Access Policy Manager
Smart Card Auth Getting Started
F5’s Remote/Application Access Solutions

AUTHENTICATION, AUTHORIZATION, REMOTE ACCESS AND SSO TO ALL APPLICATIONS USING ACCESS POLICY MANAGER (APM)
F5 Access Policy Manager (APM)

- Transform one type of authentication into another so an app may understand and use it without installing additional agents
- Allow flexible selection of SSO technique appropriate to the application
- Allow for centralized session control of all applications, even SaaS
Working Environment

- Windows 10 Workstation
- BIG-IP VE 15.1.0.2
  - Licensed and provisioned with LTM and APM
- LAMP Web Server
Enable Smart Card Authentication in BIG-IP APM

1. Generate CA Bundle(s)
2. Client SSL Profile
3. Create Pool
4. Virtual Server
5. Verify Virtual Server works
6. Access Profile
7. Verify Smart Card authentication works
8. Customize message boxes
Generate CA Bundles and Import into BIG-IP

C:\Users\grabe\Desktop\CAfiles\RootCAs>type *.cer > DODRootCAs.pem

System » Certificate Management: Traffic Certificate Management: SSL Certificate List » Import SSL Certificates and Keys

<table>
<thead>
<tr>
<th>SSL Certificate/Key Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import Type</td>
</tr>
<tr>
<td>Certificate Name</td>
</tr>
<tr>
<td>Certificate Source</td>
</tr>
<tr>
<td>Free Space on Disk</td>
</tr>
</tbody>
</table>

Cancel | Import
Create CA Bundle using CA Bundle Manager
Create ClientSSL Profile
Create Pool

Configuration:
- Name: F5DODUG-Pool
- Health Monitors: Common, https

Resources:
- Load Balancing Method: Round Robin
- Priority Group Activation: Disabled
- New Members:
  - Address: 192.168.10.39
  - Service Port: 443
  - Protocol: HTTPS
Create Virtual Server

Local Traffic » Virtual Servers: Virtual Server

General Properties
- Name: F5DODUG-VS
- Description: 
- Type: Standard
- Source Address: Host Address
- Destination Address/Mask: Host Address 192.168.10.7
- Service Port: 443

Configuration:
- Protocol: TCP
- Protocol Profile (Client): tcp
- Protocol Profile (Server): (Use Client Profile)
- HTTP Profile (Client): http
- HTTP Profile (Server): (Use Client Profile)
- HTTP Proxy Connect Profile: None

FTP Profile:
- RTSP Profile:
- SSH Proxy Profile:

SSL Profile (Client):
- Default Pool: F5DODUG-Pool
- Default Persistence Profile: None
- Fallback Persistence Profile: None

SSL Profile (Server):
- Policies:

Options:
- Cancel
- Restart
- Finished
Verify Virtual Server Works

Welcome to the Web Server

Accept: text/html, application/xhtml+xml, image/jxr, */*
Accept-Language: en-US
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Host: www.test.local
Connection: Keep-Alive

SSL/TLS Information

Certificate Cipher ECDHE-RSA-AES128-GCM-SHA256
Certificate Protocol TLSv1.2
Create OCSP Responder Server

<table>
<thead>
<tr>
<th>General Properties</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>DISA_OCSP</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>OCSP Responder</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Configuration:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://ocsp.disa.mil/">http://ocsp.disa.mil/</a></td>
</tr>
<tr>
<td><strong>Certificate Authority File</strong></td>
<td>DOD-CA-BUNDLE.crt</td>
</tr>
</tbody>
</table>

[Image of the web interface showing the OCSP Responder server setup with the specified details.]
Create Access Profile
Modify Access Policy
Apply Access Profile to Virtual Server
Verify Virtual Server Works with Access Profile

You've selected the wrong certificate. Please select the certificate that contains PIV@mil.

Click here to continue
Department of Defense
Consent Banner

You are accessing a U.S. Government (USG) Information System (IS) that is provided for USG-authorized use only. By using this IS (which includes any device attached to this IS), you consent to the following conditions:

- The USG routinely intercepts and monitors communications on this IS for purposes including, but not limited to, penetration testing, COMSEC monitoring, network operations and defense, personnel misconduct (PM), law enforcement (LE), and counterintelligence (CI) investigations.

- At any time, the USG may inspect and seize data stored on this IS.

- Communications using, or data stored on, this IS are not private, are subject to routine monitoring, interception, and search, and may be disclosed or used for any USG authorized purpose.

- This IS includes security measures (e.g., authentication and access controls) to protect USG interests—not for your personal benefit or privacy.

- Notwithstanding the above, using this IS does not constitute consent to PM, LE or CI investigative searching or monitoring of the content of privileged communications, or work product, related to personal representation or services by attorneys, psychotherapists, or clergy, and their assistants. Such communications and work product are private and confidential. See User Agreement for details.

OK, Proceed To Application
More Resources

APM Training videos
• https://devcentral.f5.com/s/articles/Free-F5-Training-Getting-Started-with-BIG-IP-APM

YouTube
• https://www.youtube.com/user/f5networksinc
• https://www.youtube.com/user/devcentral
• https://www.youtube.com/channel/UCMiRji4gfmK-PKB6CN5HmGA - F5 Government Solutions

LearnF5
• https://account.f5.com/learnf5/signin

Webinars
• https://www.f5.com/company/events/webinars

Support
• https://support.f5.com/csp/home
Appendix/Backup APM
Access Policy: /Common/F5DODUG_AccessProfile

Properties | Branch Rules

Add Branch Rule

Expression: expr {{mcget (session.logon.last.username)} != "UPN-NOT-FOUND"}

Name: PIV Extracted

Name: fallback

Properties | Branch Rules

Variable Assign

Add new entry

Assignment:

1. session.logon.last.upn = set 509e_fields [split {mcget (session.ssl.cert.x509extension)} "\n"]; # For each element in the list: for each field $x509e_fields { # If the element contains UPN: if { $field contains "othername:UPN" } { ## start of UPN variable set start [expr {string first "othername:UPN" $field} +14] } } # UPN format is <user@domain> # Return the UPN, by finding the index of opening and closing brackets, then use string range to get everything between. return [string range $field $start [expr {string first "$" $field} +14] $field $start] - 1 ]; } } # Otherwise return UPN Not Found: return "UPN-NOT-FOUND";

2. session.logon.last.username = set upn {mcget (session.logon.last.upn)}; if {[string first @ $upn] >= 0} { return [string range $upn 0 [expr { [string first @ $upn] - 1 }] ]; } else { return $upn; }

Expression: expr {mcget (session.logon.last.username)} != "UPN-NOT-FOUND"
Consent Banner Appendix
UPN Assign Appendix

session.logon.last.upn

set x509e_fields [split [mcget {session.ssl.cert.x509extension}] "\n"];
# For each element in the list:
foreach field $x509e_fields {
    # If the element contains UPN:
    if { $field contains "othername:UPN" } {
        ## set start of UPN variable
        set start [expr { [string first "othername:UPN<" $field] +14}]
        # UPN format is <user@domain>
        # Return the UPN, by finding the index of
        # opening and closing brackets, then use
        # string range to get everything between.
        return [string range $field $start [expr { [string first ">" $field $start] - 1 } ] ]; } } else { return "UPN-NOT-FOUND"; }

session.logon.last.username

set upn [mcget {session.logon.last.upn}]; if {[string first "@" $upn] >= 0} {
    return [string range $upn 0 [expr { [string first "@" $upn] - 1 } ] ]; } else { return $upn; }