Rapport de Devoir Maison d'initiation à la Cybersécurité

Matthieu Jolimaitre matthieu.jolimaitre@epita.fr

Table of Contents

Exploring Macs and Hash Functions

Task 2: Checking Software Digests

1. In Task 2 you should have shown that the software you downloaded from the website was the same software posted on the website. What does it mean if they do not match? If an attacker could break into the website to replace the software with something malicious, what else would the attacker need to replace to get away with it?

Un fichier transmis par un tiers peut avoir été compromis. Pour garantir que l'état du fichier est le bon, on peut comparer son empreinte avec celle publiée par une source de confiance pour ce fichier en bon état.

> **Exemple** Le site de la distribution Archlinux propose à des miroirs bénévoles de distribuer le fichier ISO d'installation de la distribution.
> Nous pouvons télécharger le fichier distribué par ovh.net :

```
wget "https://archlinux.mirrors.ovh.net/archlinux/iso/2024.06.01/archlinux-x86_64.iso"
```

L'empreinte d'un fichier est un hachage relativement court du contenu du fichier.

> **Exemple** Pour générer l'empreinte « SHA-256 » nous pouvons utiliser la commande UNIX correspondante :

```
sha256sum "archlinux-x86_64.iso"
# 4cc7e1c9f4e97b384f0d8731f317b5995bde256fcc17160d32359cab923c5892 archlinux-x86_64.iso
```

Une empreinte égale à celle déclarée par le site garanti que le contenu du fichier téléchargé correspond au fichier que le site souhaite distribuer.

Exemple Le site de la distribution Archlinux publie que l'empreinte du fichier iso authentique est le suivant :

```
File integrity checksums and PGP signatures for the latest releases can
be found below:
- SHA256:
4cc7e1c9f4e97b384f0d8731f317b5995bde256fcc17160d32359cab923c5892
```

Nous notons que l'empreinte est la même que celle du fichier que nous avons téléchargé. Cela signifie que le fichier distribué par ovh.net n'a pas été altéré par rapport au fichier original.

Task 3: Exploring the "Avalance Effect"

2. Describe the differences between the two digests of iou.txt when the difference between the two inputs is only one bit.

Une différence légère entre deux fichiers génère des empreintes très différentes, cette propriété est désirable puisqu'elle sert à facilement remarquer une altération qu'un fichier a subi.

3. Describe your experience to find another message that matches the original digest of iou.txt

Selon l'algorithme utilisé, trouver un second fichier qui aura la même empreinte qu'un premier fichier peut être très difficile. Sauf vulnérabilité dans l'algorithme de hachage utilisé, la meilleure stratégie demeure la tentative par « force brute », c'est à dire essayer toutes les possibilités de messages.

Cela est une qualité pour un algorithme d'empreinte : Il sera plus difficile pour un pirate de remplacer un fichier sain par un fichier dangereux sans que cela ne soit détectable par un changement d'empreinte.

Deux fichiers différents partageant la même empreinte est un phénomène appelé collision de hachage.

4. Referring to your observations and experiences recorded in worksheet items #2 and #3, how does the avalanche effect make it difficult to find two messages that hash to the same value?

L'effet « avalanche » est qu'une faible altération d'un fichier altère grandement son empreinte.

Il rend très difficile la tâche de « corriger » une différence dans le hachage du fichier après une altération.

Task 4: Exploring Second Pre-Image Resistance

5. The last four hex digits of the SHA256 digest of declare.txt

```
sha256sum "declare.txt"
# f98b1c252622e4a4f6edae5bdd135deb554693f6b3e5b4c35a1f30497bd77bc8
declare.txt
```

Les 4 derniers digits: `7bc8`.

6. The number of attempts to match on the last hex digit of the digest for declare.txt.

Tentative	Nombre d'éssai	Tentative	Nombre d'éssai
1	1	6	3
2	79	7	21
3	33	8	34
4	4	9	52
5	2	10	2

7. Referring to the data reported in item #6 above, why are the results so different amongst the ten attempts to find a match on the last digit of the digest? If a hex digit only has 16 possible outputs, why would it take more than 16 times to sometimes find a collision?

Plusieurs candidats tirés consécutivements peuvent finir par le même digit, ce qui implique que les empreintes de 16 essais consécutifs peuvent contenir plusieurs fois le même digit, et donc ne pas contenir le digit recherché.

8. The number of attempts to match on the last two hex digits of the digest for declare.txt.

Tentative	Nombre d'éssai	Tentative	Nombre d'éssai
1	157	6	58
2	38	7	415
3	93	8	72
4	827	9	71
5	875	10	84

9. The number of attempts to match on the last three hex digits of the digest for declare.txt.

Tentative	Nombre d'éssai	Tentative	Nombre d'éssai
1	1045	6	6062
2	2224	7	2711
3	3242	8	33
4	1883	9	399
5	26159	10	741

10. Referring to the graph in the Excel spreadsheet, what kind of pattern do you see in the graph? What does this suggest if you tried to find a match against the entire hash for declare.txt?

La relation du nombre approximatif d'essai nécessaire en fonction de la taille du hachage que nous voulons trouver identique semble être exponentielle.

Cela signifie que vouloir trouver une collision d'un hachage entier n'est pas réaliste.

11. The number of attempts to find two random messages whose digests match on the last byte (i.e., the last two hex digits).

Tentative	Nombre d'éssai	Tentative	Nombre d'éssai
1	13	6	35
2	10	7	16
3	12	8	40
4	22	9	5
5	10	10	25

12. Referring to the tables in items #8 and #11, when only two hex digits needed to match, explain why it required less effort to find a collision in #11 than it did in #8.

La complexité du problème est moindre : À chaque tentative N, nous cherchons N-1 valeurs plutôt qu'une seule.

Dès la troisième tentative, nous cherchons deux hachages spécifiques, il est deux fois plus probable de réussir.

Task 6: Exploring Message Authentication Codes

13.Describe your observations about the differences in the outputs when different keys are used to generate an HMAC for the same file.

…

14.At the end of Task 6 you found a MAC key through a "brute force" effort. What could an adversary do if he could determine the MAC key that is used to protect the integrity of communications between two people?

…

PCAP Library Programming

Unknown trace

What link-layer is included in the trace?

```
pcap_analyzer ./trace2.pcap links
# link  EN10MB  Ethernet
```

La trace contient un échantillon de trafic Ethernet.

What is the snap length and what is the significance of the snapshot length? The link type defined in the packet trace header is important as we must skip over the correct amount of data to reach the IP packet (which is what were really interested in).

La « snap length » est un paramètre des accesseurs d'un paquet, il sert à limiter la taille des données lues dans un paquet.

Find the documentation for PcapNg online. Briefly (no more than 2 or 3 sentences) describe the differences between pcap and PcapNg.

Le PcapNg introduit les fonctionnalités suivantes : - Un seul fichier peut contenir plusieurs liens. - Des annotations peuvent être ajoutés aux trammes. - Des structures spécialisés permettent de compacter les données réccurentes (addresses, clés).

Basic traffic stats

How many IPv4 packets does the trace contain (as IPv4 count:)?

```
pcap_analyzer ./trace2.pcap stats
# Count:                 30611000
# Count IPv4:            28893393
# non-IPv4 count:         1717607
# First timestamp:  1474265898.92
# Last timestamp:   1474309098.10
# Avg packet rate:         708.60
# Errors:                1717607
#
# Main Protocols:
# - 6 (TCP - Transmission Control)       28893393 ( 94.39%)
#
# Unique sources:                   988082
# Unique destinations:               32769
# Source with most bytes:   "[58, 51, 150, 96]"
# Source with most packets: "[58, 51, 150, 96]"
```

La trace contient 28 893 393 paquets Ipv4.

How many non-IPv4 packets does the trace contain (as non-IPv4 count:)?

La trace contient 1 717 607 paquets non-Ipv4.

What is the timestamp of the first packet in the trace, including at least two decimal places. (as First timestamp:)?

Le timestamp du premier paquet de la trace est 1 474 265 898.92 secondes.

What is the average packet rate (in packets per second to two decimal places) of the trace (as Avg packet rate:)?

Le taux de paquets de la trace est 0.0014 paquet par seconde.

What is the packet protocol distribution? (A table showing the 5 top protocols and their respective contributions is fine.)

94.39% des paquets sont des paquets TCP. Les autres paquets n'ont pas d'entête Ipv4.

Plot a histogram of the packet size distribution (the Python numpy and matplot packages are installed on the Labtainer).

…

How many unique IPv4 source addresses are present in the trace (as Unique sources:)?

La trace montre des paquets ayant 988 082 sources différentes.

How many unique IPv4 destination addresses are present in the trace (as Unique destinations:)?

La trace montre des paquets ayant 32 769 destinations différentes.

Create a cumulative distribution function (CDF) plot. The x-axis is the number of bytes sent and the y-axis is the cumulative fraction of sources.

…

Which source sent the most bytes (as Source with most bytes:)?

L'addresse ayant envoyée le plus grand nombre d'octets est 58.51.150.96.

Which source sent the most packets (as Source with most packets:)? Based on your analysis of the trace:

L'addresse ayant envoyée le plus grand nombre de packets est 58.51.150.96.

List 3 characteristics of the traffic that seem unusual to you.

- Le taux de trafic est élevé.
- Beaucoup de paquets sont envoyés par 58.51.150.96 (publique, classe A).
- Il y a bien plus d'addresses de sources que de destinations.

Provide a reasonable explanation for what traffic the trace represents, taking into account the unusual characteristics you have identified.

Le trafic est probablement issu d'un routeur.