

# مشروع **FTPChat**

مقدم من:

**أحمد عمر سعد**

الصف السادس الابتدائي

مدرسة جيل المستقبل الدولية

## الفهرس

- المعلومات الشخصية
- ملخص البحث
- المشكلة
- فرضيات البحث
- النتائج
- ما الذي تم إنجازه؟
- الخطط المستقبلية
- إجراءات وخطوات عمل المشروع والتجريب
- تصور كيفية عمله
- ملخص النتائج
- التوافق
- المراجع
- شكر وتقدير

## المعلومات الشخصية

الاسم	أحمد عمر سعد
العمر	11
الصف	السادس
المحافظة:	سوهاج
كود الطالب:	493859824
المدرسة:	مدرسة جيل المستقبل الدولية
الادارة التعليمية:	أحصيم
رقم الهاتف والایمیل:	+201040946638 / +201205201210 Email: ahmedomardev@outlook.com
المجال:	أنظمة البرمجيات (SOFT)

## **ملخص البحث:**

FTPChat هو بروتوكول مراسلة قائم على لغة Python يحل محل الاتصال التقليدي عبر Websockets، وهو يعيد توظيف خوادم FTP مثل تلك المدمجة في أجهزة التوجيه ZTE أو المستضافة على منصات مثل SFTPCloud.io لتصبح مراكز مراسلة آمنة ومركبة.

بدلاً من الحفاظ على اتصالات Websockets الدائمة، يتيح FTPChat للعملاء تبادل الرسائل عن طريق تنزيل ملف مشترك من خادم FTP وفك تشفير محتوياته، وإلaco رسالتهم الخاصة، وإعادة تشفير المحتوى المحدث باستخدام تشفير أحدى الأبجدية من 24 طبقة، وإعادة تحميله. تسمح هذه الدورة لكل عميل بالعمل كقارئ وكاتب، مما يحافظ على تدفق الرسائل المتزامن بين جميع المشاركين.

يدعم البروتوكول الاتصال غير المتزامن والمترافق، مما يجعله مثالياً للبيئات التي يكون فيها الوصول إلى Websockets مقيداً أو محظوظاً بواسطة Firewalls أو غير ممكن تقنياً. يضمن التشفير متعدد الطبقات أن تظل جميع الرسائل آمنة أثناء النقل والتخزين، حتى على خوادم FTP المتاحة للجمهور وي العمل بلا اتصال انترنت (في الشبكات المحلية) ويستهلك كميه قليلة من الطاقة مما يساعد على الحفاظ على البيئة

## **ملخص موجز:**

FTPChat هو بروتوكول مراسلة قائم على لغة Python يستبدل Websockets برسائل مشفرة قائمة على FTP

يستخدم خوادم FTP كمحاور آمنة، مما يتيح الاتصال غير المتزامن حتى في أجهزة الكمبيوتر التي تعمل ببرامج الحماية أو الشبكات القديمة.

يتم تشفير الرسائل باستخدام تشفير من 24 طبقة ويتم تبادلها عبر دورات تحميل/تنزيل.

FTPChat مناسب للبيئات التي يتم فيها حظر Websockets أو تكون غير موثوقة. وي العمل بلا اتصال انترنت (في الشبكات المحلية)

## **مميزاته:**

1. يتيح البرنامج نقل الرسائل بين العديد من أجهزة الكمبيوتر دون استخدام انترنت
2. يتميز بقدر عالي من الأمان والخصوصية نظراً لعدم اتصاله بالإنترنت بالإضافة لتوفير البرنامج 24 مستوى من مستويات التشفير
3. انخفاض التكلفة مقارنة بالأجهزة الأخرى التي تقوم بنفس الوظيفة
4. صغر حجم الجهاز مما يتيح سهولة التنقل به
5. يحقق المشروع أهداف التنمية المستدامة فهو يستهلك طاقة أقل بكثير مما تستهلكه النماذج المشابهة المستخدمة حالياً

## المشكلة

في العديد من بيئات الشبكات المقيدة أو القديمة، مثل تلك الموجودة خلف جدران الحماية أو على أجهزة التوجيه المنزلية أو في البيئات التعليمية، يتم حظر الاتصال التقليدي القائم على Websockets أو عدم دعمه. وهذا يجعل من الصعب إنشاء أنظمة مراسلة في الوقت الفعلي أو تطبيقات Peer-to-peer.

يحل FTPChat هذه المشكلة عن طريق استبدال اتصالات Websockets بترحيل الرسائل القائم على FTP بدلاً من الاتصالات المباشرة، يستخدم العملاء خادم FTP لتبادل الرسائل المشفرة عن طريق تحميل وتنزيل ملف مشترك. وهذا يسمح بالاتصال الآمن وغير المتزامن أو المتزامن حتى في الشبكات التي تكون فيها غير متوفرة أو غير موثوقة أو معطلة عن قصد.

## فرضيات البحث

1. من الممكن إنشاء نظام مراسلة آمن دون الاعتماد على الاتصال القائم على Websockets.
2. يمكن إعادة استخدام خوادم FTP كوسطاء موثوقين لتبادل الرسائل المشفرة.
3. يمكن أن يوفر نظام تشفير أحدى الأبجدية متعدد الطبقات حماية كافية للرسائل المخزنة على خوادم FTP العامة أو شبه العامة.

### الغرض من التصميم:

يهدف التصميم إلى إنشاء بروتوكول مراسلة يعمل في بيئات الشبكات المقيدة أو القديمة حيث يتم حظر اتصالات Websockets أو عدم دعمها. باستخدام FTP كطريقة نقل، يتبع النظام المراسلة المشفرة غير المتزامنة أو المتزامنة بين العملاء. الهدف هو توفير حل آمن وصغير الحجم وسهل الوصول للاتصال عبر الأجهزة والأنظمة الأساسية التي تفتقر إلى الاتصال المباشر.

## النتائج

تم تصميم **FTPChat** لحل مشكلة حظر اتصالات Websockets أو عدم دعمها في بيئات الشبكات المقيدة. في كثير من الحالات، مثل شبكات المدارس أو أجهزة التوجيه القديمة أو الأنظمة التي تعمل بـFirewall، لا يمكن إرسال الرسائل في الوقت الفعلي عبر Websockets بسبب القيود التقنية أو سياسات الأمان.

يستبدل **Websockets** **FTPChat** بترحيل الرسائل القائم على **FTP**. باستخدام خوادم **FTP** كوسطاء، يمكن للعملاء تبادل الرسائل المشفرة بشكل غير متزامن أو متزامن دون الحاجة إلى اتصالات مباشرة. هذا يجعل **FTPChat** مثالياً للبيئات التي تفشل فيها بروتوكولات الشبكات التقليدية، مع الحفاظ على اتصال آمن وموثوق.

# ما الذي تم إنجازه؟

## ما قمت بتنفيذها:

- تحويل مفهوم المراسلة الأساسية الذي يعتمد على وحدة التحكم فقط إلى بروتوكول بسيط وآمن.
- صممت **FTPChat** لتكون قائمة على Terminal User Interface، بدون واجهة مستخدم رسومية، مما يجعلها مثالية لخوادم Linux والبيئات التي تقييد الواجهات الرسومية مع خيار واجهة المستخدم الرسومية.
- بناء النظام الأساسي في Python باستخدام خوادم FTP لارسال الرسائل المشفرة بين العملاء.
- دمج نظام تشفير أحدى الأبجدية مكون من 24 طبقة لتأمين محتوى الرسائل.

## التجارب:

- اختبار **FTPChat** على منصات FTP العامة مثل SFTPCloud.io وعلى أجهزة التوجيه المزودة بـ FTP مدمج مثل ZTE ZXHN H188A.
- التحقق من التوافق مع بيئات خوادم Linux المستخدمة من قبل الشركات التي تفرض الوصول عبر Terminal User Interface فقط.
- محاكاة اتصالات متعددة المستخدمين لضمان التزامن وسلامة التشفير وسهولة الاستخدام.

## التصميمات:

- تم إنشاء بنية معيارية تفصل بين التشفير ومعالجة FTP وتنسيق الرسائل.
- ركزنا على وضوح Terminal User Interface وإمكانية الوصول إليها دون الاعتماد على حجم الشاشة أو التخطيط الرسومي.

## **الخطط المستقبلية**

1. ترقية نظام التشفير من خلال دمج خوارزميات متقدمة مثل AES أو RSA مع الحفاظ على الأداء السريع للبيانات المقيدة.
2. تنفيذ ميزات إدارة الجلسات بما في ذلك مصادقة المستخدم وتنبيه الرسائل والتحكم في تسجيل الدخول لدعم بيانات متعددة المستخدمين.
3. إدخال ضغط الرسائل لتقليل حجم الملفات وتحسين الأداء، خاصة على الشبكات البطيئة أو المحدودة.
4. عمل نسخة لهواتف المحمولة Android و iOS

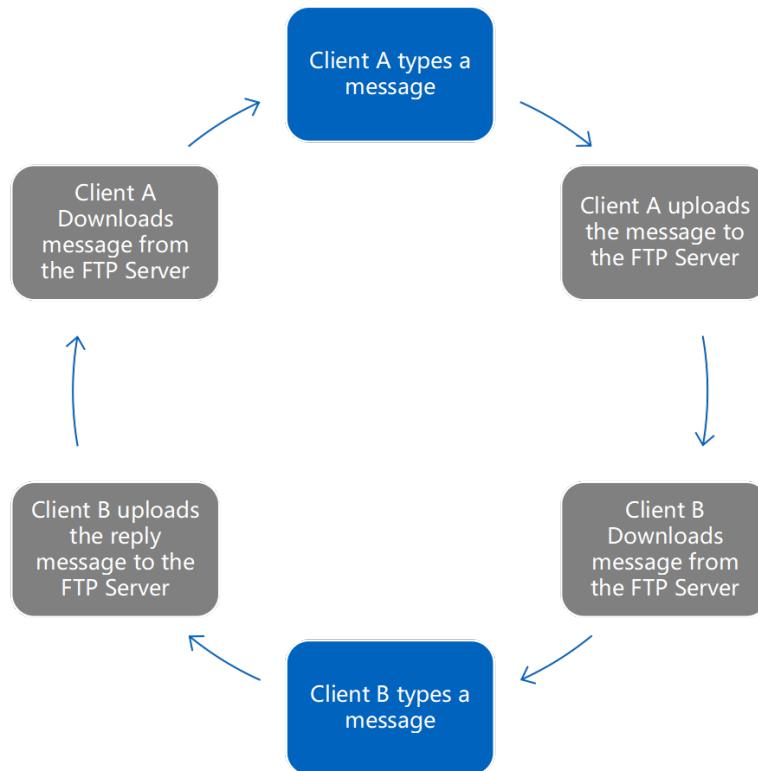
## إجراءات وخطوات عمل المشروع والتجريب

1. تحديد المشكلة (عدم استقرار Websockets والهجمات الإلكترونية)
2. تعلم لغة Python
4. عمل بروتوكول جديد لـ FTPChat
5. تنفيذ التشفير
6. اختبار البروتوكول
7. الاستضافة على أجهزة التوجيه و SFTPCloud.io
8. تصميم واجهة رسومية وغير رسومية

### ملحوظة:

- تم تصميم هذا البروتوكول بشكل مستقل من خلال التجارب الشخصية والتفكير النقدي والاختبارات المتكررة.
- لم يتم استخدام أي أوراق بحثية خارجية أو قوالب كود من جهات خارجية أثناء التطوير. تم تنفيذ جميع طرق التشفير بشكل مستقل.
- المشروع حالياً مغلق المصدر. لا يتتوفر كود المصدر للجمهور في هذه المرحلة.
- تتضمن الخطط المستقبلية إمكانية إصداره كمصدر مفتوح بعد إجراء المزيد من الاختبارات والتوثيق ومراجعة الأمان.

## تصور كيفية عمله



## ملخص النتائج

- بناء نظام مراسلة مشفر بدون Websockets
- استضافة على جهاز التوجيه ZTE و SFTPCloud.io
- عدم الحاجة إلى أدوات Websocket

- تمكين الوصول العالمي مع تشفير معياري

## التوافق

1. FTP .  
- المنفذ: 21 (TCP)  
- الوظيفة: نقل الملفات بين العميل والخادم باستخدام قنوات التحكم والبيانات.  
- السلوك: يعمل في الوضع النشط أو السلبي اعتماداً على تكوين Firewall وNAT.

- المصدر: وثائق شبكات Cisco

SFTP .2

(TCP)22 - المنفذ:

الوظيفة: ينقل الملفات بأمان عبر SSH.

السلوك: يستخدم قناة مشفرة واحدة لتبادل البيانات.

المصدر – Microsoft Learn

3. سلوك Firewall وإمكانية الوصول إلى المنفذ

عادةً ما يُسمح بالمنفذ 21 افتراضياً في:

شبكات المدارس والجامعات

أجهزة التوجيه القديمة المزودة ببروتوكول FTP مدمج مثل ZTE ZHZN H188A

أنظمة المؤسسات التي تستخدم FTP لتحديثات البرامج الثابتة أو إدارة التكوين

عادةً ما يُسمح بالمنفذ 22 في:

أجهزة التوجيه التي تدعم SSH (OpenWRT، DD-WRT)

منصات السحابة مثل AWS و Azure

الشبكات الداخلية الآمنة مع وصول إداري

## مراجع المصدر

1. Cisco: "FTP uses TCP Port 21 for control and data transfer."

Link:

[https://www.cisco.com/c/en/us/td/docs/ios/sw\\_upgrades/interlink/r2\\_0/user/ugftpc1.html](https://www.cisco.com/c/en/us/td/docs/ios/sw_upgrades/interlink/r2_0/user/ugftpc1.html)

2. Microsoft Learn: "Port 22 is used for secure file exchange over SSH."

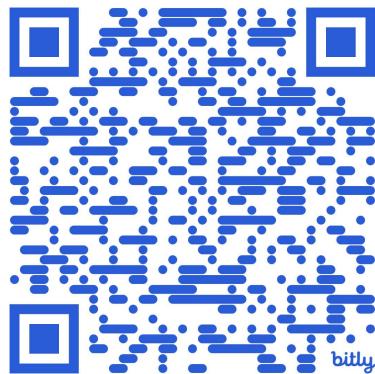
Link:

<https://learn.microsoft.com/en-us/troubleshoot/azure/general/secure-file-exchange-transfer-files>

3. IBM Docs: "Sockets require open ports and persistent connections, which may be blocked by firewalls."

Link:

<https://www.ibm.com/docs/en/i/7.4.0?topic=programming-how-sockets-work>



## شكر و تقدير

أود أن أعبر عن تقديرني العميق لأستاذتي في تكنولوجيا المعلومات والاتصالات أ.نيره السيد.

