Édition 2022

DOSSIER DE CANDIDATURE
PRÉSENTATION DU PROJET

~$ ETM

Encrypted Terminal Messaging

Ce document est l'un des livrables à fournir lors du dépôt de votre projet : 4 pages maximum (hors documentation).

Pour accéder à la liste complète des éléments à fournir, consultez la page **Préparer votre participation**.

Vous avez des questions sur le concours ? Vous souhaitez des informations complémentaires pour déposer un projet ? Contactez-nous à **info@trophees-nsi.fr**.

NOM DU PROJET : ETM *(Encrypted Terminal Messaging)*

> PRÉSENTATION GÉNÉRALE :

Idée et objectifs :

L'idée principale d'ETM est d'avoir une messagerie fonctionnant dans un terminal de manière simple, intuitive et sécurisée pour tout type d'utilisateur, quel que soit son niveau en informatique ou la plateforme depuis laquelle est exécuté le programme. En effet, le problème de la vie privée étant devenu un enjeu majeur, nous avons voulu créer une messagerie cryptée et totalement transparente sur son mode de fonctionnement car open source et conservant un mode de fonctionnement le plus simple possible. Pour cela, 4 objectifs principaux étaient à atteindre :

- ETM doit pouvoir s'exécuter sur le maximum possible d'appareils pouvant exécuter python ;
- La structure du code d'ETM, son mode de fonctionnement et son interface doivent être les plus simples possible pour le rendre compréhensible par le plus grand nombre et permettre sa modification et son utilisation de manière simple ;
- Un chiffrement RSA de bout en bout doit être implémenté pour permettre une communication sécurisée (la clé de chiffrement fait actuellement 4096 bits, paramètre facilement modifiable) ;
- ETM doit fonctionner en P2P : les communications ne passant par aucun serveur, les questions de privacité et de facilité d'utilisation sont en partie réglées (car pouvant fonctionner de manière autonome) ;

Origine et intérêts du projet :

Le projet est né après une réflexion sur les problèmes et les besoins liés à la technologie actuelle ; 2 enjeux se sont démarqués : l'accessibilité et la sécurité. C'est de là que ETM tire son épingle du jeu : modifiable et utilisable par tous de manière aisée, il suffit de savoir lancer un programme python dans un terminal pour pouvoir en profiter. Son fonctionnement étant le plus simple possible, ETM est modifiable par quiconque possède des bases suffisantes en programmation python et en fonctionnement réseau (principalement sockets réseaux et multiprocessing). Finalement, les conversations étant chiffrées de bout en bout par un cryptage RSA 4096 bits, le tout est le plus sécurisé possible. La structure P2P libère le programme de la contrainte de devoir passer par des serveurs et tous les inconvénients qui vont avec.

> ORGANISATION DU TRAVAIL :

Présentation de l'équipe :

Équipe 1 : développement d'ETM :

- Sasha Guérin-Loison : développeur python et concepteur ;
- Marwan Doghmi : concepteur et testeur principal ;

Équipe 2 : création du site web :

- Hugo Villepontoux : créateur site web et testeur ;
- Yassine Diallo : créatrice site web et testeuse ;

Répartition des tâches :

Les tâches ont été réparties de la manière la plus équitable possible : avant chaque modification du projet avait lieu une réunion des concepteurs qui se chargeaient de définir précisément ce qui devait être fait. Ensuite, Marwan recherchait les solutions les plus optimales possibles pour l'ajout de la fonctionnalité. Après concertation, s'en suivait la phase d'implémentation à proprement parler : Sasha se chargeait d'apporter les modifications nécessaires au code. Finalement avaient lieu les phases de test où Marwan repérait les bugs, comportements anormaux / inattendus ou problèmes d'optimisation et en identifiait la source précise afin de permettre à Sasha de corriger ces problèmes et d'optimiser au mieux le code. Pour le site web, Ugo et Yassine suivaient là même d'organisation, adaptée au site web.

Organisation du travail :

- Réunions : très fréquentes (quotidiennes ou presque) mais assez courtes (10-30mn) sauf problème, réunion de conception ou tout autre raison particulière ;
- Travail en dehors de l'établissement : occupe la majeure partie du travail : séances de développement, debugging, test, ou conception occupaient la majeure partie du travail et se faisaient en dehors de l'établissement ;
- Travail au sein de l'établissement : assez rare, principalement pour demander conseil au professeur en cas de besoin ;
- Outils utilisés : pour le développement, Visual Studio Code. Pour le test, terminal Windows 10/11, Linux (Debian) et mobile (Termux sur Android 11/12). Pour la communication, Discord. Pour le stockage / partage de code, Github. Site web, Wix ;

LES ÉTAPES DU PROJET :

1 – Phase de conception initiale (définition précise de l'idée / concept du projet, définition des fonctionnalités à mettre en place ainsi que de leur fonctionnement) ;
2 – Implémentation des fonctionnalités (procédure répétée pour chacune d'elle) :
- Définition précise de la fonctionnalité et de son fonctionnement ;
- Développement ;
- Test ;
- Debugging / optimisation ;
- Test final de la fonctionnalité ;

3 – Test final du projet et perspectives d'évolution ;
4 – Rédaction de la documentation ;

> FONCTIONNEMENT ET OPÉRATIONNALITÉ :

Avancement du projet :

ETM et son site sont totalement terminés pour leur première version. Tous les bugs connus et ayant pu être détectés lors des tests ont été corrigés, et toutes les fonctionnalités qui devaient être implémentée dans celle-ci le sont pleinement : ETM 1.0 est donc terminé à 100 %.

Bien sûr, un projet pouvant toujours être amélioré, nous avons des idées de fonctionnalités à implémenter : le partage de fichiers est la fonction la plus à même d'être implémentée prochainement.

Approches mises en œuvre (debugging, facilité d'utilisation) :

Pour vérifier l'absence de bugs, la même procédure a été répétée après chaque implémentation de fonction : test de l'ensemble des fonctions du projet, test approfondi de la fonction implémentée (lors de ce test, nous essayons de faire bugger la fonctionnalité / le programme pour corriger toutes les bugs / erreurs pouvant survenir lors d'une utilisation normale. Par exemple, saturation du socket, envoi de caractères spéciaux, tentative d'injection de commande, etc.). Aussi, nous faisons tester le programme par la 2ème partie de l'équipe qui gère le site web et qui aborde donc le programme avec la même démarche qu'un utilisateur lambda.

Pour s'assurer de la facilité d'utilisation, nous avons fait en sorte que l'utilisateur n'ait, la plupart du temps, qu'à choisir entre plusieurs options et que quand il ait à entrer une valeur, les structures de test implémentées dans le code s'assurent que la valeur soit valide : l'utilisateur ne peut pas se tromper en interagissant avec le programme car il sera toujours guidé. Aussi, la documentation est la plus claire possible pour que l'utilisateur, même peu expérimenté, y trouve les réponses à toutes les questions qu'il est susceptible de se poser concernant l'utilisation d'ETM.

Difficultés rencontrées et solutions apportées :

Les principales difficultés concernaient la mise en place d'une organisation efficace. Des difficultés plus techniques se sont également présentées avec les concepts un peu plus avancés en python (multiprocessing ou sockets réseau). Dans tous les cas, prendre du recul et du temps pour arranger ce qui devait l'être et approfondir nos connaissances lorsque cela s'imposait a résolu le problème de manière naturelle.

> OUVERTURE :

Idées d'amélioration :

- Partage de fichiers ;
- Possibilité d'être à plus de 2 dans un même salon ;
- Possibilité de bloquer des adresses IP ;
- Affichage du statut (hors ligne ou en ligne) des contacts lors de leur affichage ;
- Possibilité d'ajouter un mot de passe aux salons lors de leur création ;
- Affichage du pseudo et non de l'adresse IP dans les discussions ;
- Implémenter nativement une fonction VPN (de type Hamachi) pour pouvoir se passer du port forwarding et de l'utilisation de logiciels tiers ;
- Possibilité de quitter le salon quand on l'héberge mais que personne ne se connecte sans devoir quitter le programme ;
- Création d'une version « parallèle » disposant d'une interface graphique ;

Stratégie de diffusion :

Ceci se ferait en plusieurs étapes :

- Créer un serveur Discord pour le projet, sans le révéler au grand public dans un premier temps ;
- Créer un site web temporaire n'affichant qu'un compte à rebours et le logo d'ETM ;
- Créer un fichier .exe (à l'aide de python et auto-python-to-exe), affichant lui aussi un compte à rebours et le logo d'ETM, ainsi que des faits méconnus / inquiétants / marquants sur les risques encourus par-rapport à notre vie privée aujourd'hui « à cause de » la technologie et de l'incompréhension de celle-ci. Ce fichier afficherait aussi un lien qui redirigerait vers le site web temporaire, accompagné d'une légende du type : *« Découvrez comment y faire face »* ;
- Diffuser ce logiciel en le stockant dans des clés USB (frappées du logo d'ETM) de faible capacité que l'on déposerait au sol dans la rue / dans des lieux fréquentés ;
- Une fois que le compte à rebours arriverait à sa fin, le .exe et le site web temporaire afficheront des liens redirigeant non plus vers le site web temporaire, mais vers le site officiel, le serveur Discord et surtout vers le lien de téléchargement du projet ;

Analyse critique :

Il faut tout d'abord dire qu'il aurait été plus simple de pouvoir convertir ETM.py en .exe également, mais la fonction listener() pose un problème : le programme n'a pas été conçu pour pouvoir se re-exécuter dans son entièreté une fois lancé. Or, c'est ce que fait cette fonction en tant que sous-processus lorsque ETM est converti en .exe. Cette erreur peut sûrement être réglée, mais cela comprend une modification du mode de fonctionnement d'ETM, ce qui est délicat à ce stade du projet. D'un point de vue organisation, bien que celle pour laquelle nous avons opté ait porté ses fruits, il aurait été intéressant de voir comment cela se serait passé si au lieu de nous répartir en petits groupes, nous avions tous travaillé sur la même tâche en même temps avant de passer à la suivante.