درس ۸

چگونه برنامه های ویندوز را کراک کنیم؟

دست گرمی

من، یک نرم افزار قدیمی تر ویندوز را برای کار برگزیده ام (WIN4MANT.EXE, 562271 bytes, Version1.10, by Joseph B. Albanese؛شما میتوانید با جستجو نمودن وب آن را بیابید.) در پایان این درس خواهید دید که چقدر راحت میتوانید این کار را انجام دهید. انتخاب کرده‌ام تا به شما نشان دهم که چگونه از یک ترفند بسیار ساده و دقیق مفید و سودمند برای بکارگیری برنامه‌های محافظت شده رمزی، استفاده کنید: (محدودیت داده). در تقریباً کل روالهای محافظت، همانطور که قبلاً یاد گرفته‌اید، لحظه‌ای وجود دارد که در stack، پژواک (ECHO) واقعی، عدد رمز یا کلمه رمز «صحیح» را نشان میدهد. آدرس این ECHO، تغییر می‌کند. اما اکثر مواقع، دامنه آن بین 0X90+ و0X90 – بایت از یکی از آدرسهایی است که ورودی استفاده کننده در آن وجود دارد . این، ناشی از محدودیتهای روبرداری داده‌های ویندوز در وسایلی است که مورد استفاده محافظت کننده‌هاست. .. اما این استفاده همچنان کاهش یافته است. .. مخصوصاً پس از این درس: =)

بکارگیری وین فرمت

این کاربرد، به تنهایی، بی مفهوم است، بعید میدانم که بخواهید از آن استفاده کنید.. اما با ایـن همـه مد خنثی سازی خارق العاده آن،برای ما بسیار جالب است: چنانچه احساس کنید به وین‌فرمت نیـاز دارید، میتوانید آن را در حال حرکت، ثبت نکنید.

این ویژگی، بقدری برای محققان سودمند است که علاقمند هستند تا در مورد الگوریتمـهای کلمـه رمز همراه با کدهای ارزشمند و بی ارزش تحقیق کنند، بدون آنکه، نیاز باشد تا هر دفعه یک کد بـا ارزش را حـذف کننـد. بـه منظـور بکـارگیـری تمرینـها، برنامـه‌هایی را انتخـاب کنیـد کـه دارای محافظت‌هایReversible «برگشت پذیر» (کمیاب) باشد یا اینکه بتـوان آن را چندیـن بـار مجـدداً ثبت کرد. برنامه‌هایی که ثبت با ارزشی در INI*، یا فایلهای مخصوص دارند، هـم اینکـار را انجـام خواهند داد. شما فقط باید چندین خط را تغییر دهید تا آنها را ثبت نکنید.

ترفند این درس : (محدودیت داده) یا مجاورت کلمه رمز بر اسـاس نیـاز محـافظت کننـد و بـرای توجه نمودن به کار محافظت ، ضمن ترجمه می‌باشـد. او بـاید ارتبـاط بیـن عـدد ورودی اسـتفاده کننده، تبدیل ورودی استفاده کننده و پاسخ صحیح عدد را (به زبان نامهفوم مـا: بینـگو) درک کنـد. این روابط باید دائماً، چک شود تا از کد محافظت، رفع عیب کند.

غالباً آنها. در کنار یکدیگر در قسمت پشته، قرار گرفته و برای آنها این امکان را فراهم می‌آورنـد تـا از طریق همان پنجره. مشاهده شوند. اغلـب اوقـات، ECHO نـه چنـدان دور از یکـی از محلـهای ورودی استفاده کننده، عینیت می‌یابد.

لطفاً بکارگیری را شروع کنید.

ابتدا WINCE و سپس WINFORMAT را روشن کنید.

ابتدا HELP و سپس REGEISTRATION را انتخاب کنید.

فیلدهای ثبت را با ORC+ORC+ تحت عنوان Registrant و 12127272 را بعنوان کد فعالســـازی پر کنید. (هرچه که دوست دارید انتخاب کنید).

گزینه wince CTRL+D

میخواهیم ببینیم که نام این واژه نامفهوم چیست؟ task:

```
CTRL+D              ;switch to Winice
:task               ;let's see what's the name of this crap
TaskName  SS:SP StackTop StackBot StackLow TaskDB  hQueue  Events
WINWORD   1AD7:85F2 4A52  8670      7532     1247    122F    0000
PROGMAN   1737:200A 0936  2070      1392     066F    07F7    0000
DISKOMAT *2C5F:6634 1D3C  6AC6      5192     2CB7    2C9F    0000
```

کدام پنجره ورودی را دریافت می‌کند؟ :hwnd DISKOMAT

```
WinHandle  Hqueue  QOwner    Class Name        Window Procedure
0EB4(0)    2C9F    DISKOMAT  #32769            04A7:9E6B
0F34(1)    2C9F    DISKOMAT  #32768            USER!BEAR306
365C(1)    2C9F    DISKOMAT  #32770            2C3F:0BC6
36BC(2)    2C9F    DISKOMAT  Button            2C3F:1CEA
3710(2)    2C9F    DISKOMAT  Edit
```

... و تعداد زیادی از این پنجره‌های نامربوط

بیایید محل دقیق کد را مشخص کنیم، به دلایلی ، اولین پنجره مربوط در این قسمت، پنجره EDIT (یا ویراستار) است.

نقطه توقف را تعیین کنید؛ bmsg 3710 wm gettext

برنامه babe را اجرا کنید تا آن را پیدا کنید. CTRL+D

توقف به خاطر BMSG 3710 wm GET Text c=d

Hwnd=3710 wparam=…….

خوب! حالا محل دقیق babe را مشخص کرده‌ایم. حالا قدری دور و بر آن را می‌گردیــم: بــه stack نگاه کنید تا آخرین فراخوانی babe خود را مکان‌یابی کنید (اگر فــوراً آن را نشـان نـداده. فقـط بـه تعیین آدرس محــل آن بپردازیــد، مثـلاً روی ()Getwindowtext یـا دیسـکومت BPRW را انجـام دهید.(بسیار مفید) و سپس جستجو کنید و همچنان جستجوی stack را از نو بگیرید. سرانجام ایــن کار ناموفق خواهد مــاند. سـپس بدنبـال ورودی خـود در در حافظـه بگردیـد (در انتخـاب کننـده

30:01ffffffff، طبق معمول) و دامنـه نقطـه توقـف در آن بـا Readwrtie، و سـپس stack،stack

stack ... تا اینکه لیست واقعی فراخوانی‌هایی که در مقام محافظت از babe شما برمی‌آیـد را پیـدا

کنید.

```
USER(19) at 073F:124C [?] through 073F:1239
CTL3D(02) at 2C3F:0D53 [?] through 2C3F:0D53
DISKOMAT(01) at 2C97:20B9 [?] through 2C97:20B9
DISKOMAT(01) at 2C97:3D94 [?] through 2C97:3D94
DISKOMAT(01) at 2C97:49E2 [?] through 2C97:4918
DISKOMAT(04) at 2C7F:EA20 [?] through 2C7F:EA20
USER(01) at 04A7:19BE [?] through USER!GETWINDOWTEXT
== CTL3D(02) at 2C3F:24BE [?] through 04A7:3A3C


 Beautiful stack fishing! Do immediately a BPX on babe:EA20.
2C7F:EA35 9A25ABA704   CALL   USER!GETWINDOWTEXT
2C7F:EA3A 8D46AE       LEA    AX,[BP-52]     ;load ptr "+ORC+ORC"
2C7F:EA3D 16           PUSH   SS   ;save pointer segment
2C7F:EA3E 50           PUSH   AX   ;save pointer offset
2C7F:EA3F 9A768D872C   CALL   2C87:8D76; get strlen "ORC+ORC"
2C7F:EA44 83C404       ADD    SP,+04
2C7F:EA47 3D2800       CMP    AX,0028
2C7F:EA4A 762C         JBE    EA78
...
2C7F:EA97 8D46AE       LEA    AX,[BP-52]     ;load ptr "+ORC+ORC"
2C7F:EA9A 16           PUSH   SS     ;various algors on input□
2C7F:EA9B 50           PUSH   AX     ;follow here, we do not
...                               ;need to care
2C7F:EAB2 0F851101     JNE    EBC7
2C7F:EAB6 8D8E5CFF     LEA    CX,[BP+FF5C]  ;ptr "12121212"
2C7F:EABA 16           PUSH   SS
2C7F:EABB 51           PUSH   CX
2C7F:EABC 9A768D872C   CALL   2C87:8D76 ;get strlen "12121212"
2C7F:EAC1 83C404       ADD    SP,+04
2C7F:EAC4 50           PUSH   AX
2C7F:EAC5 8D865CFF     LEA    AX,[BP+FF5C] ;ptr "12121212" HERE!
2C7F:EAC9 16           PUSH   SS
2C7F:EACA 50           PUSH   AX
```

جستجوی زیبای stack! بلافاصله یک BPX روی babe انجام دهید: EA 20

جستجوی زیبای لیست! بلافاصله یک BPX روی babe انجام دهید: EA 20

جستجوی زیبای لیست! بلافاصله یک BPX روی babe انجام دهید: EA 20

جستجوی زیبای لیست! بلافاصله یک BPX روی babe انجام دهید: EA 20

... و غیره، الگوی مختلفی روی ورودی وجود دارند.

بسیار خوب، کافی است: حالا مسلماً بدنبال کدی هستیم که رشته اعداد را بصورت الگوریتم نشـان داده و سپس، یک جایی، باید دست به یک مقایسه وحشتناک بزنید که در آن، پسرهای خـوب را از دیوانه‌های بد جدا می‌کنید.

میتوانید امتحان کنید، بگردید و جستجو کنید.

اما، اکنون همان «لحظه اسرارآمیز» اکو است! ما آن را می‌شناسیم و حسش می‌کنیم! اکو بـاید در هـر جایی وجود داشته باشد... چگونه میتوانیم آن را بیابیم؟ جستجوی 12121212 در حافظـه، حداقـل ده محل مختلف را پیدا می‌کند.

:S 30:0 1fffffff '12121212'

این الگو در 0030:8145AD6A یافت می‌شود.

آیا باید، بدنبال رویدادهای (وقایع) رشته '12121212' بگردیم، یعنـی آغـاز شـدن آن بـا دو تـا در ۸۰۰۰۰۰۰۰، روبرداری 0x90 در اطراف آن... تا Echo را پیدا کنیم؟ میتوانیم، و باید اینکار صـورت گیرد، اما این zen نیست....این خسته کننده است.

در دیگر محافظت کننده‌ها، این محلها را میتوان به همین منظور تکثـیر کـرد تـا مـانع کراکـر شـود. شیوه‌های دیگری هم بـاید وجـود داشـته باشـد... بلـه ! راه سـریعتری هـم وجـود دارد... آخریـن بارگذاری رشته ورودی عددی در کد، بـه منظـور کراکینـگ، بـارگذاری سـمت راسـت، می‌باشـد، محافظتهایCOZ، غالباً از این الگو پیروی می‌کند، (به خاطر داشته باشید: ما در یک قسمت سـخت استاک کد قرار داریم... اگر بخواهید بیشتر کراک کنید، پیشنهاد می‌کنیم مقالات مناسبی در مورد کار استاک، ترفندهای استاک و نیروهای رمزی آن از طریق پردازشگرهای Intel بخوانید).

بارگذاری نام استرینگ ـ شمارش نام استرینگ من

بارگذاری نام استرینگ ـ تغییر نام استرینگ

بارگذاری نام استرینگ ـ شمارش نام استرینگ

بارگذاری نام استرینگ

ECHO باید در این قسمت باشد.

تغییر کد استینگ

ECHO باید در این قسمت باشد.

نام استرینگ تغییر یافته را با کد استرینگ تغییر یافته مقایسه کنید.

این یعنی،در خط

```
2C7F:EAC5 8D865CFF   LEA  AX,[BP+FF5C]  ;ptr "12121212"
```

شما در جایی،اکو خواهید داشت... فقط حافظه اطراف اشاره‌گر را روبرداری کنید [BP+FF5C]

```
:d 2c5f:61e8   ;these numbers will differ in your computer
02 62 2F 06 02 00 26 2E-A3 4E A3 4E 01 00 38 30  .b/...&..N.N..80
33 37 2D 36 34 36 2D 33-38 33 36 00 01 06 02 00  37-646-3836.....
2F 06 75 62 C3 2E B7 04-F2 24 2F 06 CE 6E 2F 06  /.ub.....$/..n/.
49 00 5A 00 01 00-04 2C 2F 06 AE 24 36 62 00 00  I.Z......,/..$6b
74 62 7A 2E B7 04 36 62-01 00 C2 62 2F 2C 26 2E  tbz...6b...b/,&.
03 01 BA 0F AE 24 5F 02-C9 01 5E 02 BA 01 5F 02  .....$_...^..._.
31 32 31 32 31 32 31 32-00 0C 00 BC 02 00 00 00  12121212........
00 49 00 BA 0F-AE 24 F2 24 2F 06 00 00 00 00 00  ....I....$.$/...
AF 17 00 E2 5F-7A 62 FE FF 79 1B BA 0F 00 00 00  ........_zb..y...
96 0B 01 00 02 4E 00-37 01 8A 62 D2 0F 8F 17 00  .....N..7..b....
2F 06 00 37 01-98 62 20 10 16 03 2F 06 00 00 00  /.....7..b ..../.
C2 62 2B 4F 52 43 2B 4F-52 43 00 0D AE 24 2F 06  .b+ORC+ORC......
```

به این رونوشت، نگاه کنید : همه افراد اینجا هستند! اشـــاره‌گرهـــای اســتاک در وســط در اســترینگ 12121212 قرار دارند. یا وجود بایتهای 0X50 قبل از آن، میتوانید ECHO قبلی مناسب ما را پیـــدا کنید. و با بایتهای 0X50 پس از آن میتوانید عنوان خود را مشاهده کنید: اینجا ORC+ORC+ این کراک شده است! کد مورد نظر برای ORC+ORC من، 8037-646-3836 است.

حال، عمل جایگزینی خود را آغاز کنید: اگر مشتاقید تا عمل کراکینگ را یاد بگیرید:

چیزی ثبت نکنید و یک کد جدید برای عنوان خود پیدا کنید. برای هر نام دیگری که عنـــوان خـــود شماست، از اعداد ترتیبی «استفاده نکنید. » این کار، دزدی شرم آوری است، نه عمل کراکینگ.

و در وب، تنبیه خواهید شد، من این کار را غیرقانونی میدانم، و از جیب‌برهای احمق متنفرم.

– هر دو الگوریتم کدگذاری را مطالعه کنید، یکی برای نام ورودی و دیگری بـرای شـماره ورودی، این برای جلسات کراکینگ شما بسیار مفید است.

Compare را پیدا کنید، یعنی، کدی کـه دو گروه "good guy" و "bad cracker" را از هـم مجـزا می‌کند.

برای این محافظت، یک کراک واقعی بوجود آورید، تا همه فکر کنند که اسـتحقاق آن را داریـد، بـا هر نام و هر شماره رمزی که دوست دارید،

(Cracking Snap 32)

Snap 32 (SNAP32.EXE 356.352 bytes, 24/11/95, Version 2.54,

یک برنامه اشتراک افزار SNAP SHOT برای ویندوز ۹۵ است که به استفاده کننده اجازه میدهد تـا تمام صفحه، بخشهایی از آن، یا یک پنجره جداگانه را نگه دارد. این تسـت (امتحـان) خیلـی رایـج است. قبل از آنکه برنامه را بخرید، که محدود به ۳۰ روز استفاده می‌باشد.

میتوانید آن را هر جا روی وب پیدا کنید. اگر بلد نیستید وب را جستجو کنید، در پایان ایـن درس، روش صحیح پیدا کردن فایلهایی که در شبکه به آن نیاز دارید را پیدا نموده بـه طـور اتومـاتیک‌وار برای شما بصورت E-mail فرستاده می‌شود.

(چیزهایی باید در این درس یاد بگیرید: سرچینگ (جستجو)، این کـار از کراکینـگ مـهمتر اسـت!)

snap32 چندان جالب نیست، اما محافظت از آن عبارتست از : به منظور جلـوگیـری از کراکرهـای اتفاقی، استرینگها را مقایسه نمی‌کند، آن، یک حاصل جمع رمزی را با حاصل جمع رمـزی دیـگری (از نام استرینگ) مقایسه می‌کند. و :

حاصل جمعهای رمزی را درGDI جمع کنید، نه در خود کد؛

برای ارزیابی ورودی، به جای کد ساده، به جدول مراجعه کنید.

دستکاری رمزی. شماره ورودی را با دستگاه رمزی نام ورودی مقایسه کنید.

روش کراکینگ برای بییشتر برنامه‌های ویندوز، بسیار ساده و جذاب و قابل فهم است:

۱- به نام کودک خود و انتخابگر صف مراجعه کنید.

تکلیف: این فرمان winice95 است که باید پس از پیدا کردن snap32 آن را تایپ کنید و گواهی ورود به nag windows را بگیرید.

Tasknmae

```
TaskName SS:SP     StckTp   StckBt  StckLw TaskDB  Hqueue  Events
Snap32   0000:0000 006 AC000 006B0000       270E    D27    0000
```

بسیار خوب،babe، snap32 است، صف آن 0xD27 است، TakDB آن، 0x27OE می‌باشد.

۲) به مدولهای babe خود مراجعه کنید:

```
:map32 snap32       ;Your command
Owner     Obj Name  Obj#  Address         Size      Type
SNAP32    .text     0001  0137:00401000  00043000  CODE  RO
SNAP32    .rdata    0002  013F:00444000  00002E00  IDATA RO
SNAP32    .data     0003  013F:00447000  00009000  IDATA RW
SNAP32    .idata    0004  013F:00471000  00001C00  IDATA RW

SNAP32    .rsrc     0005  013F:00473000  00001600  IDATA RO
SNAP32    .reloc    0006  013F:00475000  00004C00  IDATA RO
```

خوب، کد مورد نظر در انتخابگر، ۱۳۷ (طبق معمول) است، و شما ٤۳۰۰۰ بایت کد از ٤۰۱۰۰۰ تا ٤۰۱۰۰۰+٤۳۰۰۰ دارید؛ Readonly, Readwirte, DATA (یعنی داده، خواندن و نوشتن، فقط خواندنی) در انتخابگر 13F وجود دارند.

۳) به عنوان محافظت NAG window مراجعه کنید.

```
:hwnd snap32             ;Your command
Window Handle  Hqueue  SZ  Qowner   Class Name Window Procedure
 0350(1)       0D27    32  SNAP32   #02071     144F:0560
  0354(2)      0D27    32  SNAP32   #02071     17CF:102E
  ... and many more windows that we do not care of.
```

... و پنجره‌های بیشتری که چندان به آنها توجهی نداریم.

بسیار خوب، برای اهداف کراکینگ خود، عنوان 0x350 وجود دارد. اکثر مواقع، میخواهید کـه nag window که کراک میکنید اولین window در لیست‌گیری hwnd باشد. به شماره‌های داخل پرانـــتز نگاه کنید که هرکدام یک عنوان هستند: (۱) یک مادر است. (۲) بچه‌ها هستند. خیلی وقتها، در زیر classname (نام گروه) بـه واژه edit برمیخوریـد (آن را قبـل از کراکینـگ ویـن فورمـت مشـاهده می‌کنید)... در بعضی مواقع، آدرس کد window procedure (کار پنجره) در یک لیست بیش از ۲۰ تایی، برای یک یا دو پنجره فرق می‌کند.

٤) پیغام نقطه توقف گتکس wm (یا هر wm دیگری که برای مکان‌یـابی کـد کودکـان بـه نظرمـان می‌رسد).

مکان یابی کد، در کراکنیگ پنجره‌ها بسیار مهم است... این osابلهانه، کد را حرکت میدهد، داده‌ها و استاک، خارج شده و همواره صفحه‌ها را وارد آن می‌کند. بنابراین به سراغ قسـمتهای Invalid (بـی ارزش)، خواهید رفت بدون مکان‌یابی صحیح.

نکات مکان‌یابی بطور کلی عبارتند از:

```
BMSG xxxx WM_GETTEXT    (good for passwords)
BMSG xxxx WM_COMMAND    (good fro OK buttons)
BPRW *your babe* TW     (good for tracking)
u USER!GETWINDOWTEXT    (u and then BPX inside the code)

u GETDLGITEM            (for the Hwnd of an Item inside a
                        Dialog Box)
CSIP NOT GDI            (if you have too many interferences)
u USER!SHOWWINDOW       (bpx with counter occurrence to get to
                        the "right" window)
u GETSYSTEMTIME         (for "time-crippled" software)
```

و نکات مکان یابی دیگری هم وجود دارد که بعداً یاد میگیرید. اگر واقعاُ ناچارید مکان‌یـابی کنیـد، یک BMSG xxxx WM MOVE انجام داده و ســپس nag window را جابجـا کنیـد، ایـن کـار همیشگی است: اجازه دهید ادامه دهیم:

فرمان شما؛ bsmg 350 wm gettext

پس کد، آماده مکان یابی است.

۵- برنامه را در نقطه توقف اجرا کنید.

فرمان شما برای خروج winice و اجرای آن CTRL+D تا زمانیکه در نقطه توقف مشاهده شود.

۵) محل داده را برای رشته ورودی خود جستجو کنید (٤ گیگابایت از ۰:۳۰ ... به خاطر داشته باشید که dataها (داده‌ها) همیشه در ۰:۳۰ الی 30:ffffffff هستند و کد «همیشه» در ۰:۲۸ الی ffffffff : ۲۸ می‌باشد) . در اکثر محافظتها، رشته شماره ثبت باید رشته نام استفاده کننده، که نمیتواند محدود باشد را بدین منظور هماهنگ(مچ)کند که به استفاده کننده اجازه دهد تا نام احمقانه‌ای که دوست دارند انتخاب کند. بخی از محافظتها به سمبلهای مشخصی در رشته نام استفاده کننده نیاز دارند. دراین اتفاقات نادر،برای نام استفاده کننده میتوان آنچه را انتخاب نمود که ما در این قسمت با یک رشته شماره‌های مثبت انجام میدهیم. یک نکته را باید به خاطر بسپارید: همواره با محافظتی شروع کنید که بدنبال شماره شما بگردد، در صورت نیاز، محاظتی را کراک کنید که نام شما را پیدا کند. حال بیایی جستجو کنیم:

فرمان :S 30:0 1ffffffff 12121212;

الگوی یافت شده در 0030:80308612

80000000 مناسب است با وجود ویدئوها، آینه‌ها و BIOS دوره پایین‌تر، بیش از(حدود C000000) یک زباله‌دانی OS خواهید داشت. نکته‌ای که باید به خاطر بسپارید اینست که : همیشه در اولین آدرسهای 80000000 تحقیق کنید.

۷) نقطه توقف روی دامنه حافظه در این استرینگ

بدین ترتیب: یک watch window dex3 es:di تهیه کنید، بزودی می‌بینید که چگونه یک چنین پنجره اتوماتیکی چقدر در کراکینگ کمله رمز مفید است.

```
:bpr 30:80308612 30:80308612+8 RW  ;Your command
```

خوب، حالا شروع به جستجوی قسمتهای مربوط به کد می‌کنیم. به خاطر داشته باشید که بـاید هـر کپی از استرینگی که محافظت آن را بوجود آورده را در یک نقطه متوقف سازید. روال عـادی یـک کپی خاص، که غالباً در طرحهای محافظت کپی ویندوز بکار میرود، درون طرح وجود دارد.

```
0117:9E8E 66C1E902      SHR     ECX,02
0117:9E92 F36766A5      REPZ MOVSD      ;makes a copy in es:di
0117:9E96 6659          POP     ECX
0117:9E98 6683E103      AND     ECX,+03
0117:9E9C F367A4        REPZ MOVSB
0117:9E9F 33D2          XOR     DX,DX
```

در حقیقت، این تکه از کپی‌برداری کد، غالباً برای تأیید کلمه رمزی بکار میرود که گاهی اوقات نیاز دارید تا آن را در 0117:9e62، bpx کنید تا ترتیب صحیح استاک را بدست آورید...

اما میخواهیم بدون چنین ترفندهای کوچکی به کارمان ا دامه دهیم. فقط تمام کپیهایی که محـافظت میسازد را در BPRing نگهدارید (نقطه توقف در دامنه حافظه).

۸) اجازه دهید تا برنامه BABE اجرا شود. این در یک نقطه، تمـام دسـتکاریهای اسـترینگ ورودی شما را متوقف خواهد ساخت. یکی از آنها واقعاً جادو می‌کند.

۱-۸) مرحله ارزیابی

برنامه‌های زیادی وجود دارد که ورودیهای شما را کنترل و ارزیابی می‌کند. یکی از رایج‌تریـن ایـن برنامه‌ها اینست که شماره‌های شما واقعاً شماره‌ها باشند.

یعنی، در دامنه 0x30-0x39 باشند. این معمولاً بدین صورت انجام می‌شود.

```
CMP  EAX,+30
JB   no_number
CMP  EAX,+39
JA   no_number
```

اغلب اوقات، محافظت‌کننده‌ها، از جدولها استفاده می‌کنند. خود شماره، بعنوان یـک اشـاره‌گر بـای جدول حاضر و آماده‌ای استفاده می‌شود که جادوی مربوطه بعنوان محافظت استفاده می‌شود. تصور

کنید که شماره ٤ در ورودی شما به کدی اشاره کند که برنامه ارزیابی شما را فوراً به خارج بفرستد... یا تصور کنید که شماره ۷. اگر در ورودی شما مشاهده شود، کدی را جستجو کند که کل برنامه را از دیسک هارد شما دور کند. کراکر ابله نمیداند که هرگز نخواهد فهمید که نباید از شماره ٤ استفاده کند...و مسلماً از شماره ۷ هم همینطور! بعداًیاد خواهد گرفت... بله جدولها برای چنین ترفندهای پردردسری استفاده میشوند. در اینجا، کد مربوطه، برای بخش ارزیابی محافظت ما عبارت است از :

```
0137:4364AE 8A16       MOV     DL,[ESI] ;load license number
0137:4364B0 33C0       XOR     EAX,EAX  ;zero AX
0137:4364B2 668B0451   MOV     AX,[ECX+2*EDX] ;look table for 84
0137:4364B6 83E008     AND     EAX,+08  ;OK if AND'S TO zero
0137:4364B9 85C0       TEST    EAX,EAX  ;and therefore
0137:4364BB 7403       JZ      004364C0 ;go on
0137:4364BD 46         INC     ESI      ; ready for next number
0137:4364BE EBCD       JMP     0043648D
:strip_-_&_+_signs
0137:4364C0 33DB       XOR     EBX,EBX  ;clean BX
0137:4364C2 8A1E       MOV     BL,[ESI] ;load license number
0137:4364C4 46         INC     ESI      ;ready for next
0137:4364C5 8BFB       MOV     EDI,EBX  ;save copy
0137:4364C7 83FB2D     CMP     EBX,+2D  ;is it a "-"?
0137:4364CA 7405       JZ      004364D1
0137:4364CC 83FB2B     CMP     EBX,+2B  ;is it a "+"?
```

۲-۸) دستکاری ٠جمع کردن شماره‌های سحر آمیز)

حاصلجمع نقاط توقف شما در دامنه حافظه برای ظهور رشته "12121212" باعث تعجب شما خواهد شد،اثرات بصری نامطلوبی در تصاویر ایجاد کرده، و بخش بعدی کد را وارد می‌کنید. (دقت کنید که این بخش از محافظت در درون GDI قرار دارد، در درون انتخابگر کد snap 32) :

```
0557:11BD 33C0         XOR  EAX,EAX        ;zero AX
0557:11BF 66648B06     MOV  AX,FS:[ESI]    ;load number
0557:11C3 83C602       ADD  ESI,+02        ;point to next
0557:11C6 66833C4700   CMP  WORD PTR [EDI+2*EAX],+00
0557:11CB 0F8424010000 JE   000012F5
0557:11D1 668B0442     MOV  AX,[EDX+2*EAX] ;load from magic table
0557:11D5 03D8         ADD  EBX,EAX        ;save sum in EBX
0557:11D7 49           DEC  ECX            ;till we are done
0557:11D8 75E5         JNZ  000011BF       ;loop along
```

جالب است، اینطور نیست؟ محافظت در این برنامه GDI مورد استفاده قرار گرفته تا حاصل جمعی خلق کند که به شماره‌های ورودی شما بستگی دارد. حالا به کراک نزدیک شده‌ایم...

میتوانید آن را حس کنید؟ اگر نمیتوانید، خود را برای خوردن یک ودکای مارتینی آماده کنید.

روش درست آن اینگونه است:

یک لیوان های بال بردارید

چند تکه یخ درون آن بیندازید (۲ الی ۳تا)

ماریتنی درای به آن اضافه کنید (از مارتینی و رزی) . یک سوم آن را پر کنید.

موسکوسکاجا و ودکا به آن اضافه کنید (و ودکای واقعی) . دو سوم آن را پر کنید.

کمی اسانس لیمو به آن اضافه کنید (از مالتایا جنوب فرانسه).

زیتون سبز به آن اضافه کنید (از ایتالیا یا اسرائیل)

داروی تقویتی هندی Schweppes به آن اضافه کنید تا لبش را پر کنید.

آرام و ریلکس بنشینید، کم کم،جرعه جرعه بنوشید و احساس کنید که طرح محافظتی که کراک می‌کنید «حرکت می‌کند»... مثل یک جریان آب... یک جذر و مد آهسته . اگر باور ندارید، امتحان کنید. حالا باید بگردیم و ببینیم که کجا محافظت، حاصلضرب سحرآمیز را ذخیره می‌کند(و حالا به کد خودتان یعنی snap32 دقت کنید، این قسمت واقعی محافظت است.

۳-۸) اختفای مضحک حاصلح جمع رمزی

```
0137:40437E 83C404       ADD     ESP,+04
0137:404381 8B4DE8       MOV     ECX,[EBP-18]
0137:404384 8945F0       MOV     [EBP-10],EAX  ;***HERE!***
0137:404387 68FF000000   PUSH    000000FF
0137:40438C 8D8574FBFFFF LEA     EAX,[EBP+FFFFFB74] ;load string
0137:404392 50    PUSH   EAX                       ;push it
0137:404393 E886410100   CALL    0041851E         ;manipulate
0137:404398 8D8574FBFFFF LEA     EAX,[EBP+FFFFFB74] ;load string
0137:40439E 50    PUSH   EAX                       ;push it
0137:40439F E88C210300   CALL    00436530         ;manipulate
```

همانطور که مشاهده میکنید، محافظت. کار بسیار ساده‌ای است : این حاصل جمع رمزی قبل از دستکاری استرینگ ورودی ، فقط در دو خط مخفی می‌شود. ما در اینجا، به روش صحیحی، محل

۱۳۷:٤٠٤۳۸٤ را پیدا کرده‌ایم، از طریق یافتن رشته‌ای که در GDI دستکاری شده است اما در واقع، میتوان آن را خیلی سریع با چک کردن آنچه پیرامون تمام دستکاریهای استرینگ داده رخ میدهد. پیدا کرد. آیا واقعاً لازم است تا بنبال تمام دستکاریهای شماره ثبت خود و همچنین تمام دستکاریهای نام استفاده کننده خود بگردیم؟ خیر، در مجموع خیر:

فقط باید یک BPR در محل استاک مشخص کنیم که محافظت، حاصل جمع [EBP-10] را مخفی نموده و خواهیم دید که چه اتفاقی می‌افتد: ۹۰ درصد از این محافظتها فقط دو حاصل جمع بودجود می‌آورند، یک حاصل جمع از نام استفاده کننده شما و یک حاصل جمع از شماره ثبت شما... در جاهایی، مقایسه‌ای صورت خواهد گرفت که باید از این محل استفاده کند (یا یک نسخه از آن... خواهیم دید).

٤-۸) مقایسه جادوها با رشته ورودی دوتایی

نقطه توقف در دامنه حافظه در محل حاصل جمع [EBP-10] که در کد قبلی مشاهده نمودید و در این قسمت از کد، قرار خواهید گرفت.

```
0137:404412 E82F050000   CALL 00404946
0137:404417 83C40C       ADD  ESP,+0C
0137:40441A 3B45F0       CMP  EAX,[EBP-10] ;comp AX & magicsum
0137:40441D 740F         JZ   0040442E
0137:40441F 68C0874400   PUSH 004487C0
0137:404424 E8149E0000   CALL 0040E23D
0137:404429 83C404       ADD  ESP,+04
0137:40442C EB5B         JMP  00404489
0137:40442E 893DA0714400 MOV  [004471A0],EDI
0137:404434 85FF         TEST EDI,EDI
```

این نتجه کار شماست! ما مقایسه بین شماره رمزی نام استفاده کننده (برای استرینگ ORC+ORC+) من که در اینجا عبارت است از 0X7C25621B در AX (نیازی نیست بدانیم که چگونه این در اینجا آمده است... چیز بی ربطی است!) و شماره اجازه استفاده 12121212(که رمز آن عبارت است از 0X00BF47C)ذخیره شده در (اشاره گر۱۰) را پیدا کردیم.

حالا چگونه شماره ورودی صحیح برای ORC+ORC+ را پیدا کنیم؟ خوب، ساده است... شماره رمز، باید یکجور باشد... بنابراین :

```
Cracked=Dec(0x7C25621B)
Cracked=2082824731
```

تمام آن، همین بود. Snap 32 قبلی، کراک شده است. حال، میتوانید به منظرو توزیع این برنامه بدون محافظت ساده آن، یک کراک تهیه کنید. کاربردهای خوب کراک شده باید آزادانه در اختیار افرادی قرار گیرد که واقعاً به آنها نیاز داشته و استطاعت مالی برای خرید آن ندارند.

فراموش نکنید که در چنین جامعه غیرقابل تحملی، ۵ درصد از شهروندان، خودشان صاحب ٥٦ درصد سرمایه صنعتی بوده و ٦٣ درصد از دستگاههای تبلیغات، سرنوشت میلیونها برده‌ای را به طرز مؤثری در وضعیت بهتری قرار میدهند که با تماشای تلویزیون شناخته میشوند. بنابراین، کاربردها را کراک نموده و به افرادی که مورد توجه‌تان هستند و آنهایی که واقعاً به آن نیاز دارند بدهید... فقط برای هر کسی توضیح دهید که چگونه آن را انجام میدهید. .. این واقعاً به آنها کمک می‌کند: دادن اطلاعات، نه وسایل. از عنوان و کدهای من برای کراک کردن این برنامه استفاده نکنید، از عنوان و کدهای خودتان استفاده کنید. من میتوانم عنوان و کد خودم را فقط بعنوان یک ابزار کمکی برای درس کراکینگ در اختیار شما بگذارم. من به اندازه کافی، شما را راهنمایی کرده‌ام... دزدها، از کدهای دیگران استفاده می‌کنند، نه کراکرها. شما هم جزو کراکر هستید! این را همواره به خاطر بسپارید، مجدداً جستجو کنید، بدقت کراک کنید و به اینکار ادامه دهید.

چگونه بدون حرکت هست، در اینترنت به دنبال فایلها بگردیم.

بسیار حیرت انگیز است : بیشتر افراد، در اینترنت دنبال چیزی می‌گردند بدون آنکه چگونگی استفاده از وب را بدانند. من از خودگذشتگی می‌کنم و برایتان توضیح میدهم که چگونه بدنبال خیلی از نمونه‌های snap32 بگردید، کودکی که ما در این درس کراک کردیم.

۱- از این لیست یک archie انتخاب کنید (دیگر برایتان توضیح نمیدهم که archie چیست، خودتان باید بلد باشید... اگر بلد نیستید واقعاً باید خجالت بکشید):

```
ac.at      131.130.1.23        Austria
     archie.belnet.be          193.190.248.18      Belgium
     archie.funet.fi           128.214.6.102       Finland
     archie.univ-rennes1.fr    129.20.254.2        France
     archie.th-darmstadt.de    130.83.22.1         Germany
     archie.ac.il              132.65.16.8         Israel
     archie.unipi.it           131.114.21.10       Italy
     archie.uninett.no         128.39.2.20         Norway
```

۲- پیام را به archie خود E-mail کنید:

```
     To:       archie.univie.ac.at (for instance)
     Subject:                      (nothing on this field)
     Body:     set search sub      (substrings too)
               set maxhits 140     (max 140 hits)
               set maxhitspm 9     (not the same file all over)
               find snap32         (we want this)
```

۳- پس از چند لحظه، (در هر Email). پاسخ خود را خواهید یافت: در اینجا، پاسخ archie اتریشی عبارت است از:

```
Host ftp.wu-wien.ac.at    (137.208.8.6)
Last updated 17:48  9 Aug 1995
Location: /pub/systems/windows.32/misc
FILE    -rw-r-----  128957 bytes  15:59 16 Jun 1995  snap32.zip
Host space.mit.edu    (18.75.0.10)
Last updated 00:45  4 Mar 1996
 Location: /pub/mydir
  FILE    -rw-r--r--  407040 bytes  11:55 28 Nov 1995  snap32.exe
```

فایل خود را ftpmail کنید (نگاه کردن به فایلها یا لیستهای کامپیوتری، درست نیست: (پردرد سر و غیر موجه). مجدداً، من لازم نمیبینم که برایتان توضیح دهم که خدمات‌رسان ftpmail چیست:

خودتان باید آن را یاد بگیرید... یکی از ftpmailهای این لیست را انتخاب کنید.

(تعداد بیشتری وجود دارد...خودتان یاد خواهید گرفت):

```
   bitftp@vm.gmd.de                          (Germany)
   ftpmail@ieunet.ie                         (Ireland)
   bitftp@plearn.edu.pl                      (Poland)
   ftpmail@ftp.sun.ac.za                     (South Africa)
   ftpmail@ftp.sunet.se                      (Sweden)
   ftpmail@ftp.luth.se                       (Sweden)
   ftpmail@src.doc.ic.ac.uk                  (United Kingdom)

To:       ftpmail@ftp.sun.ac.za.   (for instance)
Subject:                           (leave blank)
Body:     open space.mit.edu       (the last occurrence that
```

```
                                the archie sent)
cd/pub/mydir                    (get the correct subdir)
bin                             (prepare for BINARY)
get snap32.exe                  (I want this)
quit                            (bye)
```

۵- خدمات رسان ftpmail ، از شما یک دریافتی میخواهد:

پاسخ ftp mail....

ftpmail، کار زیر را از شما دریافت کرده است:

```
reply-to +ORC
open space.mit.edu +ORC@now.here
get snap32.exe
```

Ftpmail، کار شما را بدین صورت ردیف کرده: 1834131827.5514

اولویت شما عدد ۱ است (۰: بیشترین، ۹: کمترین)

درخواست sunsite.doc.ic.ac.uk قبل از کارهای دیگر انجام خواهد شد.

در این صف، قبل از این کار، ١٤ کار دیگر در رأس قرار دارند.

٤ تا کنترل کننده ftpmail نیز در دسترس هستند.

فرستادن پیام به ftpmail فقط شامل این پیام خواهد بود:

Delete 1834131821.5514

پس از چند لحظه، پیام دوم را، دریافت می‌کنید، در فایل شما کدگذاری نشده است...

تمام کارها انجام شده است،بله آقا! لزومی ندارد در www، وقت را تلف کنید، لزومی ندارد از یک جانک سایت به جانک سایت بعدی بروید یا ساعتها منتظر بمانید تا فایلی از خدمات‌رسان متغیر اطلاعات را منتقل کند! اتلاف وقت در زندگیتان، که می‌توانید از آن برای خواندن اشعار استفاده کنید، عشق را بسازید، به ستاره‌ها نگاه کنید، بین جزایر Aegean به آرامی قایق سواری کنید یا یک جلسه کراکینگ خوبی را آغاز کنید. دیگر اتلاف وقت چه معنایی دارد، وقتی ماشینها یا دستگاهها قادرند تا جستجوهایی که نیاز دارید را بهتر، سریعتر از شما بطور مفیدی انجام دهند.

بله! شما قادرید هرچه که میخواهید روی web بیابید، بدون اینکـه بـه فراهـم کننـده اینـترنت پـول هنگفتی بپردازید... اینطور نیست!