Single sign-on websites with Apache httpd: Integrating with Active Directory for authentication and authorization

Michael Heldebrant
Solutions Architect, Red Hat

Outline

Authentication overview
- Basic
- LDAP
- Kerberos

Host based Authorization
User based Authorization

Configure Kerberos by Integrating with Active Directory
Combining Kerberos and LDAP for Single Sign On
Authentication and LDAP Authorization

Why use httpd for security?

CVE lists 5014 vulnerabilities for a search of php and 248 for apache httpd - (Common Vulnerabilities and Exposures 3/4/14)

A php, perl, python, etc based application can get access to the web server authenticated user by environment variables:

- Mediawiki - Extension:AutomaticREMOTE USER
- Drupal – webserver_auth module
- Cacti
- Nagios

Also:
Subversion repositories
Git repositories

Authentication

Verify that the user is who they say they are. Usually a username and password.

Require Directive - authentication in httpd

For example, any authenticated user is

    Require valid-user

or specific user

    Require user mheldeb

What is basic authentication

Username and Password in a flat file (like /etc/passwd and /etc/shadow)

Authentication: Users send a username and a password to the server. The server then hashes the password and checks for a match in the flat file

Authorization: Groups of users can be specified in a flat file (like /etc/group)

Basic authentication

Server controls the user and password list
Self contained
Does not scale for large numbers of users
Can easily get out of sync with user changes
Users can't change their passwords easily

Basic Requirements

htpasswd

utility to manage users and passwords

Configure the server to use Basic

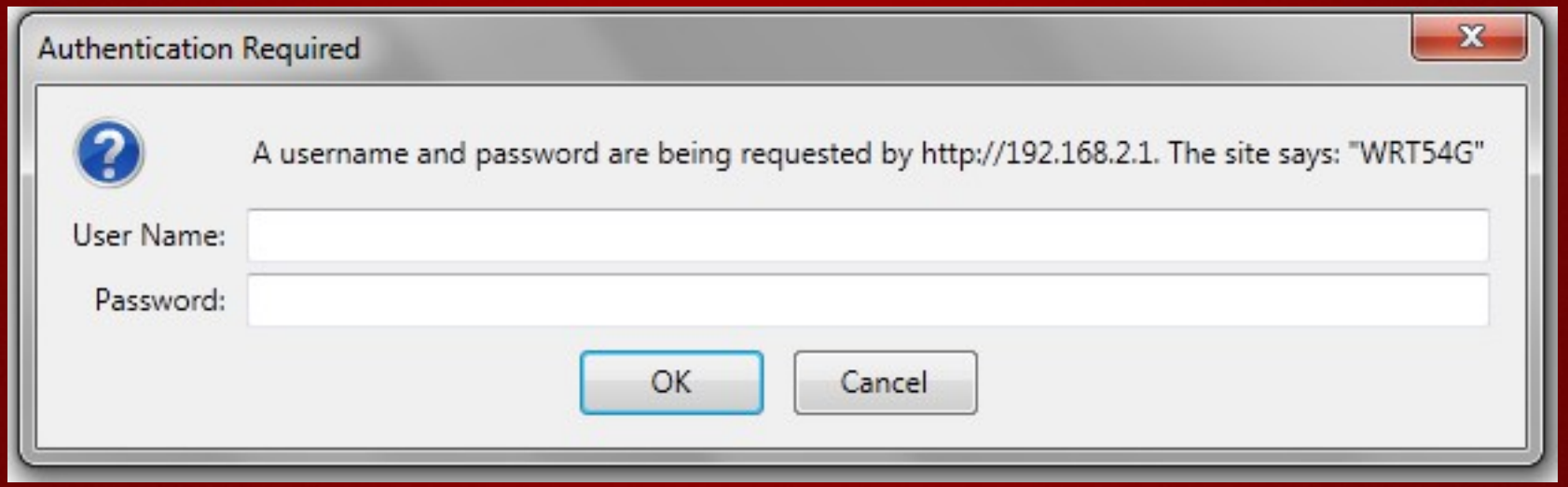
```
AuthType Basic
AuthBasicProvider File
AuthUserFile /path/to/file
AuthName "You choose"
```

You can use unique AuthNames within the same server.

What does a client see with Basic

Navigate to the secured URL

Popup asking for the Username and Password from the browser with the configured AuthName



What is LDAP

Lightweight Directory Access Protocol - a subset of x.500

Red Hat Directory Server and Active Directory are LDAP servers

Authentication: User attempts to bind to the LDAP server with their Distinguished Name (DN) and password

Authorization: memberOf attribute provides group membership in user object

LDAP authentication

LDAP is a DIT of users and passwords, groups, etc

Users must still provide a name and password for each authentication

Users can change their own passwords using existing functionality (Windows password change or passwd on a unix server configured for LDAP)

If you're not using https, passwords are going over the network in plain text

LDAP requirements

LDAP server	ldaps://ad.your.com
Where in the DIT to look for users	dc=your,dc=com
Do you need to bind as a search user?	Depends on your organization
Why would I need a bind user to authenticate users?	If username conversion to the dn requires a search (username dn could be CN="First Last",OU=Domain Users,DC=your,DC=com)

LDAP requirements continued

SSL or TLS must be used or passwords will be disclosed

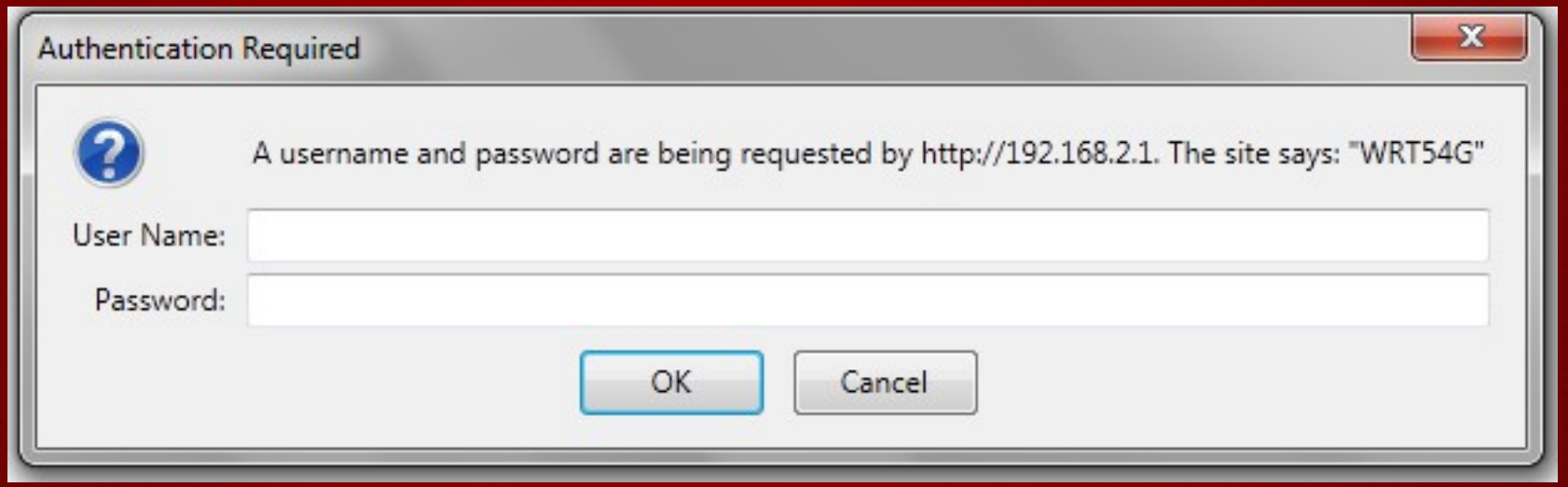
Configure CA certificate as a trusted certificate authority:

Use ldaps and set LDAPTrustedGlobalCert CA_BASE64 /etc/ssl/certs/yourca.pem

What does a client see with LDAP

Navigate to the secured URL

Same Popup as Basic asking for the Username and Password from the browser with the configured AuthName



What is Kerberos

Developed at MIT, also a Heimdal implementation

Designed for secure authentication over insecure networks:
the user password is never sent over the network

Web Browsers leverage SPNEGO -> GSSAPI (Kerberos)

Simple and **P**rotected GSSAPI **Nego**tiation Mechanism
Generic Security Services Application Program Interface
The dominant GSSAPI mechanism implementation in use is Kerberos

krb5-server and Active Directory are AS and KDC servers
Kerberos is not a way to find group membership alone

Kerberos (simplified)

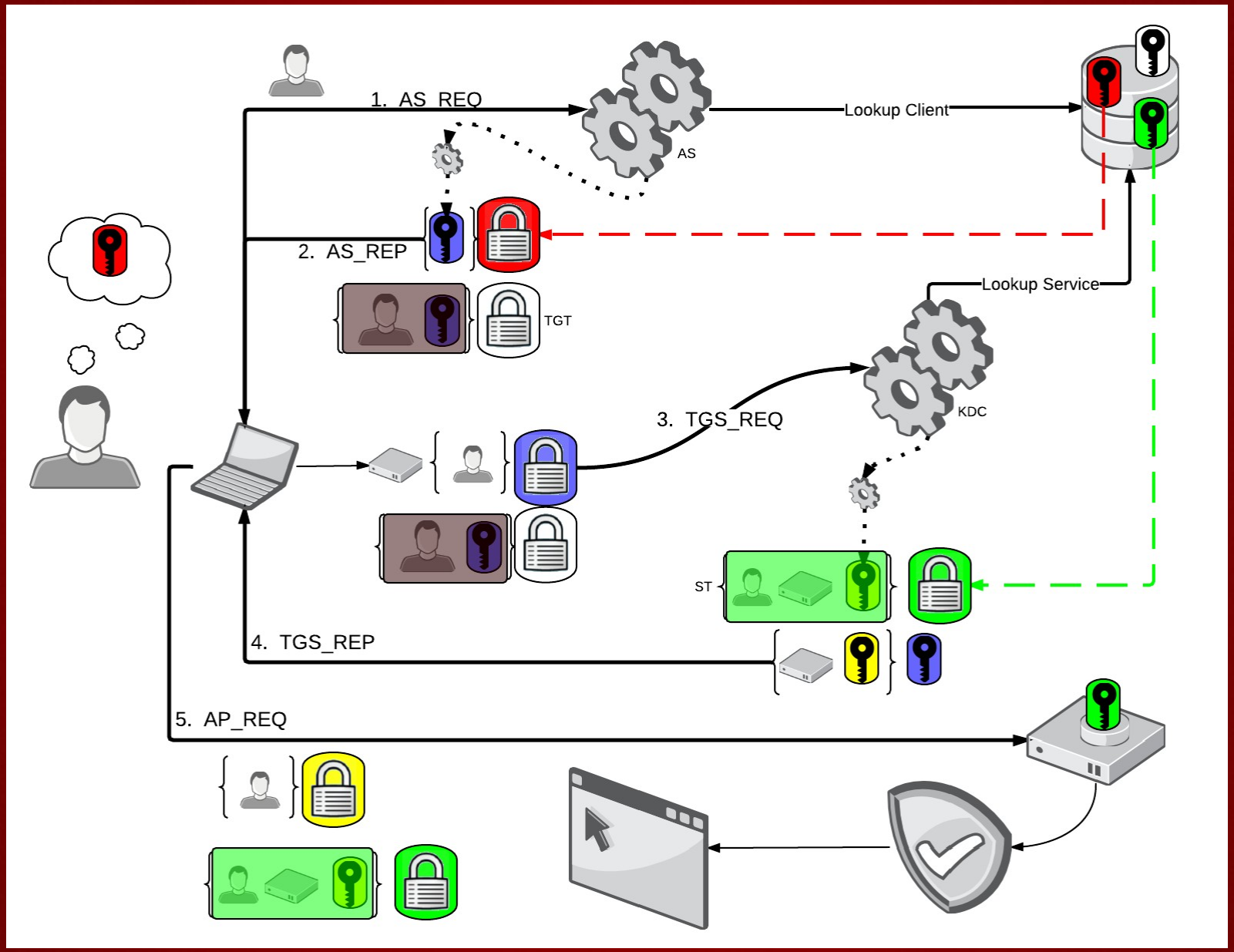
User gets a Ticket Granting Ticket (TGT) from the KDC, this happens at login/screen unlock for a Windows AD user

User wants to access a service, the client application gets a service ticket from the KDC using the cached TGT

The service ticket is presented to the service from the client application

If the service ticket is valid the user is authenticated

Kerberos (less simplified)



Single Sign On

Kerberos can provide Single Sign On to securely authenticate a user to the web server even over http alone

Using Active Directory it is possible to have desktop users login/unlock a screen and never see a password popup for authentication.

Kerberos provides no encryption of content, SSL/TLS is needed to protect data in transit or if you allow a fallback to password popup.

What does a client see with SSO

Navigate to the secured URL

Website as expected (they get authenticated to the website transparently)

Outline

Authentication overview
- Basic
- LDAP
- Kerberos

Host based Authorization
User based Authorization

Configure Kerberos by Integrating with Active Directory
Combining Kerberos and LDAP for Single Sign On
Authentication and LDAP Authorization

Authorization by host

Allow/Deny Directive – authorization in httpd

Works on:

- **ip address**
- **network**
- **dns names or matching domain**
- **apache environment variables (not covered)**

Order Directive – modify authorization behavior

Authorization continued

Order matters for security:

Allow,Deny

First, all Allow directives are evaluated; at least one must match, or the request is rejected. Next, all Deny directives are evaluated. If any matches, the request is rejected. Last, any requests which do not match an Allow or a Deny directive are denied by default.

Deny,Allow

First, all Deny directives are evaluated; if any match, the request is denied **unless** it also matches an Allow directive. Any requests which do not match any Allow or Deny directives are permitted.

Authorization continued

Satisfy Directive – combine authentication and authorization

Any or All are the options

- host **or** user
- host **and** user

Example - users must authenticate to commit to subversion, but allow read only access to a compile server for checkouts

Subversion example

```
<Location>
...
  Order Allow,Deny
  <LimitExcept OPTIONS PROPFIND GET REPORT>
  Require valid-user
  </LimitExcept>
  Require valid-user
  Allow from 192.168.1.100
  Satisfy Any
</Location>
```

Use LimitExcept as it matches anything NOT listed

Authorization by user

Should any authenticated user have access?

For example:
Access to payroll information
Administrative users to admin pages

Authorization examples:

- Usernames
- Group files local to the server
- LDAP

Authorization examples

Require valid-user	Any authenticated user can get access, including any fake user account in active directory, guest etc
Require user username	Limit to a specific user
Require ldap-group group	Limit access to a particular department via group

Authorization gotchas

Multiple Require directives act as OR

Allow directives are AND with Require unless you put Satisfy Any

LDAP Authentication
LDAP Authorization

LDAP for both phases allows users to bind to authenticate and authorize

ldap-group - mailing lists/groups in memberOf attributes of user object

ldap-attribute - department or any attribute can be used

ldap-user - specific DN

ldap-filter - complex LDAP filter

LDAP config example

Directory or Location:
AuthName "Windows Login"
AuthType Basic
AuthBasicProvider ldap
AuthLDAPBindDN bindaccount@YOUR.COM
AuthLDAPBindPassword passwordsecret
AuthLDAPUrl "ldaps://ad.your.com/dc=your,dc=com?userPrincipalName
AuthzLDAPAuthoritative on
Require valid-user

Global:
LDAPTrustedGlobalCert CA_BASE64 /etc/ssl/certs/yourca.pem

Outline

Authentication overview
- Basic
- LDAP
- Kerberos

Host based Authorization
User based Authorization

Configure Kerberos by Integrating with Active Directory
Combining Kerberos and LDAP for Single Sign On
Authentication and LDAP Authorization

Kerberos steps

NTP - time must be within 5 minutes

Server configured as kerberos client

Get a machine account and allow an HTTP service principal

Use samba to join the AD domain and use external keytabs

Add an HTTP service principal to the machine account

Configure apache to use mod_auth_kerberos with keytab

Kerberos configuration

With SRV records in DNS trivial to configure Linux as a kerberos client

/etc/krb5.conf

Configure:
default realm for your organization
mapping from dns to realm

Lookup the kdc and admin servers by SRV records in DNS

/etc/krb5.conf

```
[libdefaults]
 default_realm = YOUR.COM
 dns_lookup_realm = true
 dns_lookup_kdc = true

[domain_realm]
 .your.com = YOUR.COM
 your.com = YOUR.COM
```

testing with kinit/klist

```
[mheldeb@server ~]$ kinit
Password for mheldeb@YOUR.COM:
[mheldeb@server ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_5386
Default principal: mheldeb@YOUR.COM

Valid starting     Expires            Service principal
03/04/14 21:26:27  03/04/14 07:26:30
krbtgt/YOUR.COM@YOUR.COM
        renew until 03/04/14 21:26:27


Kerberos 4 ticket cache: /tmp/tkt5386
klist: You have no tickets cached
```

smb.conf (RHEL6 samba)

```
[global]
        workgroup = YOUR.COM
        realm = YOUR.COM
        security = ADS
        passdb backend = tdbsam
        kerberos method  = dedicated keytab
        dedicated keytab file = /etc/krb5.keytab
```

Integrating with Active Directory

"Easy" way for Unix admin

AD admin creates a machine account for the server with full control for your AD credentials

Add HTTP/servername to the machine account

Join the domain, add the HTTP SP

Say Thank You!

Windows client:
setspn.exe -S HTTP/servername servername

httpd Server:
net ads join -U youradname
net ads keytab add HTTP

Integrating with Active Directory 2

"Easy" way for the untrusting AD administrator

ktpass to map a fake user account to a single SP
    can only get one service per mapped account

import that into /etc/krb5.keytab using ktutil
    read_kt imported-file
    write_kt /etc/krb5.keytab

keytab example (samba)

```
[root@server ~]# ktutil
ktutil:  rkt /etc/krb5.keytab
ktutil:  l
slot KVNO Principal
---- ---- ---------------------------------------------------------------------
   1    3         HTTP/server.your.com@YOUR.COM
   2    3         HTTP/server.your.com@YOUR.COM
   3    3         HTTP/server.your.com@YOUR.COM
   4    3                   HTTP/SERVER@YOUR.COM
   5    3                   HTTP/SERVER@YOUR.COM
   6    3                   HTTP/SERVER@YOUR.COM
```

Kerberos requirements

mod_auth_kerb

In RHEL

your-web-site.conf

```
AuthType Kerberos
AuthName "Windows Login"
Krb5Keytab /etc/krb5.keytab
KrbAuthRealms YOUR.COM
KrbMethodNegotiate on
KrbVerifyKDC on
#KrbMethodK5Passwd off
```

Common problems: (check httpd error_log)

Service principal not in keytab	Does it actually have an HTTP service principal in there - user ktutil to check
Can't read keytab	Permissions on the file might be root only for read, chgrp to httpd group and chmod g+r
preauthentication failed	Wrong password from a user
time out of sync	Clock skew greater than 5 minutes. Use NTP

Browser configuration

Internet Explorer

- The site must be Local Intranet or Trusted Site
- Normally in AD this is the default

Firefox

In about:config

add the server to network.negotiate-auth.trusted-uris

SPNEGO can use NTLM as a mechanism so it is usually disabled over http by default for security

Kerberos Authentication
LDAP Authorization

The kerberos module will present the authenticated username as username@REALM to other modules in apache httpd

We can configure the LDAP URL to use the userprincipalname attribute in AD to find the DN of the user which should match the Kerberos username for an authenticated user.

This allows the authn module to pass an SSO authenticated user to the LDAP authz module to control access

Kerberos and LDAP requirements

Kerberos	Set up everything but the require valid-user
LDAP	Set up everything as for LDAP except the AuthType

Full config example

```
AuthType Kerberos
AuthName "Windows Login"
Krb5Keytab /etc/krb5.keytab
KrbAuthRealms YOUR.COM
KrbMethodNegotiate on
KrbVerifyKDC on
#KrbMethodK5Passwd off
AuthLDAPBindDN bindaccount@YOUR.COM
AuthLDAPBindPassword passwordsecret
AuthLDAPUrl "ldaps://ad.your.com/dc=your,dc=com?userPrincipalName
AuthzLDAPAuthoritative off
Require ldap-attribute department="Admins"
```

Virtual hosts and Kerberos

You only need a service principal for the actual dns name for the ip address of the webserver

Kerberos uses virtual server name -> ip address -> reverse lookup to get service principal

You must have a PTR record for the A address even without virtual hosts

If you have a service IP address dns name that does not match the machine account name you must have the AD admin allow the machine account to have non matching Service Principals

What else can I do with an AD bound Unix server?

ssh – uses host SPN

samba – uses host SPN

imap – uses imap SPN

NFSv4 – uses nfs SPN

LDAP – uses ldap SPN

Questions?