Lab 3a: Hash Methods

You will be assigned a folder in the DFET Cloud. The hashcat version has a time-out, so enter the following command:

date -s "1 OCT 2015 18:00:00"

Demo: http://youtu.be/Xvbk2nSzEPk

1 Hashing

No	Description	Result

1	Using (either on your Windows desktop or on Kali):
	
	`http://asecuritysite.com/encryption/md5`
	
	Match the hash signatures with their words ("Falkirk", "Edinburgh", "Glasgow" and "Stirling").
	
	`03CF54D8CE19777B12732B8C50B3B66F`
	`D586293D554981ED611AB7B01316D2D5`
	`48E935332AADEC763F2C82CDB4601A25`
	`EE19033300A54DF2FA41DB9881B4B723`
		`03CF5:` Is it [Falkirk][Edinburgh][Glasgow][Stirling]?
		
		`D5862:` Is it [Falkirk][Edinburgh][Glasgow][Stirling]?
		
		`48E93:` Is it [Falkirk][Edinburgh][Glasgow][Stirling]?
		
		`EE190:` Is it [Falkirk][Edinburgh][Glasgow][Stirling]?

2	Repeat Part 1, but now use openssl, such as: echo -n 'Falkirk' | openssl md5	`03CF5:` Is it [Falkirk][Edinburgh][Glasgow][Stirling]? `D5862:` Is it [Falkirk][Edinburgh][Glasgow][Stirling]? `48E93:` Is it [Falkirk][Edinburgh][Glasgow][Stirling]? `EE190:` Is it [Falkirk][Edinburgh][Glasgow][Stirling]?

3	Using (either on your Windows desktop or on Kali): `http://asecuritysite.com/encryption/md5` Determine the number of hex characters in the following hash signatures.	MD5 hex chars: SHA-1 hex chars: SHA-256 hex chars: SHA-384 hex chars: SHA-512 hex chars: How does the number of hex characters relate to the length of the hash signature:

3	From your Windows desktop or Kali, for the following /etc/shadow file, determine the matching password:
	
	`bill:$apr1$waZS/8Tm$jDZmiZBct/c2hysERcZ3m1`
	`mike:$apr1$mKfrJquI$Kx0CL9krmqhCu0SHKqp5Q0`
	`fred:$apr1$Jbe/hCIb$/k3A4kjpJyC06BUUaPRKs0`
	`ian:$apr1$0GyPhsLi$jTTzW0HNS4Cl5ZEoyFLjB.`
	`jane: $1$rqOIRBBN$R2pOQH9egTTVN1Nlst2U7.`
		The passwords are password, napier, inkwell and Ankle123.
		[Hint: openssl passwd -apr1 -salt ZaZS/8TF napier]
		
		Bill's password:
		
		Mike's password:
		
		Fred's password:
		
		Ian's password:
		
		Jane's password:

4	From your Windows desktop or Kali, download the following: `http://asecuritysite.com/files02.zip`	Which file(s) have been modified:

	and the files should have the following MD5 signatures:
	
	`MD5(1.txt)= 5d41402abc4b2a76b9719d911017c592`
	`MD5(2.txt)= 69faab6268350295550de7d587bc323d`
	`MD5(3.txt)= fea0f1f6fede90bd0a925b4194deac11`
	`MD5(4.txt)= d89b56f81cd7b82856231e662429bcf2`	

5	From your Windows desktop or Kali, download the following ZIP file:
	
	`http://asecuritysite.com/letters.zip`
	
	View the Postscript files using:
	
	`http://view.samurajdata.se/`
		Outline what the letters contain:
		
		Now determine the MD5 signature for them. What can you observe from the result?

6	Select either Windows or Kali for this part:
	
	On Kali, download the following ZIP file and run the two programs, and run them in a command console:
	
	`http://asecuritysite.com/files01u.zip`
	
	Or on Windows, download the following ZIP file and run the two programs, and run them in a command console:
	
	`http://asecuritysite.com/files01.zip`
		What do the programs do?
		
		Now determine the MD5 signature for them. What can you observe from the result?

2 Hashing Cracking (MD5)

No	Description	Result
1	On Kali, next create a words file (**words**) with the words of "napier", "password" "Ankle123" and "inkwell" Using hashcat crack the following MD5 signatures (hash1):	`232DD...634C` Is it [napier][password][Ankle123][inkwell]? `5F4DC...CF99` Is it [napier][password][Ankle123][inkwell]?

	232DD5D7274E0D662F36C575A3BD634C 5F4DCC3B5AA765D61D8327DEB882CF99 6D5875265D1979BDAD1C8A8F383C5FF5 04013F78ACCFEC9B673005FC6F20698D Command used: `hashcat -m 0 hash1 words`	6D587...5FF5 Is it [napier][password][Ankle123][inkwell]? 04013...698D Is it [napier][password][Ankle123][inkwell]?
2	Using the method used in the first part of this tutorial, find crack the following for names of fruits (the fruits are all in lowercase): FE01D67A002DFA0F3AC084298142ECCD 1F3870BE274F6C49B3E31A0C6728957F 72B302BF297A228A75730123EFEF7C41 8893DC16B1B2534BAB7B03727145A2BB 889560D93572D538078CE1578567B91A	FE01D: 1F387: 72B30: 8893D: 88956:

3 Hashing Cracking (LM Hash/Windows)

All of the passwords in this section are in lowercase.

No	Description	Result
1	On Kali, and using John the Ripper, and using a word list with the names of fruits, crack the following pwdump passwords: fred:500:E79E56A8E5C6F8FEAAD3B435B51404EE:5EBE7DFA074DA8EE8AEF1FAA2BBDE876::: bert:501:10EAF413723CBB15AAD3B435B51404EE:CA8E025E9893E8CE3D2CBF847FC56814:::	Fred: Bert:
2	On Kali, and using John the Ripper, the following pwdump passwords (they are names of major Scottish cities/towns): Admin:500:629E2BA1C0338CE0AAD3B435B51404EE:9408CB400B20ABA3DFEC054D2B6EE5A1::: fred:501:33E58ABB4D723E5EE72C57EF50F76A05:4DFC4E7AA65D71FD4E06D061871C05F2::: bert:502:BC2B6A869601E4D9AAD3B435B51404EE:2D8947D98F0B09A88DC9FCD6E546A711:::	Admin: Fred: Bert:

3	On Kali, and using John the Ripper, crack the following pwdump passwords (they are the names of animals): fred:500:5A8BB08EFF0D416AAAD3B435B51404EE:85A2ED1CA59D0479B1E3406972AB1928::: bert:501:C6E4266FEBEBD6A8AAD3B435B51404EE:0B9957E8BED733E0350C703AC1CDA822::: admin:502::333CB006680FAF0A417EAF50CFAC29C3:D2EDBC29463C40E76297119421D2A707:::	Fred: Bert: Admin:

Repeat all 3.1, 3.2 and 3.3 using **Ophcrack**, and the rainbow table contained on the instance (rainbow_tables_xp_free).

Lab 3b: Digital Certificates

1 Introduction

No	Description	Result
1	From: `http://asecuritysite.com/encryption/digitalcert` Open up certificate 1 and identify the following.	Serial number: Effective date: Name: Issuer: What is CN used for: What is ON used for: What is O used for: What is L used for:
2	Now open-up the ZIP file for the certificate, and view the CER file.	What other information can you gain from the certificate: What is the size of the public key: Which hashing method has been used: Is the certificate trusted on your system: [Yes][No]

3	For Example 2 to Example 6. Complete the following table:	

Cert	**Organisation (Issued to)**	**Date range when valid**	**Size of public key**	**Issuer**	**Root CA**	**Hash method**	**Is it trusted?**
1							
2							
3							
4							
5							
6							

2 PFX files

We have a root certificate authority of My Global Corp, which is based in Washington, US, and the administrator is admin@myglobalcorp.com and we are going to issue a certificate to My Little Corp, which is based in Glasgow, UK, and the administrator is admin@mylittlecorp.com.

No	Description	Result
1	We will now view some PFX certificate files, and which are protected with a password: `http://asecuritysite.com/encryption/digitalcert2`	For Certificate 1, can you open it in the Web browser with an incorrect password: Now enter "apples" as a password, and record some of the key details of the certificate: Now repeat for Certificate 2:
2	Now with the PFX files (contained in the ZIP files from the Web site), try and import them onto your computer. Try to enter an incorrect password first, and observe the message.	Was the import successful? If successful, outline some of the details of the certificates:

3 Creating certificates

Now we will create our own self-signed certificates.

No	Description	Result
1	Create your own certificate from: `http://asecuritysite.com/encryption/createcert` Add in your own details.	View the certificate, and verify some of the details on the certificate. Can you view the DER file?

4 Creating a self signed certificate

You will be assigned a folder in vCentre. Navigate to Production->crypto->netxx and then startup your Kali instance.

We have a root certificate authority of My Global Corp, which is based in Washington, US, and the administrator is admin@myglobalcorp.com and we are going to issue a certificate to My Little Corp, which is based in Glasgow, UK, and the administrator is admin@mylittlecorp.com.

No	Description	Result
1	On Kali, login and get an IP address using: `sudo dhclient eth0`	

2	Create your RSA key pair with:
	
	`openssl genrsa -out ca.key 2048`
	
	Next create a self-signed root CA certificate ca.crt for My Little Corp:
	
	`openssl req -new -x509 -days 1826 -key ca.key -out ca.crt`
		How many years will the certificate be valid for?
		
		Which details have you entered:

3	Next go to Places, and from your Home folder, open up ca.crt and view the details of the certificate.	Which Key Algorithm has been used: Which hashing methods have been used: When does the certificate expire: Who is it verified by: Who has it been issued to:

4	Next we will create a subordinate CA (My Little Corp), and which will be used for the signing of the certificate. First, generate the key:
	
	`openssl genrsa -out ia.key 2048`
	
	Next we will request a certificate for our newly created subordinate CA:
	
	`openssl req -new -key ia.key -out ia.csr`
	
	We can then create a **certificate from the subordinate CA** certificate and **signed by the root CA**.
	
	`openssl x509 -req -days 730 -in ia.csr -CA ca.crt -CAkey ca.key -set_serial 01 -out ia.crt`
		View the newly created certificate.
		
		When does it expire:
		
		Who is the subject of the certificate:
		
		Which is their country:
		
		Who signed the certificate:
		
		Which is their country:
		
		What is the serial number of the certificate:
		
		Check the serial number for the root certificate. What is its serial number:

5	If we want to use this certificate to digitally sign files and verify the signatures, we need to convert it to a PKCS12 file:	Can you view ia.p12 in a text edit?

	`openssl pkcs12 -export -out ia.p12 -inkey ia.key -in ia.crt -chain -CAfile ca.crt`	

6	The crt format is in encoded in binary. If we want to export to a Base64 format, we can use DER:
	
	`openssl x509 -inform pem -outform pem -in ca.crt -out ca.cer`
	
	and for My Little Corp:
	
	`openssl x509 -inform pem -outform pem -in ia.crt -out ia.cer`
		View each of the output files in a text editor (ca.cer and then ia.cer). What can you observe from the format:
		
		Which are the standard headers and footers used: