M2 informatique (parcours GLIA)
Année universitaire 2019-2020 - Session 1

Consignes : Les documents sont interdits. Les calculatrices et téléphones portables sont interdits.

Examen d'administration réseau

Exercice 1 – Le protocole QUIC (9 points)

Le protocole QUIC est un protocole dont les buts principaux incluent : la réduction du temps d'établissement des connexions HTTP sécurisées, le multiplexage sans blocage de tête de ligne (au travers de multiples flux), le déploiement en ne modifiant que les extrémités.
Question 1.1 (1 point) : Depuis HTTP 1.1, une connexion HTTP persistente peut être utilisée pour télécharger plusieurs documents (par exemple : une page HTML et toutes les images de la page). Lorsqu'une connexion HTTP se base sur un unique flux TCP, un blocage (dit "de tête de ligne") peut arriver si un des documents n'est pas disponible, est long à être transmis, ou a subi une perte. Expliquez pourquoi, en donnant des arguments techniques liés à TCP.
Question 1.2 (1 point) : Quels sont les mécanismes de sécurité des connexions TCP ? Expliquez en quoi un protocole (comme QUIC) peut accélérer l'établissement d'une connexion HTTPS, sachant qu'une connexion HTTPS est une combinaison du protocole HTTP et du chiffrement TLS.
Question 1.3 (1 point) : Que signifie le fait que le déploiement ne doit pas modifier les extrémités ?

Le draft IETF quick-transport-23 indique « To enable a receiver to limit memory commitment to a connection and to apply back pressure on the sender, streams are flow controlled both individually and as an aggregate. »
Question 1.4 (1 point) : Rappelez ce qu'est le contrôle de flux *(flow control)* de TCP.
Question 1.5 (1 point) : Pourquoi est-il important de faire du contrôle de flux agrégés dans QUIC ?
Question 1.6 (1 point) : Comment ce contrôle de flux peut-il être implémenté ?

Le draft indique aussi « QUIC endpoints can use Explicit Congestion Notification (ECN) [RFC3168] to detect and respond to network congestion. ECN allows a network node to indicate congestion in the network by setting a codepoint in the IP header of a packet instead of dropping it. Endpoints react to congestion by reducing their sending rate in response (…) »
Question 1.7 (1 point) : En quoi ce mécanisme de contrôle de congestion est potentiellement plus efficace que celui de TCP ?

Le draft IETF indique aussi « An ACK frame SHOULD be generated for at least every second ack-eliciting packet. This recommendation is in keeping with standard practice for TCP [RFC5681]. A receiver's delayed acknowledgment timer SHOULD NOT exceed the current RTT estimate or the value it indicates in the "max_ack_delay "transport parameter. » et « In order to assist loss detection at the sender, an endpoint SHOULD send an ACK frame immediately on receiving an ack-eliciting packet that is out of order. The endpoint MAY continue sending ACK frames immediately on each subsequently received packet, but the endpoint SHOULD return to acknowledging every other packet after a period of 1/8 x RTT, unless more ACK-eliciting packets are received out of order. »
Question 1.8 (1 point) : Expliquez la différence entre les acquittements retardés de TCP et de QUIC.
Question 1.9 (1 point) : Expliquez la dernière partie de la citation.

Exercice 2 – Routage RIP (3 points)

La figure 1 représente un réseau de cinq routeurs fonctionnant avec le protocole RIP. À un instant *t*, les nœuds ont les vecteurs de distance suivants : Pour a : {(b,1,b) (c,2,b) (d,3,b) (e,2,b)}, pour b : {(a,1,a) (c,1,c) (d,2,e) (e,1,b)}, pour c : {(a,3,e) (b,1,b) (d,2,e) (e,1,e)}, pour d : {(a,3,e) (b,2,e) (c,2,e) (e,1,e)}, et pour e : {(a,2,b) (b,1,b) (c,1,c) (d,1,d)}. Pour rappel, le triplet (*x*,*y*,*z*) signifie que le nœud *x* est à distance *y* en passant par *z*.

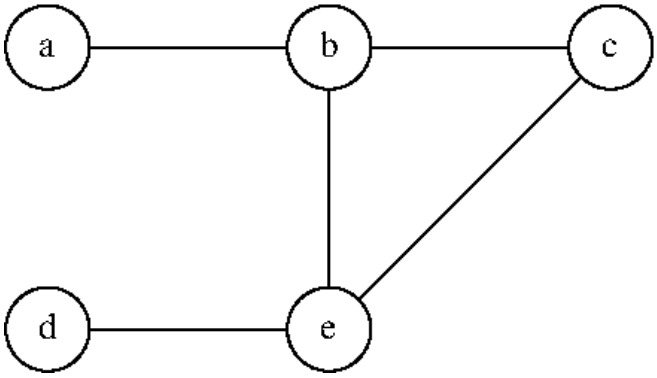

Figure 1 : un exemple de réseau pour RIP.

Question 2.1 (1 point) : Est-ce que le routage a convergé ?
Question 2.2 (1 point) : Supposons qu'un lien soit ajouté entre les routeurs a et d. Quel est l'impact sur le vecteur de distances de a ? Quel sera l'impact (à terme) sur le vecteur de distances de b ?
Question 2.3 (1 point) : Représentez les vecteurs de distance de tous les nœuds, une fois que le routage aura convergé, sur la nouvelle topologie (avec le lien entre a-d).

Exercice 3 – Gestion de parc informatique (4 points)

Un administrateur réseau/système doit avoir un outil efficace de gestion de son parc informatique, tant au niveau matériel qu'au niveau logiciel.
Question 3.1 (1 point) : Pourquoi faut-il avoir une bonne connaissance du matériel ? Et du logiciel ?
Question 3.2 (1 point) : Quelles sont les contraintes pour gérer les interventions ?
Question 3.3 (1 point) : Quelles sont les contraintes pour gérer les licences logicielles payantes ?
Question 3.4 (1 point) : Quelles sont les contraintes pour gérer le renouvellement des machines ?

Exercice 4 – Analyse de logs (4 points)

Dans cet exercice, nous cherchons à étudier une attaque à partir de logs HTTP d'un serveur Apache. L'attaque a été détectée car des fichiers sur le serveur ont été modifiés à 6h03, alors qu'ils ne devraient pas être modifiables. Il semble que l'attaque ait été une attaque d'injection sur un module PHP. La figure 2 donne un extrait des logs.

```
68.180.228.246 - - [28/Mar/2015:06:02:02 +0100] "GET /lidentite-0 HTTP/1.1" 200 8409 1
157.55.39.120 - - [28/Mar/2015:06:02:22 +0100] "GET /printmail/5732 HTTP/1.1" 200 8176 1
157.55.39.120 - - [28/Mar/2015:06:02:35 +0100] "GET /print/1832 HTTP/1.1" 200 4531 0
157.55.39.120 - - [28/Mar/2015:06:02:43 +0100] "GET /print/5041 HTTP/1.1" 200 6582 0
81.163.131.61 - - [28/Mar/2015:06:03:07 +0100] "POST /index.php?q=fckeditor%2Fxss HTTP/1.1"
404 29446 1
81.163.131.61 - - [28/Mar/2015:06:03:09 +0100] "POST /index.php?q=ckeditor%2Fxss HTTP/1.1"
200 3 0
81.163.131.61 - - [28/Mar/2015:06:03:10 +0100] "POST /index.php?q=ckeditor%2Fxss HTTP/1.1"
```

```
200 13 0
81.163.131.61 - - [28/Mar/2015:06:03:11 +0100] "GET /sites/all/themes/wtm7973n.php
HTTP/1.1" 200 109 0
91.194.60.86 - - [28/Mar/2015:06:05:01 +0100] "GET /cron.php?cron_key=SkL5Ge7 HTTP/1.1" 200
3 10
207.46.13.9 - - [28/Mar/2015:06:05:12 +0100] "GET /print/2862 HTTP/1.1" 200 4324 1
148.251.68.67 - - [28/Mar/2015:06:05:18 +0100] "GET / HTTP/1.1" 200 50604 1
188.165.15.98 - - [28/Mar/2015:06:05:28 +0100] "GET /concours HTTP/1.1" 200 9948 0
157.55.39.120 - - [28/Mar/2015:06:05:51 +0100] "GET /node/6164 HTTP/1.1" 200 9574 1
207.46.13.9 - - [28/Mar/2015:06:06:42 +0100] "GET /breves/www.news.fr HTTP/1.1" 200 9787 1
83.169.91.4 - - [28/Mar/2015:06:06:58 +0100] "GET / HTTP/1.1" 200 50604 1
81.163.131.61 - - [28/Mar/2015:06:07:06 +0100] "POST /sites/all/themes/wtm7973n.php
HTTP/1.1" 200 1123 1
81.163.131.61 - - [28/Mar/2015:06:07:08 +0100] "GET /wp-conf.php?t8449n=1 HTTP/1.1" 200
29207 0
125.209.235.176 - - [28/Mar/2015:06:07:14 +0100] "GET /robots.txt HTTP/1.1" 200 649 0
125.209.235.179 - - [28/Mar/2015:06:07:15 +0100] "GET / HTTP/1.1" 200 10929 1
```

Figure 2 : extrait de logs (source : https://www.octopuce.fr/analyser-une-attaque-avec-les-logs-dapache2/, consulté le 28 octobre 2019).

Question 4.1 (1 point) : Quelles lignes vous semblent suspicieuses ?

L'attaque utilise une faille connue du module Drupal (cf SA-CONTRIB-2014-098) : « The CKEditor module (and its predecessor, FCKeditor module) allows Drupal to replace textarea fields with CKEditor 3.x/4.x (FCKeditor 2.x in case of FCKeditor module) - a visual HTML editor, sometimes called WYSIWYG editor. Both modules define a function, called via an ajax request, that filters text before passing it into the editor, to prevent certain cross site scripting attacks on content edits (that the JavaScript library might not handle). Because the function did not check a CSRF token for anonymous users, it was possible to perform reflected XSS against anonymous users via CSRF. »

Question 4.2 (2 points) : Expliquez l'attaque en vous aidant des lignes suspicieuses et de la description de la faille.

Question 4.3 (1 point) : Comment corriger l'attaque ?