Combating Common Web App Authentication Threats

Bruce K. Marshall, CISSP, NSA-IAM
Senior Security Consultant
bmarshall@securityps.com

SECURITY PS
STRATEGIC INFORMATION SECURITY

Key Presentation Topics

- Understanding Web App Authentication
- Managing User Authentication
- Securing Session Authentication

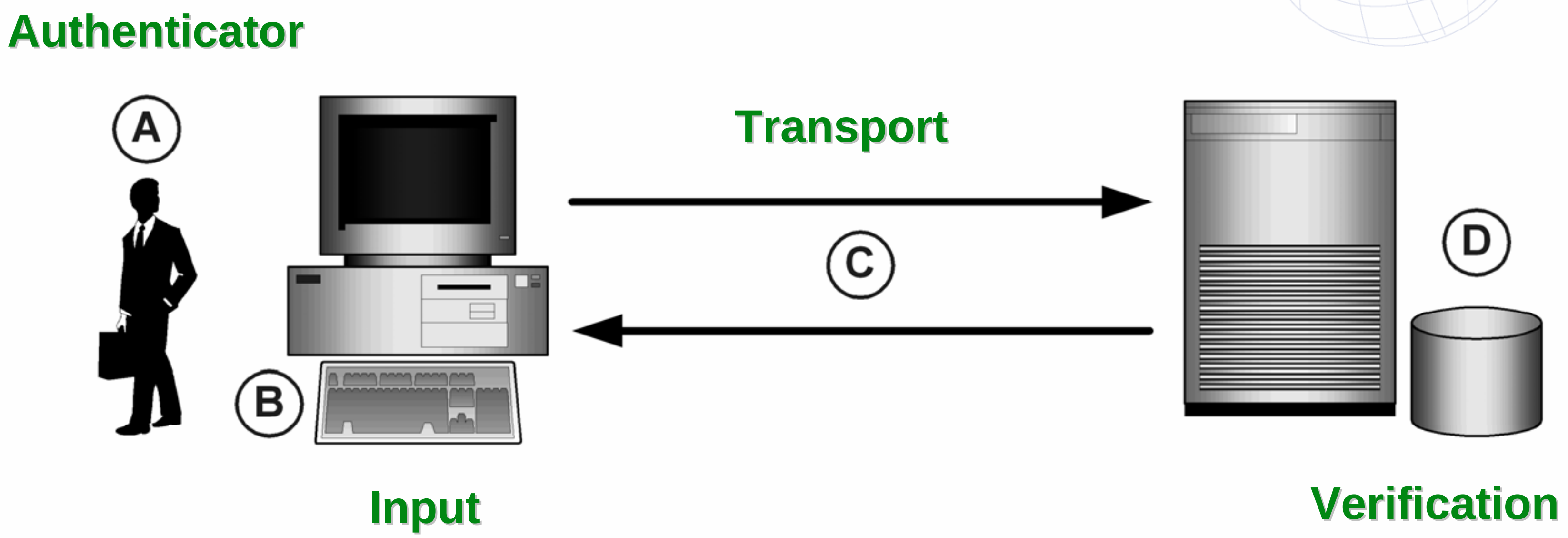
SECURITY PS
STRATEGIC INFORMATION SECURITY

Web App Authentication Challenges

- Authentication takes place with every browser-server interaction
- HTTP natively transmits data unencrypted
- Developers often fail to understand their responsibility for good authentication design
- Attackers are getting better at defeating web app authentication systems

SECURITY PS
STRATEGIC INFORMATION SECURITY

Authentication System Model



SECURITY PS
STRATEGIC INFORMATION SECURITY

Protecting Web Content

- Segment protected content from unprotected
- Authenticate users prior to granting content access
- Map only appropriate user permissions or roles to content
- Don't rely on obscurity!

Type of Web Authentication

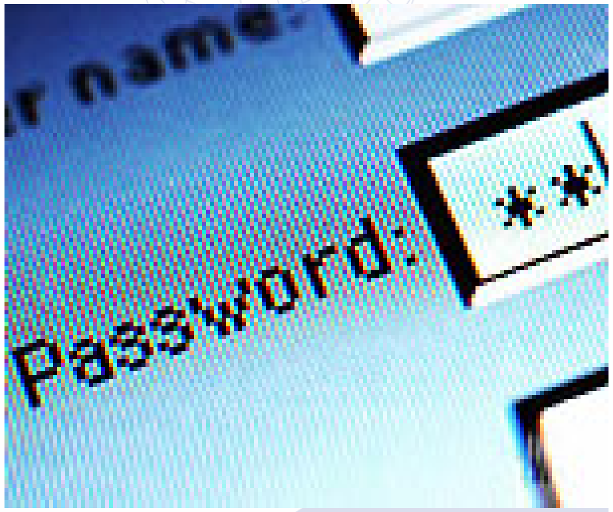
- HTTP integrated
  - Basic
  - Digest
  - NTLM / Kerberos
- Form-based
  - POST delivered parameters

Protecting Data with SSL

- Allows Web server to prove identity w/ certificate from trusted Certificate Authority
- Initiates encrypted communications between browser and Web server
- Supports multiple encryption algorithms for weak to stronger protection
- May be needed during entire Web session, and not just during authentication

User Authentication

- Normally relies on username and password
- Consider using a unique, but not meaningless, username standard
  - Not Social Security numbers
  - Not overly simple/predictable numbers or names
  - Be wary of email addresses



SECURITY PS
STRATEGIC INFORMATION SECURITY

Enforcing Good Passwords

- Don’t leave it all up to the user’s discretion
- Enforce basic requirements
  - Length
  - Character Composition
  - Name and word rejection
  - Maximum lifetime
- Start with a good and unique default
  - Require change upon first login

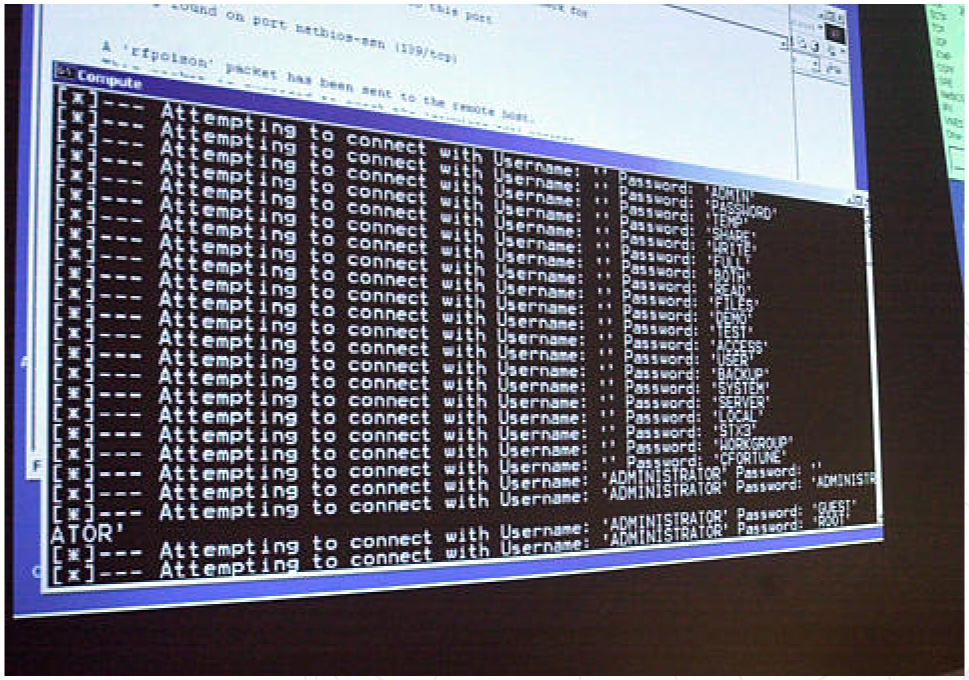
SECURITY PS
STRATEGIC INFORMATION SECURITY

Authentication Error Messages

- Prevent disclosure of username match in login failure messages
  - "*User account not found*" or "*Password incorrect*"
  - "*Error retrieving/updating the SecurityUserEntity*" or "*User ID or password entered is not valid*"

SECURITY PS
STRATEGIC INFORMATION SECURITY

Auditing Authentication Failures



```
... found on port netbios-ssn (139/tcp)

A 'rfpoison' packet has been sent to the remote host.
```

```
[x]--- Attempting to connect with Username: '' Password: 'ADMIN'
[x]--- Attempting to connect with Username: '' Password: 'PASSWORD'
[x]--- Attempting to connect with Username: '' Password: 'TEMP'
[x]--- Attempting to connect with Username: '' Password: 'SHARE'
[x]--- Attempting to connect with Username: '' Password: 'WRITE'
[x]--- Attempting to connect with Username: '' Password: 'FULL'
[x]--- Attempting to connect with Username: '' Password: 'BOTH'
[x]--- Attempting to connect with Username: '' Password: 'READ'
[x]--- Attempting to connect with Username: '' Password: 'FILES'
[x]--- Attempting to connect with Username: '' Password: 'DEMO'
[x]--- Attempting to connect with Username: '' Password: 'TEST'
[x]--- Attempting to connect with Username: '' Password: 'ACCESS'
[x]--- Attempting to connect with Username: '' Password: 'USER'
[x]--- Attempting to connect with Username: '' Password: 'BACKUP'
[x]--- Attempting to connect with Username: '' Password: 'SYSTEM'
[x]--- Attempting to connect with Username: '' Password: 'SERVER'
[x]--- Attempting to connect with Username: '' Password: 'LOCAL'
[x]--- Attempting to connect with Username: '' Password: 'STX3'
[x]--- Attempting to connect with Username: '' Password: 'WORKGROUP'
[x]--- Attempting to connect with Username: '' Password: 'CFORTUNE'
[x]--- Attempting to connect with Username: 'ADMINISTRATOR' Password: ''
[x]--- Attempting to connect with Username: 'ADMINISTRATOR' Password: 'ADMINISTR
ATOR'
[x]--- Attempting to connect with Username: 'ADMINISTRATOR' Password: 'GUEST'
[x]--- Attempting to connect with Username: 'ADMINISTRATOR' Password: 'ROOT'
[x]--- Attempting
```

- Log all successful and failed logins
- Alert staff when failed logins hit threshold
- Consider IP block or account lockout
- Notify user of last successful login and unsuccessful attempts

SECURITY PS
STRATEGIC INFORMATION SECURITY

Forgotten Passwords

- Human-centered systems for dealing with forgotten passwords are more costly
- Automated systems pose security challenges
  - View password hint
  - Provide answer to secondary secret
  - Provide answers to pre-selected questions
  - Email existing/new password to user
- Consider forcing logoff after password change

Password Storage

- Password database should be well protected
- Obfuscate stored passwords using a one-way cryptographic hash function
- Seed hash function for greater security

SECURITY PS
STRATEGIC INFORMATION SECURITY

Alternative Authenticators

- Make sure that alternative means of authenticating are appropriately secure
  - Order numbers
  - Phone numbers
- Consider stronger authentication factors
  - Hardware tokens
  - Software tokens
  - Client-side certificates
  - Biometrics

SECURITY PS
STRATEGIC INFORMATION SECURITY

Session IDs

- Identify the user to the Web application with a temporary ID
- Usually stored and transmitted as a “cookie”
- Can be stored in the URL
- Assigned either after or prior to user authentication
- As valuable as a password

SECURITY PS
STRATEGIC INFORMATION SECURITY

Cookie Assignment & Use

```
GET http://www.shopapp.com/ HTTP/1.1
Host: www.shopapp.com
```

```
HTTP/1.1 200 OK
Date: Wed, 03 Aug 2005 19:55:04 GMT
Set-Cookie: FPB=dc1hj7k1g11f288p;
    expires=Thu, 01-Jun-2006
    19:00:00 GMT; path=/;
    domain=www.shopapp.com
Connection: close
```

```
GET http:// www.shopapp.com/home.asp HTTP/1.1
Host: www.shopapp.com
Cookie: FPB=dc1hj7k1g11f288p
```

Critical Factors for Strong Sessions

- Privacy
  Must be difficult to capture
- Predictability
  Must be difficult to predict
- Key Space
  Must be difficult to brute force
- Time Window
  Must be valid for limited time only

SECURITY PS
STRATEGIC INFORMATION SECURITY

Requirement 1: Privacy

Lack of SID Privacy Leads to Session Theft:

1. Obtain a valid session ID from another user’s session
2. Substitute session ID and assume victim’s

Session ID Privacy Tips:

- Use SSL
- Use cookie flags *(e.g. secure, path, non-persistent)*
- Pass ID securely *(e.g. Not in URL)*

Requirement 2: Very Low Predictability

Predicting A Session ID:

- Gather a number of cookies
- Find pattern; predict existing or future IDs
- Use predictions to steal user sessions

```
sessionID=49PAKD43301356F
sessionID=49PAKD43301357F
sessionID=49PAKD43301358F
sessionID=49PAKD43301359F
sessionID=49PAKD43301360F
sessionID=49PAKD43301361F
```

Example: Single increment pattern. Simple to predict.

Session IDs – Key Space

Requirement 3: Large Key Space

Brute Forcing a Session ID:

- Gather a number of cookies
- Find any pattern to reduce “key space”
- Use a script to generate and test cookies

```
sessionID=49AKD494958F
sessionID=49AKD483492F
sessionID=49AKD459304F
sessionID=50AKD431333F
sessionID=50AKD412983F
sessionID=50AKD463340F
```

Example: ID with constant, pattern, and randomized values. Brute force-able.

SECURITY PS
STRATEGIC INFORMATION SECURITY

Session IDs – Time Window

Requirement 4: Limited Time Window

- Limiting the time for an attacker to brute force, predict, or steal a session
- Must balance timeframe with annoyance to user
- Associate a server-side timestamp with each session ID
- Refresh timestamp each time a request associated with the session is received
- Give users a logoff button that expires session ID

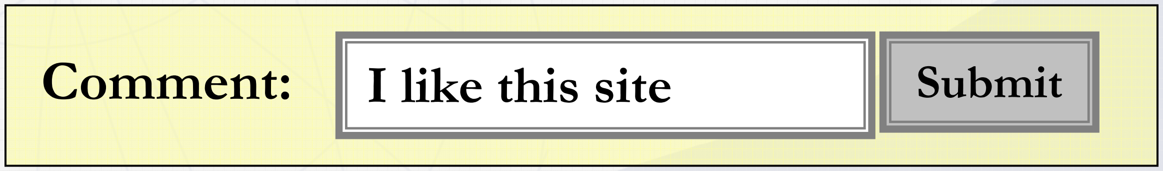
SECURITY PS
STRATEGIC INFORMATION SECURITY

Hidden Parameter Manipulation

Allow an app to access data hidden from user

```
<form action="/comment.asp" method="POST">
   Comment: <input name="comment" size=20>
   <input type="submit">
   <input type="hidden" name="userid"
value="bmarshall">
</form>
```

User Sees:



Comment: I like this site

App Sees:

- comment=I like this site
- userid=bmarshall

Parameter Injection

Expected application behavior is changed by inserting parameters into a request

Common examples:

- admin=1
- Mode=debug
- discount_code=102

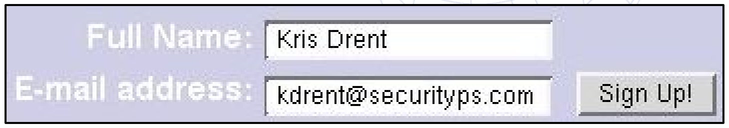
Cross-Site Scripting (XSS)

- Your application may be tricked into serving up an attacker's HTML or scripts to users
- Commonly used to steal the user's session ID
- May be used to steal username & password credentials from a form

SECURITY PS
STRATEGIC INFORMATION SECURITY

Normal Application Operation:

Step 1: User submits input

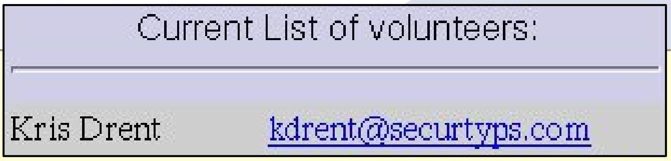


Full Name:	Kris Drent
E-mail address:	kdrent@securityps.com

Step 2: Application processes input, stores values:

Name = Kris Drent
Email = kdrent@securityps.com

Steps 3-4: Application retrieves values from database and places them on HTML page:

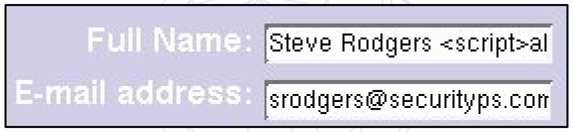


Current List of volunteers:

Kris Drent	kdrent@securtyps.com

```
<td>
  Kris Drent
</td>
<td>
  <a href="mailto:kdrent@securityps.com">kdrent@securityps.com</a>
<td>
```

Exploited Operation:


Full Name: Steve Rodgers <script>al

E-mail address: srodgers@securityps.com

Phase 1: User submits name with unexpected HTML tags:

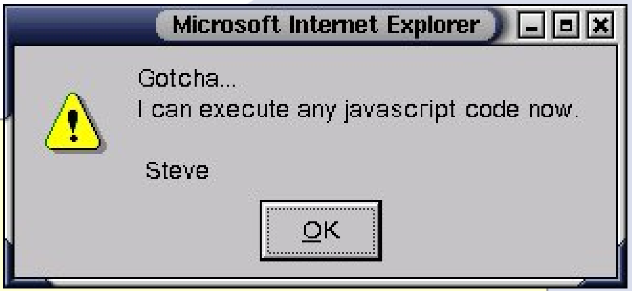
Steve Rodgers<script>alert(“Gotcha…”)</script>

Phase 2: Application processes input, stores values:

name= Steve Rodgers<script>alert(“Gotcha…”)</script>
email= srodgers@securityps.com

Phase 3-4: Application retrieves values from database and places them on HTML page:

```
<td>
  Steve Rodgers<script> alert(“Gotcha…”) </script>
</td>
<td>
  <a href=“mailto:srodgers@securityps.com”>srodgers@securityps.com</a>
<td>
```



Gotcha...
I can execute any javascript code now.

Steve

Cross-Site Scripting Solutions

- Perform data validation
  - Inspect all input for expected characters and formatting
  - Prepare all output for proper encoding
  - Build this into global app data validation library for regular reuse

SECURITY PS
STRATEGIC INFORMATION SECURITY

Phishing

- Act of tricking users into sending their login credentials or other info to attacker
- Must focus on user to hinder
  - Educate about communication policies
  - Stick with communication policies
  - Authenticate the organization to the user
- Make phishing easy to report

SECURITY PS
STRATEGIC INFORMATION SECURITY

Summary & Call to Action

- Take initiative to implement strong user authentication now
- Investigate how web apps handle session ID generation and management
- Validate input to prevent XSS and SQL injection
- Visit www.passwordresearch.com

Questions?

SECURITY PS
STRATEGIC INFORMATION SECURITY