

Partie 1

Révision technique de code source

1.

Gerrit est une application web de revue de code collaborative pour projets open source. Il utilise git pour gérer des changements de code.

2.

Un patch est un bout de code qui contient des changements qui ont été faits. Il montre aussi les différences avec la version précédente.

3.

Open : Le sujet est ouvert, c'est à dire en cours de développement.

Merged : La modification a été acceptée et fusionnée dans le projet.

Abandoned : Le sujet a été abandonné, il ne sera définitivement pas intégré au projet.

4. L'auteur est Anders Broman.

5.

Files		Open All	Diff against: Base	
File Path	Comments	Size		
Commit Message				
epan/dissectors/asn1/gprscdr/gprscdr.cnf	4			
epan/dissectors/packet-gprscdr.c	4			
	+4, -4			

2 fichiers ont subi un changement :

- epan/dissectors/asn1/gprscdr/gprscdr.cnf
- epan/dissectors/packet-gprscdr.c

6.

Le message décrivant le patch est : "Fix display of iPBInV4Address and iPBInV6Address as IPAddr"

Le but est donc de corriger le fait que iPBInV4Address et iPBInV6Address soit affichés en tant que IPAddr.

7.

La section code-review indique une note qui approuve ou non la revue. Ici une note de 2+ indique que la revue est complètement approuvée

8.1 L'auteur est Hessam Jalali

8.2 6 fichiers ont subi des modifications.

- doc/tshark.pod
- epan/print.h
- epan.print.c
- file.c
- tfshark.c
- tshark.c

8.3

-E extended and added to other -T types

new -E modes:

indent: enable/disable formatting and indentation for pdml and json

per_packet: remove root object/teg from pdml and json and output per packet
similar to ek but in generic JSON or XML.

ordinal_naming: use field ordinal prefixed with f as field name instead of
full name, useful for data aggregation.

occurrence, indent, per_packet and ordinal_naming are supported in json and pdml
only occurrence and ordinal_naming are supported in ek.

Le sujet traite divers modifications de syntaxe

8.4 La note -1 de code review indique qu'il serait préférable de ne pas intégrer le code ainsi, que cela pourrait être fait différemment.

Gestion de bogue

9.

La personne qui a trouvé le bogue s'appelle Travis. Oui c'est un être humain, il a un nom d'humain "travis darnell" et pas de programme/machine.

10. L'importance de ce bogue est faible (Low Enhancement) .

11. Les utilisateurs votent pour la priorité et la sévérité du bogue. Ces deux paramètres déterminent son importance.

12. Non car vu que l'importance est faible, le bug n'est pas critique. De plus, le bug rapporte une fonctionnalité à ajouté et non un problème de fonctionnement de l'application.

13. 2 personnes ont commenté : Michael Mann et Guy Harris.

14. Le statut CONFIRMED indique que le bogue a été validé mais aucune action n'est prévue pour le corriger / implémenter.

15. Le statut CONFIRMED indique que le bogue a été validé mais aucune action n'est prévue pour le corriger / implémenter.

Intégration Continue

16. Buildbot est un outils d'intégration continue qui automatique la compilation des programmes et/ou les tests lors des changement du code source. Dans un projet open source, il permet de garantir une bonne intégrité des contributions en les analysant à chaque ajout.

17.

Les builders sont :

Clang Code Analysis

OSX 10.6 x64

Ubuntu 16.04 x64

Visual Studio Code Analysis

Windows Server 2016 x64

Windows Server 2016 x86

Ce sont les systèmes sur et avec lesquels sont compilés le projet pour vérifier leur intégrité.

18.

Vert : La compilation sur le builders indiqué est un succès.

Orange : La compilation sur le builders indiqué a réussi mais il y a un avertissement.

Rouge : La compilation sur le builders indiqué a échoué.

19. Builder : Clang Code Analysis

Build#4628

1.	prep prepped	(0 secs)
1.	stdio	
2.	remove install directory Deleted	(0 secs)
1.	- no logs -	
3.	remove install directory_1 Deleted	(0 secs)
1.	- no logs -	
4.	remove install directory_2 Deleted	(0 secs)
1.	- no logs -	
5.	git update	(6 secs)
1.	stdio	
6.	compile ran CMake warnings	(17 secs)
1.	stdio	
2.	warnings_1}	
7.	dumpabi dumpabi failed	(1 mins, 24 secs)
1.	stdio	
8.	shell cleaned git	(0 secs)
1.	stdio	
9.	shell_1 created CMake build directory	(0 secs)
1.	stdio	
10.	compile_1 ran CMake warnings	(43 secs)
1.	stdio	
2.	warnings_1}	
11.	scan-build built using scan-build warnings	(28 mins, 18 secs)
1.	stdio	
2.	warnings_123}	
12.	copy-scan-build-output copied scan-build output	(13 secs)
1.	stdio	
13.	shell_2 cleaned git	(0 secs)
1.	stdio	
14.	shell_3 created CMake build directory	(0 secs)
1.	stdio	
15.	compile_2 ran CMake warnings	(21 secs)
1.	stdio	
2.	warnings_1}	
16.	cov-build built using cov-build warnings	(13 mins, 50 secs)
1.	stdio	
2.	warnings_905}	
17.	copy-cov-build-output copied cov-build output	(0 secs)
1.	stdio	
18.	shell_4 cleaned git	(0 secs)
1.	stdio	
19.	shell_5 created CMake build directory	(0 secs)
1.	stdio	
20.	compile_3 ran CMake	(38 secs)
1.	stdio	
21.	compile_4 compile	(1 mins, 36 secs)
1.	stdio	
22.	test-tsan tested threadsanitizer failed	(2 secs)
1.	stdio	
23.	shell_6 cleaned git	(0 secs)
1.	stdio	
24.	shell_7 created CMake build directory	(0 secs)
1.	stdio	
25.	compile_5 ran CMake	(22 secs)
1.	stdio	
26.	compile_6 compile	(2 mins, 11 secs)
1.	stdio	
27.	copy-sloccount generated sloccount report	(14 secs)
1.	stdio	
28.	install installed	(1 secs)
1.	stdio	
29.	valgrind-wireshark valgrinding	(running)
1.	stdio	

20.

dumpabi failed:

La compilation a échoué car le fichier window.h est introuvable
fatal error: windows.h: No such file or directory

test-tsan : le test a échoué

21. Le numéro de la révision est : 12f8f43893ea3d31e4f18b356af581e14becd8a3

L'auteur de la revue est : Gerald Combs

Partie 2

Contribution à un projet

22. Log1000 correspond ici à un groupe (public). Il peut contenir plusieurs repos, ici il en contient qu'un seul : TP3

23. Un fork est une copie du projet sur notre propre compte gitLab et ainsi créer un nouveau remote repository. Les changements effectués n'affecteront donc pas le projet dans le groupe LOG100.

24. /

25. `git clone https://gitlab.com/alexismalzieu/tp3.git`

26. Parce que le fichier html est présent localement sur notre ordinateur. Il est ainsi lisible directement par notre navigateur.

Modification de code

27.
`git add --all`

28. Un merge request est une requête faite aux administrateurs du repo principal du projet (ici le LOG1000) afin qu'ils puissent intégrer une contribution au projet. C'est une demande car les administrateurs peuvent la valider ou non.

29. L'avantage d'un merge request, est que l'administrateur principal peut accepter ou refuser la demande d'intégration de la branche au projet. Aussi, du côté des contributeurs, on aura toujours le fork du projet sur le remote repo quelque soit le résultat du merge request.

30.
Branche source : alexismalzieu/equipe77
Branche destination: tp3/master

Révision de code

31.
Nous révisons le code de l'équipe 91.

32.

Le commit associé s'appelle "modification onglet", SHA : **b6e8b886**

33.

Le fichier "index.html" a été modifié.

L'image DevOps.jpg a été ajoutée au dossier assets/img.

34.

Le fichier contient bien les éléments demandés

35.

Respect des consignes et aucun problème de syntaxe ou de code.

Note : 5/5

36.

done