Cryptography Homework 1

Required Reading

Crypto 1 Slides – Terms and Concepts

Optional Reading

A good description of the WEP problems:
http://www.dummies.com/programming/networking/understanding-wep-weaknesses/

For a heavy duty technical/mathematical explanation of WEP failures, you can read this (optional):
http://dl.aircrack-ng.org/breakingwepandwpa.pdf

Base64 notes

Base64 is used to convert binary data into a string that can be transmitted through a medium that only understands characters, like email. See this link for more information. https://en.wikipedia.org/wiki/Base64. Base64 converts a 24-bit chunk of data into four 8-bit characters, or 32 bits. Since the data and the base64 characters may not be of the same length, base64 may be padded with one or two "=" at the end. If a string ends in "=" or "==", it is almost always base64. The character set base64 uses is [a-zA-Z0-9+/], or all upper- and lower-case letters, digits 0-9, and "+" and "/". If you see an apparently random string composed of those characters, there is a good chance it is base64 (This is useful in Capture the Flag exercises.)

Base64 in Linux

The Linux app base64 takes input from the pipeline or from a file. It sends results to standard output, usually the screen. If you want the output to go to a file use redirection (> filename.) To decode a short string of base64 you can use echo and the pipeline.

```
echo -n "dGhpcyBpcyByYW5kb20gdGV4dA==" | base64 -d
```

The '-n' tells `echo` not to add a newline (\n) at the end of the string. The -d in base64 means decode.

To encode a short string use this.

```
echo -n "a short string" | base64
```

Notice that base64 encodes by default.

To decode a base64 encoded file and send the output to a file, use this.

```
base64 -d encodedFileName > aFile
```

To encode a file and send the output to a file, use this.

```
base64 someFileName > encodedFileName
```

Base64 in Windows

Certutil.exe

Many people use the Windows certutil.exe app for converting to and from base64 in Windows. Certutil.exe works with digital certificates which use base64, and you can use just the base64 components. However, certutil.exe only works with files; you can't pipe or redirect. To encode `someFileName` and put the encoded output in `encodedFileName`, do this.

```
certutil -encode someFileName encodedFileName
```

Likewise, do decode a file do this.

```
certutil -decode encodedFileName decodedFileName
```

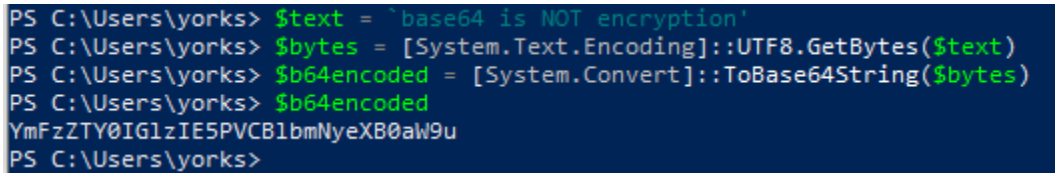
PowerShell

PowerShell does not include a cmdlet for base64. However, it can call the .Net library which does have base64 capabilities. The syntax is awkward. To encode use this.

```
$text = 'base64 is NOT encryption'
$bytes = [System.Text.Encoding]::UTF8.GetBytes($text)
$b64encoded = [System.Convert]::ToBase64String($bytes)
```

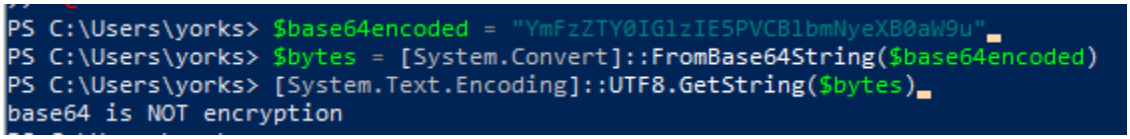
The code `[System.Text.Encoding]::UTF8.GetBytes` means go to the `System.Text.Encoding` library in .Net, and use the `UTF8.GetBytes()` method. You can see the details of the GetBytes() method here. https://docs.microsoft.com/en-us/dotnet/api/system.text.encoding.getbytes?view=netframework-4.8

The `$bytes` variable is the string we started with, but in .Net's native byte array format. Once our data is in this format, we can use `ToBase64String` from the `System.Convert` library to change the byte array to a base64 string.



To decode, use this.

```
$b64encoded = 'YmFzZTY0IGlzIE5PVCBlbmNyeXB0aW9u'
$bytes = [System.Convert]::FromBase64String($b64encoded)
[System.Text.Encoding]::UTF8.GetString($bytes)
```



This works well in scripts, but it is not ideal for command line typing unless you create and save your own function.

Note: Malware authors will often encode their malware with base64, and then execute it with powershell.exe -encodedCommand "long string of base64". In the example below, -enc is short for -EncodedCommand.

```
Host Application = powershell -noP -sta -w 1 -enc SQBGACgAJABQAFMAVgBlAHIAUwBpAG8ATgBUAGEAQgBMAGUALgBQAFMAVgBFAFIAcwBJAE8AbgAuAE0AQ
QBKAG8AcgAgAC0AZwBFACAAMwApAHsAJABHAFAARgA9AFsAUgBlAGYAXQAuAEEAUwBzAEUATQBCAGwAeQAuAEcARQBUAFQAeQBQAEUAKAAnAFMAeQBzAHQAZQBtAC4ATQBhAG4AYQBn
AGUAbQBlAG4AdAAuAEEAdQB0AG8AbQBhAHQAaQBvAG4ALgBVAHQAaQBsAHMAJwApAC4AIgBHAEUAdABGAGkARQBgAEwAZAAiACgAJwBjAGEAYwBoAGUAZABHAHIAbwB1AHAAUABvAGw
AaQBjAHkAUwBlAHQAdABpAG4AZwBzACcALAAnAE4AJwArACcAbwBuAFAAdQBiAGwAaQBjACwAUwB0AGEAdABpAGMAJwApADsASQBGACgAJABHAFAARgApAHsAJABHAFAAQwA9ACQARw
BQAEYALgBHAGUAVABWAEEAbAB1AEUAKAAkAG4AVQBsAEwAKQA7AEkAZgAoACQARwBQAEMAWwAnAFMAYwByAGkAcAB0AEIAJwArACcAbABvAGMAawBMAG8AZwBnAGkAbgBnACcAXQApA
HsAJABHAFAAQwBbACcAUwBjAHIAaQBwAHQAQgAnACsAJwBsAG8AYwBrAEwAbwBnAGcAaQBuAGcAJwBdAFsAJwBFAG4AYQBiAGwAZQBTAGMAcgBpAHAAdABCACcAKwAnAGwAbwBjAGsA
```

For Turn in

1) The file, document.jpg.b64 is a Base64 encoded file. Decode it and describe the picture. The base64 app on Linux is easiest. In Windows you can use certutil.exe or PowerShell.

2) Alice decides on a key, and then whispers the key into Bob's ear. What kind of encryption are they using, most likely? (Symmetric or asymmetric, 50/50 chance.) Suppose Eve is in the room and may be able to overhear what Alice whispers. How can Alice and Bob improve their key exchange?