Layer\_Protection\_Cheat\_Sheet

Transport Layer Protection Cheat Sheet

page discussion view source history

Log in

Contents [hide]

- 1 Introduction
  - 1.1 Architectural Decision
- 2 Providing Transport Layer Protection
  - 2.1 Benefits
  - 2.2 Basic Requirements
  - 2.3 SSL vs. TLS
  - 2.4 When to Use a Firewall
  - 2.5 Secure Server Design
    - 2.5.1 Rule - Use TLS for All Login Pages and All Authenticated Pages
    - 2.5.2 Rule - Use TLS on Any Networks (External and Internal) Transmitting Sensitive Data
    - 2.5.3 Rule - Do Not Provide Non-TLS Pages for Secure Content
    - 2.5.4 Rule - Do Not Perform Redirects from Non-TLS Page to TLS Login Page
    - 2.5.5 Rule - Do Not Mix TLS and Non-TLS Content
    - 2.5.6 Rule - Use "Secure" Cookie Flag
    - 2.5.7 Rule - Keep Sensitive Data Out of the URL
  - 2.6 Server Certificate Management
    - 2.6.1 Rule - Use TLS for All Login Pages and All Authenticated Pages
    - 2.6.2 Rule - Only Use TLS for Sensitive Data
    - 2.6.3 Rule - Only Use TLS for Sensitive Data
    - 2.6.4 Rule - Use TLS for Sensitive Data
    - 2.6.5 Rule - Use TLS for Sensitive Data
  - 2.7 Client (Browser) Configuration
  - 2.8 Additional Controls
    - 2.8.1 Extended Validation
    - 2.8.2 Client-Side Certificate
- 3 Providing Transport Layer Protection for Back End and Other Connections
  - 3.1 Secure Internal Network Fallacy
- 4 Related Articles
- 5 Authors and Primary Editors

[http://www.owasp.org/index.php/Transport\\_Layer\\_Protection\\_Cheat\\_Sheet](http://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet)

# Resources - ssllabs.com



## Recently Seen

[amazon.com](#)  
[chase.com](#)  
[bankofamerica.com](#)  
[gmail.google.com](#)

## Recent Best-Rated

**B (67)** [sparklit.com](#)  
**B (72)** [www.startssl.org](#)  
**C (60)** [ais2.uniba.sk](#)  
**C (64)** [blog.startcom.org](#)

## Recent Worst-Rated

**A (91)** [webmail.verto.com.br](#) **F (0)**  
**A (88)** [webmail.stiefel.com](#) **F (0)**  
**A (88)** [www.kaching.com](#) **F (0)**  
**A (88)** [imperva.com](#) **F (0)**

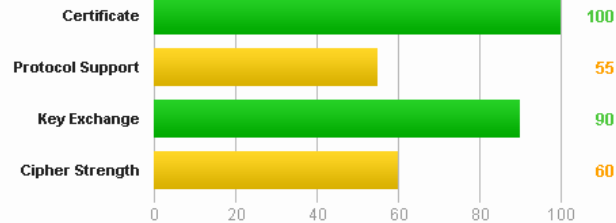
## SSL Report: [amazon.com](#) (72.21.207.65)

### Summary

Overall Rating



67



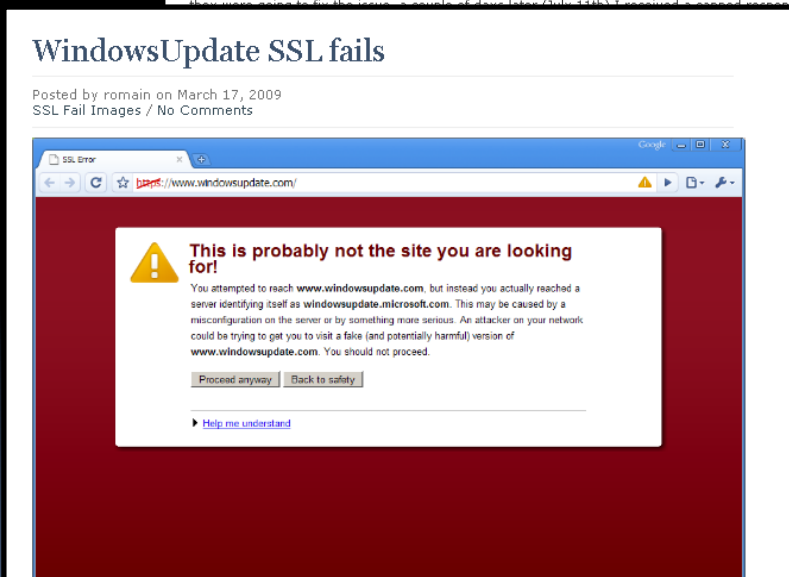
The scores are explained in the [SSL Server Rating Guide 2009](#).

# Resources – sslfail.com

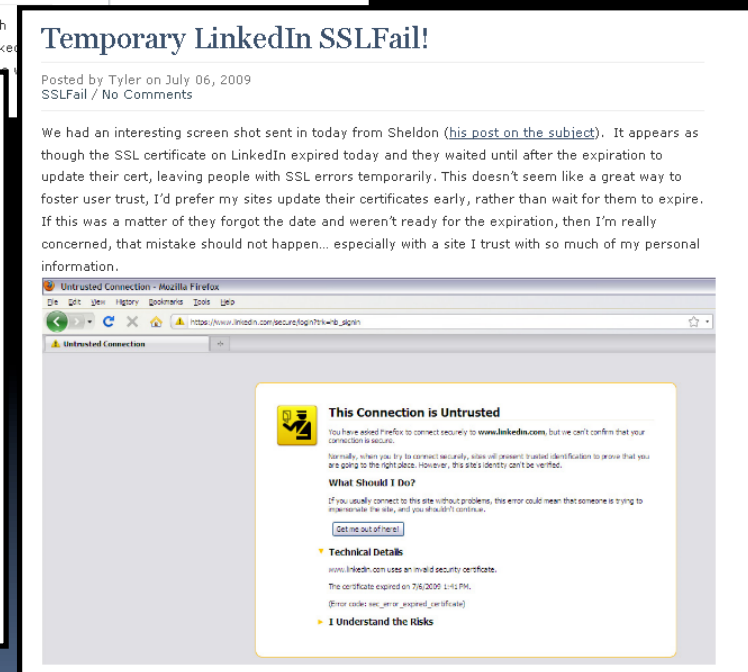
(Tyler Reguly, Marcin Wielgoszewski)



The screenshot shows the header of the sslfail.com website. It features the 'SSLFAIL' logo on the left, the text ':SSLFail:' in the center, and the IP address '1.2.840.113549.1.1' below it. A city skyline is visible in the background. Below the header, there are navigation links for 'Home' and 'Submit a Link', a search bar with a 'Go!' button, and a 'Categories' section. The main content area displays a blog post titled 'Rogers Webmail SSLFail Follow-up', posted by Tyler on July 21, 2009, with 'SSLFail / No Comments'.



The screenshot shows a WindowsUpdate SSL fail error message. The title is 'WindowsUpdate SSL fails', posted by romain on March 17, 2009, with 'SSL Fail Images / No Comments'. The error message itself is displayed in a red box with a yellow warning icon. The text reads: 'This is probably not the site you are looking for! You attempted to reach www.windowsupdate.com, but instead you actually reached a server identifying itself as windowsupdate.microsoft.com. This may be caused by a misconfiguration on the server or by something more serious. An attacker on your network could be trying to get you to visit a fake (and potentially harmful) version of www.windowsupdate.com. You should not proceed.' Below the message are buttons for 'Proceed anyway' and 'Back to safety', and a link to 'Help me understand'.



The screenshot shows a 'Temporary LinkedIn SSLFail' error message. The title is 'Temporary LinkedIn SSLFail!', posted by Tyler on July 06, 2009, with 'SSLFail / No Comments'. The text reads: 'We had an interesting screen shot sent in today from Sheldon (his post on the subject). It appears as though the SSL certificate on LinkedIn expired today and they waited until after the expiration to update their cert, leaving people with SSL errors temporarily. This doesn't seem like a great way to foster user trust, I'd prefer my sites update their certificates early, rather than wait for them to expire. If this was a matter of they forgot the date and weren't ready for the expiration, then I'm really concerned, that mistake should not happen... especially with a site I trust with so much of my personal information.' Below the text is a screenshot of a Firefox browser showing an 'Untrusted Connection' error for www.linkedin.com. The error message states: 'This Connection is Untrusted. You have asked Firefox to connect securely to www.linkedin.com, but we can't confirm that your connection is secure. Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified. What Should I Do? If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you should be cautious. Get me out of here! Technical Details: www.linkedin.com uses an invalid security certificate. The certificate expired on 7/6/2009 1:41 PM. (Error code: ssl\_error\_untrusted\_or\_expired)' Below the technical details is a link to 'I Understand the Risks'.

# Questions?

lobby -or-

mcoates@mozilla.com -or-

<http://michael-coates.blogspot.com>

