

Secure Web Overview



Sandia
National
Laboratories

Computer Security Technologies

Presented to:

1998 DOE Computer Security Training Conference

St Petersburg Florida, April 27-30, 1998

<http://doe-is.llnl.gov/ConferenceProcs.html>

Pat Moore, Sandia National Laboratories

pcmoore@sandia.gov

- Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000
- Review&Approval # SAND 981555C

This Talk Will Cover . . .



Sandia
National
Laboratories

Computer Security Technologies

- **Secure Web:** A system for providing HTTP access to UCI or Classified NTK partitioned data.
- Emphasis on what's practical today with an upgrade path to tomorrow
- Technical, but for web security neophytes
- Not covered:
 - How-to-secure details for servers or browsers
 - Vulnerabilities of Java, Active-X, CGI, etc.
 - Cryptography and PKI basics

Requirements of a Secure Web System



Sandia
National
Laboratories

Computer Security Technologies

- Authentication
- Authorization
- Integrity/Privacy
- Highly Secure Server Configuration
- Desktop/Browser Security Policy
- Plus, you may require:
 - Standards - Based, Cross Platform solution
 - Corporate Public Key Infrastructure
 - Corporate Secret Key Infrastructure

Some Definitions



Sandia
National
Laboratories

Computer Security Technologies

SSL, TLS

Secure Sockets Layer,
Transport Layer Security

HTTP, HTTPS

Hypertext Transfer Protocol,
HTTP over SSL

CGI

Common Gateway Interface

NSAPI, ISAPI

Netscape,
IIS Application Programming Interfaces

Kerberos, DCE

Secret key (password) based security
protocols

Certificate

Security credential based on public-key
signature

Some Definitions



Sandia
National
Laboratories

Computer Security Technologies

Intranet

HTTP environment restricted to company employees, contractors, and guests

Extranet

HTTP environment designed to securely serve partners and select customers in the internet

CA

Certification Authority

AC

Attribute Certificate

\$REMOTE_USER

The user that is using the browser

SSL / TLS History



Sandia
National
Laboratories

Computer Security Technologies

- Invented by *Netscape* in 1994
- Competed with IETF proposed S-HTTP
- SSL today is an IETF Standard with public reference code.
- SSL V2.0 in *Netscape Commerce Server*
 - Had an early embarrassing flaw
 - Had subtle performance and man-in-middle weaknesses
- SSL V3 in *Netscape Enterprise Server*
- TLS \cong SSL V3.1
- PCT is from *Microsoft*, similar to SSL2
 - (essentially dead, but still in the Microsoft Schannel DLL)

What SSL/TLS Gives You



Sandia
National
Laboratories

Computer Security Technologies

- Encrypted connection
 - Multiple Cipher support
 - Everything but the hostname is encrypted
- Server authentication
 - Based on server certificate
 - Certification Authority public keys are embedded in the browser
- Optional client authentication
 - Client certificate signed by CA
 - Server configured to trust CA

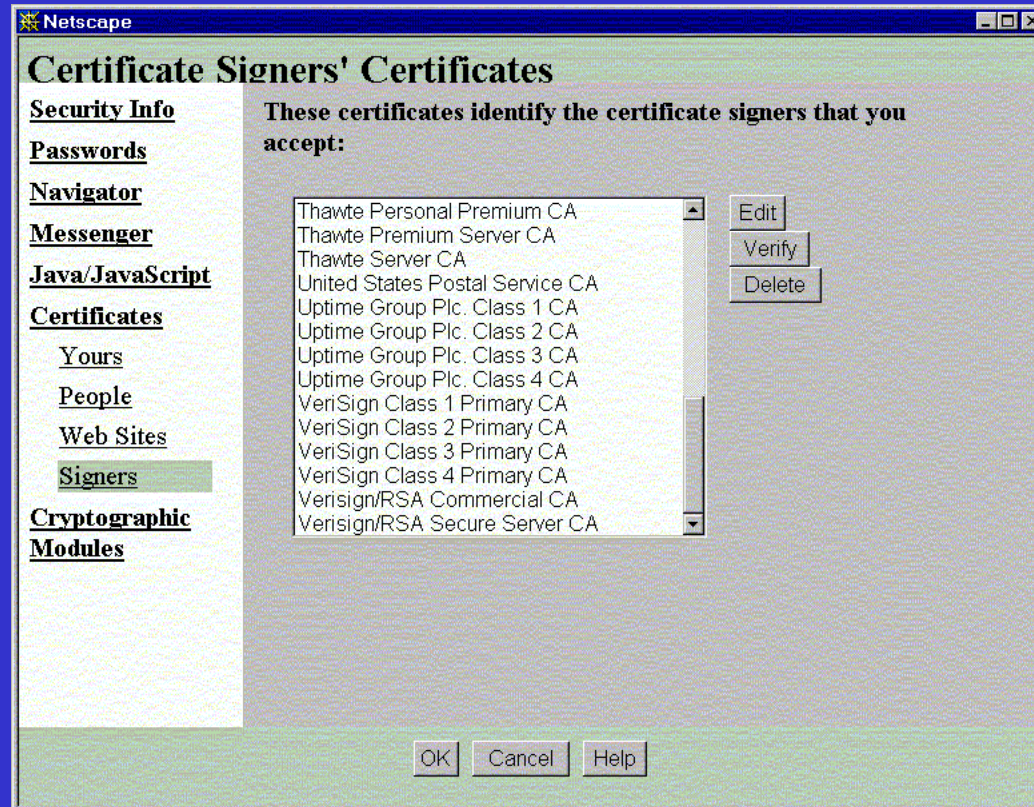
SSL Certificate Trust



Sandia
National
Laboratories

Computer Security Technologies

- *Netscape 4.04* out-of-box already trusts about 40 Certificate Authorities



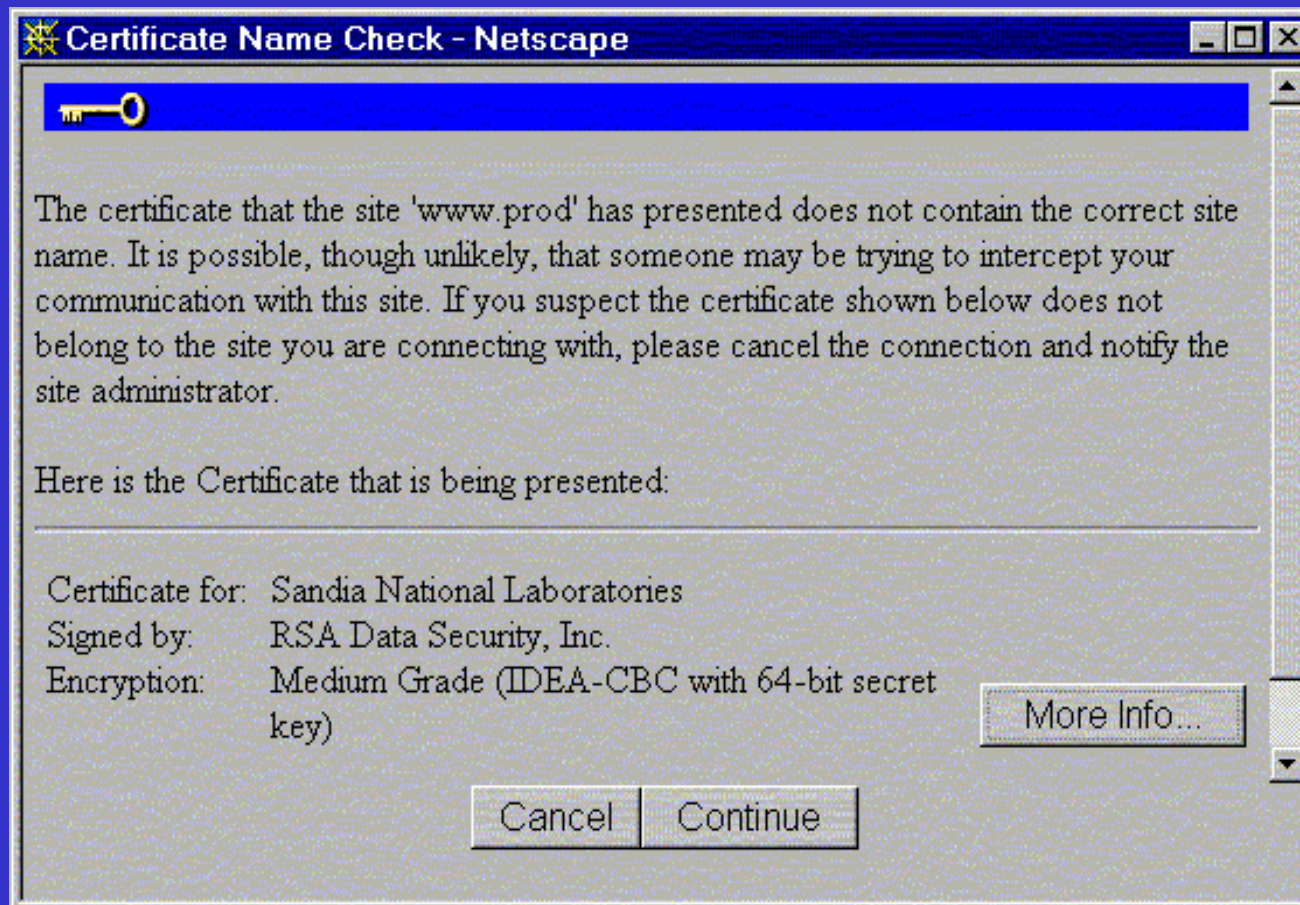
SSL Certificate Trust



Sandia
National
Laboratories

Computer Security Technologies

- Browsers warn about fishy certificates - Communicator



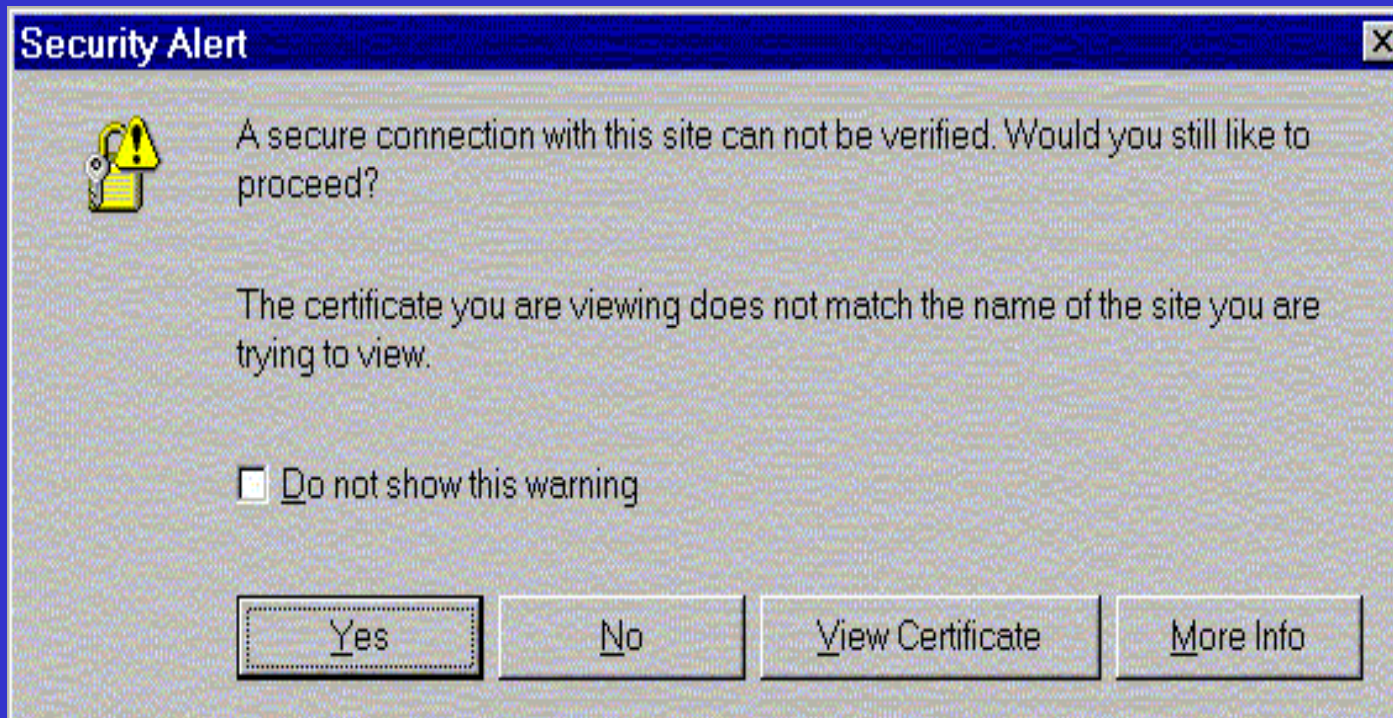
SSL Certificate Trust



Sandia
National
Laboratories

Computer Security Technologies

- Browsers warn about fishy certificates - IE4



How SSL/TLS Works (1)



Sandia
National
Laboratories

Computer Security Technologies

1. Client sends hello to server, tells server its capabilities and requirements
2. Server sends certificate to client, optionally requests client's certificate.
3. (if requested) client sends certificate to server
4. Client generates a random key, K , and sends to server encrypted in server's public key
5. (if requested) client hashes K , signs it with private key and sends it to server

How SSL/TLS Works (2)



Sandia
National
Laboratories

Computer Security Technologies

6. Server decrypts K , (optionally hashes it, verifies that it matches the client-signed hash)
7. Server selects session parameters, sends to client (Netscape chooses strongest / slowest cipher)
8. Both sides generate master cipher keys for the session based on the secret key K
9. Both sides send confirmation messages, containing encrypted hashes of the entire handshake.

SSL Authentication



Sandia
National
Laboratories

Computer Security Technologies

- Each side knows that the other has the private key associated with the certificate.
- Each side knows that the data in the certificate has not been tampered with, and that the CA has vouched for the owner of the certificate.
- May mean very little, depending on CA and class of certificate
 - Name is unique, at least within that CA

SSL Client Authentication Status



Sandia
National
Laboratories

Computer Security Technologies

- Looks like the future, for both Netscape and Microsoft
- You can do it today
 - Most just use Verisign class 1 certificates
 - Works with Entrust Web CA, Netscape CA
 - Basis of Netscape single-sign-on, FIPS 140-1 validated
- Many are still watching and waiting:
 - Enterprise PKI deployment
 - Enterprise PKI compatibility (Full Entrust)
 - Key portability / smart cards
 - Scaleable enrollment and CRL support
 - Microsoft Active Directory (NT5)

HTTP Basic Authentication



Sandia
National
Laboratories

Computer Security Technologies

- What 90%+ of today's systems are using
- Part of HTTP specification
- Familiar, and supported by all servers and browsers
- Based on password authentication
 - But - with SSL the password is encrypted in transit
- Can be integrated with password infrastructures
 - NT Domain, DCE, Kerberos, SecureId

How Basic AA Works (1)



Sandia
National
Laboratories

Computer Security Technologies

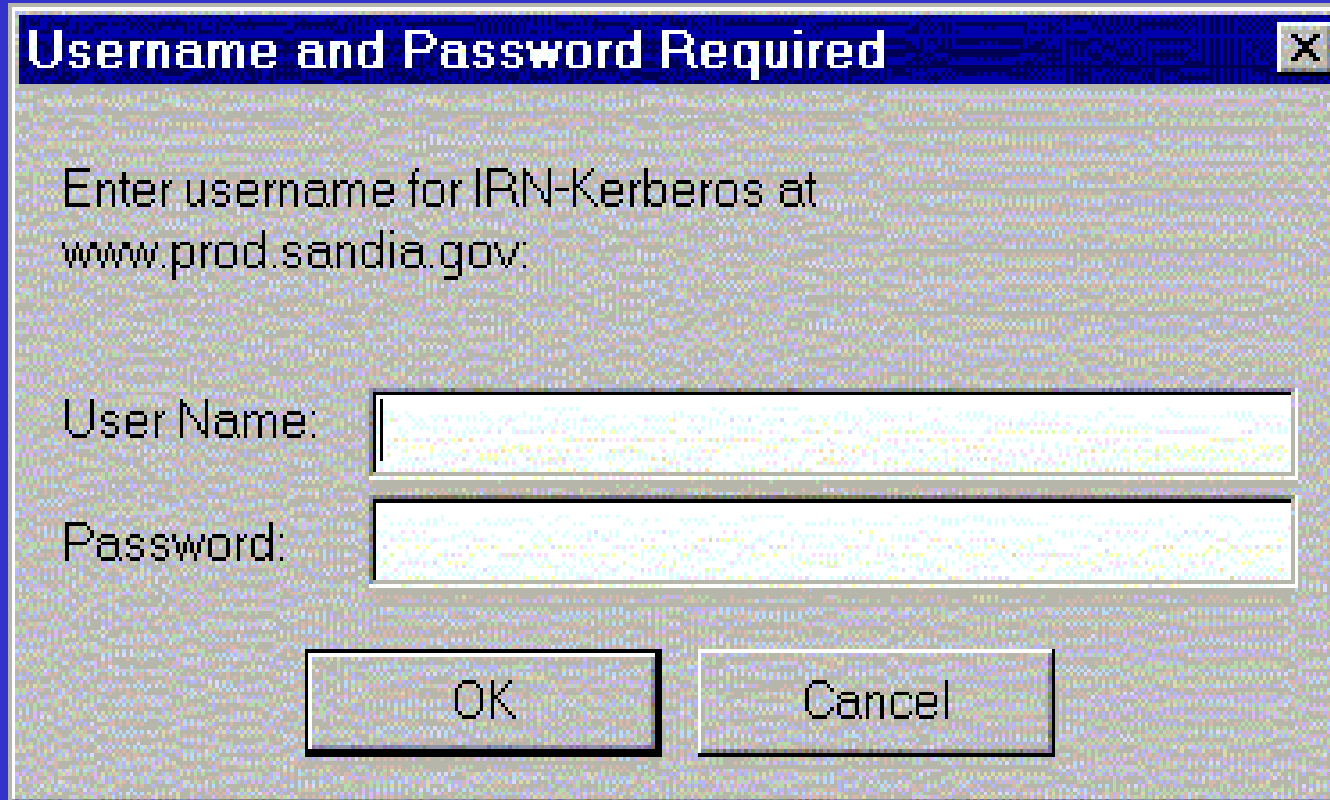
1. Browser requests a "restricted" web page
2. Server refuses to serve page without Basic AA credentials, so replies with an HTTP standard header:

401 Unauthorized
Basic

WWW-Authenticate:
realm="IRN-Kerberos"

How Basic AA Works (2)

3. Browser reacts to the 401 header with the familiar prompt:



Username and Password Required

Enter username for IRN-Kerberos at
www.prod.sandia.gov:

User Name:

Password:

OK Cancel

How Basic AA Works (3)



Sandia
National
Laboratories

Computer Security Technologies

3. User enters user name and password, clicks OK.

4. Browser repeats the request, This time with username:password uuencoded into an HTTP standard authorization header:

Authorization: Basic

"kSsdf77eK24dGSaddDAGswJD38"

How Basic AA Works (4)



Sandia
National
Laboratories

Computer Security Technologies

4. If this authorization header results in a successful request, the browser caches in memory:

(hostname/path, realm name, username:password)

- The next time the browser accesses that host, it will automatically try the cached password.

The user appears to be logged into the web server, but is actually authenticating each request with a password.

Other Authentication Methods



Sandia
National
Laboratories

Computer Security Technologies

- **Message Digest Authentication**
 - Based on standard HTTP 401 header that includes a **nonce** in addition to a realm name
 - Browser prompts for password, then sends an MD5 hash of password and nonce
 - Not in COTS browsers

- **Kerberos Ticket Authentication**
 - Also based on standard HTTP 401 header
 - Browser sends Kerberos service ticket
 - Not in COTS browsers

Other Authentication Methods



Sandia
National
Laboratories

Computer Security Technologies

- **MS Challenge Response**
 - Based on Microsoft Authentication protocol
 - Requires Microsoft server, Microsoft desktop, and Microsoft browser
- **Forms and Cookies Login**
 - Login is an "HTML forms" web application
 - User enters password into a form
 - Application returns a login credential in a cookie
 - Browsers automatically return cookies to server
 - Strength: can enforce timed logouts, easy to code
 - Weakness: non-standard, subject to vulnerabilities

Other Authentication Methods



Sandia
National
Laboratories

Computer Security Technologies

- **Proxy Authentication**
 - Based on Standard HTTP headers
 - Commonly referred to as "reverse proxying"
 - User authenticates to gateway (proxy) server
 - Proxy is trusted by back-end servers
 - Supports a single password login to multiple internal servers.
- **SET Protocol**
 - Highly specific for credit card transactions
 - Based on PKI involving customer, merchant and bank
 - Visa, MasterCard, Netscape, Microsoft supported

Other Restriction Methods



Sandia
National
Laboratories

Computer Security Technologies

- Unpublished URLs
 - Only select users are given the URL
 - Not for controlled information
 - Difficult to protect from indexers, robots, and users that publish personal bookmark pages
- IP Restriction
 - May be good enough for "official use only"
 - Assumes anti-spoof routers at critical perimeters



No Single Mechanism

- NSAPI PathCheck
- Authorization callout in application code
- Integrated database login
- File system access controls
- DCE-Web technologies
- Attribute Certificates

Authorization - PathCheck



Sandia
National
Laboratories

Computer Security Technologies

- Netscape-specific
- Managed in server configuration file
- Compatible with Netscape authentication and authorization
- Typical use:

```
<Object name = "asci-tree">  
PathCheck fn=require-auth      auth-type=basic  
PathCheck  fn=dce-groups auth-groups="asci, hpss"  
</Object>
```

Authorization - PathCheck



Sandia
National
Laboratories

Computer Security Technologies

Strengths

- ACL is clearly tied to web object
- Authorization code securely loaded by server
 - Insulated from web developers
 - Insulated against hackers

Weaknesses

- Netscape only
- Difficult to develop
 - NSAPI
 - Thread safe C language DLL or shared object

Authorization - App Callout



Sandia
National
Laboratories

Computer Security Technologies

- Authorization code is called by the web application (CGI, Livewire, Netdynamics, etc)
- Typical use:

`Is_Authorized ($REMOTE_USER, param1, param2, ...)`

Authorization - App Callout



Sandia
National
Laboratories

Computer Security Technologies

Strengths

- Can be coded in any language, on any platform
- Very flexible, allows practically any authorization system.

Weaknesses

- Hard to validate and verify
- Vulnerable to flawed app development
- Vulnerable to hacks
- Can't be applied to static html documents

Authorization - App Callout



Sandia
National
Laboratories

Computer Security Technologies

Common Strategy for Secure Servers:

- Only one or two web programs are served by the server
- Data delivered by server depends on parameters or "extra path info"
- Example:
`https://www.foo/cgi-bin/TheOneProgram/
parm1/parm2/parm3`

Authorization - DB Login



Sandia
National
Laboratories

Computer Security Technologies

- **\$REMOTE_USER** is logged into a back-end database
- DB ACLS determine what the user can see or update
- COTS solution for IIS and Microsoft SQL Server, also third party plug-ins
- Assumes that every web user has a login account to the database

Authorization - DB Proxy Login



Sandia
National
Laboratories

Computer Security Technologies

- **\$REMOTE_USER** is mapped to a DB login account, then logged into a back-end database
- Mapping determines what the user can see or update
- Some COTS systems (NetDynamics) provide mechanisms for mapping the user and protecting the login credentials

Authorization - File System ACL



Sandia
National
Laboratories

Computer Security Technologies

- **\$REMOTE_USER** gets DCE or Operating System credentials, accesses HTML files or application programs based on file ACLs
- Common Implementations:
 - Microsoft IIS
 - IBM DFS Web Secure
 - Setuid HTTPD server (typically Apache)
- Weaknesses
 - Trusting an HTTPD that can obtain user network credentials
 - Availability cross-platform

Authorization - DCE-Web



Sandia
National
Laboratories

Computer Security Technologies

- Advanced Technology from Open Software Foundation
- Originally required DCE on the desktop, now support DCE login via Basic Authentication
- Available from
 - Gradient - MMC based ACL manager
 - DASCUM - Java based ACL manager
- Both support an efficient and easy to manage mapping of web document tree into DCE ACLs

Authorization - Attribute Certs



Sandia
National
Laboratories

Computer Security Technologies

- Promising Technology based on Public Key authorization

- Hypothetical example:

IF: Valid identity cert for \$REMOTE_USER

AND: AC1 lists \$REMOTE_USER as ER technician on duty

OR: AC2 lists \$REMOTE_USER as patient's physician

Then: \$REMOTE_USER may access patient record

- No Standard yet

<http://www.ietf.org/internet-drafts/draft-ietf-tls-attr-cert-00.txt>

Secure Web Recommendations



Sandia
National
Laboratories

Computer Security Technologies

- A perfectly secured server can't enforce access controls if browsers and desktops are insecure
 - Deploy a corporate browser
 - Enforce a desktop security policy
 - Use a firewall and proxy
 - Enforce a server security policy

Secure Web Recommendations



Sandia
National
Laboratories

Computer Security Technologies

- Use SSL w/ at least DES-56 Encryption
- Use Basic AA
- Validate passwords centrally
 - (DCE, Kerberos, NT, or Netscape LDAP)
- Build Enterprise-wide PKI
 - Make sure Basic AA usernames can be extracted from X509 certificates
 - Make sure CA is compatible with web servers and browsers
- Support client authentication and active content (Java, Active-X) with Corporate PKI

References



Sandia
National
Laboratories

Computer Security Technologies

- HTTP Authentication
 - <http://www.ietf.org/internet-drafts/draft-ietf-http-authentication-01.txt>
- TLS Specification
 - <http://www.ietf.org/internet-drafts/draft-ietf-tls-protocol-05.txt>
- WWW Security FAQ
 - <http://www.w3.org/Security/Faq/>
- By same author: (Lincoln D. Stein)
 - [Web Security A Step-by-Step Reference Guide](#)
- DFS Web Secure
 - <http://www.transarc.com/afs/transarc.com/public/www/Public/ProdServ/Product/DFS/SecureWeb/>
- DCE-Web
 - http://www.gradient.com/Products/NetCrusader/WebCrusader/webc_front.htm