

Securing Your Site

- Introduction
- Operational security
- Design Security In
- Security in Depth
- Apache Security
- Webgui Security
- PKI Authentication

Introduction (1)

- Why is web security important
- What are you trying to protect
- Where is the threat
 - Insider threat
 - Hackers and script kiddies
 - Zero day attack
 - Phishing and social engineering

Introduction (2)

- What is possible
 - Depends on the environment
 - Management support critical
- What are the problems
 - Budgets and Time
 - Responsibility without authority
 - Unsecure attitudes

Operational Security

- Physically secure servers and firewalls
- Isolate server networks
- Use Network Address Translation (NAT)
- Limit login accounts on servers
- Audit everything important
- If it important enough, do it yourself

Design Security In (1)

- Design first, implement later
- Don't know what you are doing?
 - Hire people who do
 - Consider outsourcing the whole effort
- Security isn't transitive
- Keep some expertise in house
- Make security a part of contracts

Design Security In (2)

- Plan to pay what security costs
- People are assets/liabilities
- Plan for things to break

Security in Depth (1)

- You can't patch your way to security
 - Or obscure your way to it either
- The Doctrine of Least Privilege
- Keep users off the server
- Keep Stuff off the Internet (really)
- Firewalls are your friend

Security in Depth (2)

- Prototype and test
 - And test and test
- Harden your servers
- Control services
- Separate security responsibilities

Apache Security

- Use what Apache provides
- SSL for everything
- Use port based virtual hosts
- Logging
 - Otherwise, you don't know what happened
 - Protect your logs
- Use tripwire for critical servers

Webgui Security

- The startup parameters
- The web interface
- Users
 - Minimize the number of administrators
- Groups and permissions
 - Your best tool
- Page controls make it easy

Startup Parameters (1)

- WebGUI.conf
 - cookieTTL -> Set to “session”
 - passthruUrls -> Better handled separately
 - webserverPort -> SSL pages on port 443
 - adminModeSubnets -> Limit admin privileges
 - spectreSubnets -> Use actual IP address?
 - runOnLogin -> Display last login

Startup Parameters (2)

- WebGUI.conf (continued)
 - authMethods -> add a new authentication
- Spectre.conf
 - webguiPort -> Change to 8008

PKI Authentication

- Certificates
- Hijacking the Request Cycle
- Getting certificate information
- A PKI authentication module

Certificates (1)

- Cryptographic Authentication
 - Chains of trust
 - Server Certificates
 - Belongs to the server machine
 - Client Certificates
 - Belongs to the user and are portable
- Unique Identification
- Verify identity independently

Certificates (2)

HTTPS	SSL_CLIENT_A_SIG
SSL_PROTOCOL	SSL_CLIENT_A_KEY
SSL_SESSION_ID	SSL_CLIENT_CERT
SSL_CIPHER	SSL_CLIENT_CERT_CHAIN
SSL_CIPHER_EXPORT	SSL_CLIENT_VERIFY
SSL_CIPHER_USEKEYSIZE	SSL_SERVER_M_VERSION
SSL_CIPHER_ALGKEYSIZE	SSL_SERVER_M_SERIAL
SSL_VERSION_INTERFACE	SSL_SERVER_S_DN
SSL_VERSION_LIBRARY	SSL_SERVER_S_DN_x509
SSL_CLIENT_M_VERSION	SSL_SERVER_I_DN
SSL_CLIENT_M_SERIAL	SSL_SERVER_I_DN_x509
SSL_CLIENT_S_DN	SSL_SERVER_V_START
SSL_CLIENT_S_DN_x509	SSL_SERVER_V_END
SSL_CLIENT_I_DN	SSL_SERVER_A_SIG
SSL_CLIENT_I_DN_x509	SSL_SERVER_A_KEY
SSL_CLIENT_V_START	SSL_SERVER_CERT
SSL_CLIENT_V_END	

where *x509* is a component of a X.509 DN:
C,ST,L,O,OU,CN,T,I,G,S,D,UID,Email

Certificates (3)

- SSL_CLIENT_S_DN = Distinguished Name
- Common Elements of SSL_CLIENT_S_DN
 - C Country
 - O Organization
 - OU Organizational Unit
 - CN Common Name

Certificates (4)

- Variations on the DN are possible
 - More than one OU is possible
 - Other elements are possible
 - ST,L,T,I,G,S,D,UID
 - DN form is determined by policy
 - CN does not guarantee uniqueness

Certificates (5)

Getting the Distinguished Name from .pem file

```
openssl x509 -subject -noout -in <filename>
```

Output:

```
/C=US/O=Acme Corp/OU=Sales/CN=Joe.Smith
```

Hijacking the Request Cycle (1)

- The WebGUI Request Cycle
- WebGUI.pm – What it does
- PostReadRequestHandler
 - Runs BEFORE PerlInitHandler
 - Screens ALL HTTPS requests
 - Allows total control
 - Passes information to Authentication

Hijacking the Request Cycle (2)

- What you can do
 - Logging
 - Authentication
 - Account Expiration can be checked
 - Redirects and rewrites are possible

Getting Certificate Information

Cpan Module Apache::SSLLookup

```
use Apache::SSLLookup;  
my $r = Apache::SSLLookup -> new(shift);  
my $ssl_var = $r ->  
    ssl_lookup('SSL_CLIENT_S_DN');
```

Webgui PKI Authentication (1)

- At Apache Request Cycle Start
 - Get the user's DN
 - Get other useful information at the same time
 - Match the user to an authorized list
 - Lots of possible ways
 - REDIRECT if invalid user
 - Pass user information using pnotes
 - Return `Apache2::Const::DECLINED`

Webgui PKI Authentication (2)

- After Webgui Session Initialization
 - Authenticate user at session start
 - Automagically log the user in

Getting other information

Getting the uri requested by the user:

```
use Apache2::RequestReq;  
my $uri = $r->unparsed_uri();
```

Getting the remote IP address:

```
use Apache2::Connection;  
my $c = $r->connection;  
my $remote_ip = $c->remote_ip();
```

Matching the user

- Choices:
 - Simple file matches DN returns WebGUI user name
 - Definitely simple (about 10 lines of PERL)
 - Simple database match (gdbm & friends)
 - LDAP remote match with separate database
 - May be required by company
 - Takes time and sensitive to network outage
 - WebGUI database lookup (already there)

Pass information with pnotes

Pnotes – An associative database that sticks with the Apache session. Unique to each request.

```
use Apache2::RequestUtil;  
$r->pnotes("user_identity" => somestring);
```

Webgui PKI Authentication (3)

- Default Authentication Replacement
- No Passwords required from the user
- Requires a certificate for each user
- Requires changes in WebGUI configuration files

End of the Presentation

Questions?

Yes...the code is available

John Nolan

jnolan@super.org