

Criando scanner para Oracle vulneráveis a exploits do Metasploit – Inj3cti0n P4ck3t.



Inj3cti0n P4ck3t

São Paulo
2010

Nome: Fernando Henrique Mengali de Souza

Contato: fer_henrick@hotmail.com

Linguagem de Programação: Perl (Practical Extraction and Report Language)

0x01 Introdução

O Metasploit Framework possui um arsenal de ferramentas e funcionalidades para desenvolver exploits, scanear e explorar falhas em diversos tipos de softwares e serviços, como: FTP, SMTP, POP, VNC, IMAP, IIS, APACHE e outros serviços.

Qualquer usuário pode utilizar o Metasploit Framework para atacar um sistema operacional que executa um determinado serviço. Mas a diferença em usar o Metasploit, está na conhecimento e no sucesso de explorar falhas de segurança.

No artigo, descrevo como desenvolver um scanner na linguagem de programação PERL que identifica através do banner de serviço o banco de dados Oracle que são vulneráveis aos exploits do Metasploit Framework.

O scanner é simples, mas o usuário pode implementar funções sofisticadas se quiser.

0x02 Versões do banco dados Oracle vulneráveis aos exploits do Metasploit Framework

Nesse artigo, vamos criar um scanner que identifique três tipos de vulnerabilidades de buffer overflow no banco de dados Oracle. As três vulnerabilidades são exploradas com os seguintes exploits do Metasploit Framework:

- **Oracle 8i TNS Listener (ARGUMENTS) Buffer Overflow**
Exploit: tns_auth_sesskey
- **Oracle TNS Listener AUTH_SESSKEY Buffer Overflow**
Exploit: tns_auth_sesskey
- **Oracle TNS Listener SERVICE_NAME Buffer Overflow**
Exploit: tns_service_name

As falhas encontradas no banco de dados Oracle são antigas, encontradas no ano de 2001. Porém, dificilmente encontram-se servidores com banco de dados Oracle 8i, mas os exploits continuam eficazes para explorar as falhas.

0x03 Softwares para testar o Scanner

Active Perl (Interpretador Perl para Windows)

Download: <http://downloads.activestate.com/ActivePerl/releases/5.12.2.1202/ActivePerl-5.12.2.1202-MSWin32-x86-293621.msi>

Metasploit Framework 3.3

Download: <http://www.metasploit.com/releases/framework-3.4.1.exe>

Oracle 8i Version 8.1.7.0.0

Download: Não disponível.

Oracle 8i Version 10.2.0.1.0

Download: Não disponível.

Oracle 8i Version 10.2.0.4.0

Download: Não disponível.

Não foi possível apresentar ao leitor a exploração da falha ou um ambiente teste.

Porém, o scanner capturou o banner de outros tipos de banco de dados Oracle.

0x04 Desenvolvendo um Scanner para identificar Oracle vulneráveis a exploits do Metasploit

Desenvolveremos um scanner na linguagem de programação PERL para identificar banco de dados Oracle vulneráveis aos exploits do Metasploit Framework.

Faça download do interpretador Perl e instale em sua máquina.

Se você usa sistema operacional Linux, o caminho dos scripts em Perl será `usr/bin/perl`, como apresento abaixo.

```
#!/usr/bin/perl
```

Se você fez download do Active Perl e instalou no sistema operacional Windows, o caminho para inserir o script que estamos desenvolvendo

será c:\perl\bin.

```
#!c:\Perl\Bin
```

Módulos implementados no Scanner MetasploitScanOracle.pl

Os módulos IO::Socket e IO::Socket::INET, são responsáveis pela conexão com nosso host alvo ou servidor.

```
use IO::Socket;  
use IO::Socket::INET;
```

A expressão de condição "IF", verifica se alguma lista com endereços de sites ou IPs foram informados ao programa. Se a lista não foi informada, o scanner verifica o sistema operacional na variável "\$sis" e executa o comando "clear" se for Linux ou "cls" se for Windows.

Depois executa a linha "print q { ... }" apresentando como usar o programa.

A linha "exit()", finaliza o programa em execução.

```
if (!$ARGV[0]) {  
  
    $sis="$^O";if ($sis eq windows){ $cmd="clear";} else { $cmd="cls"; }  
    system("$cmd");
```

```
print q {
```

```
    Code desenvolvido por: Inj3cti0n P4ck3t  
    Nome: Fernando Henrique Mengali de Souza  
    E-mail to contact: fer_henrick@hotmail.com
```

Modo de uso: perl MetasploitScanOracle.pl listaDelps_ou_ListaDeSites.txt

```
    };  
  
    exit();  
  
}
```

A linha "open", abriu um arquivo .txt informado pelo usuário. Caso a lista de sites ou IPs não for informada, haverá uma mensagem de erro:

"Não foi possível abrir o arquivo".

```
open( SITE, "< $ARGV[0]" ) or die( "Nao foi possível abrir o arquivo: $!" );
```

Se o programa conseguir abrir o arquivo com site e IPs será atribuído ao array

ou vetor "@array".

Depois atribuímos "<SITE>", carregando a lista de sites no array.

A variável "\$numero" possui o último array para ser usado no "for".

```
our @array = <SITE>;
```

```
$numero = $#array;
```

Inicia o nosso "for", "\$i" é a variável inicial e "\$numero" é a variável final.

```
for ($i = 0; $i <= $numero; $i++) {
```

Vamos usar a variável "\$Url" para armazenar o endereço IP ou do site alvo que está armazenada em "\$array[\$i]" e atribuída a cada execução do laço.

```
$Url = "$array[$i]";
```

Se o endereço alvo não possui o protocolo HTTP, usamos um "IF" como condição.

Se endereço não possui HTTP, o if inseri HTTP. Exemplo:

Não possui o protocolo HTTP.

192.168.0.3

O "IF" verifica o endereço 192.168.0.3, não possui o protocolo HTTP. Então, inseri:

http://192.168.0.3

O endereço IP foi verificado pelo IF, o resultado foi inserir o protocolo "http", observe:

```
if($Url !~ /http:\\\\/) {
```

```
    $Url = "http://$Url";
```

```
}
```

Agora usamos o bloco para transformar nosso endereço alvo, observe:

```
$Stop = index($Url,":");
```

```
$Protocolo = substr($Url,0,$Stop);
```

```
$Start = index($Url,"/") + 2;
```

```
$Dominio = substr($Url,$Start);
```

```
$Stop = index($Dominio,"/");
```

```
$Dominio = substr($Dominio,0,$Stop);
```

```
$Start = rindex($Url,"/") + 1;
```

```
$NomeArq = substr($Url,$Start);
```

```
$Compr_Url = length($Url);
```

A variável "\$ponto" recebe o endereço pronto para ser verificado.

```
$ponto = "$Dominio \n";
```

O array "@portas", recebe um único elemento "1521", sendo a porta do banco de dados Oracle.

```
our @portas = "1521";
```

O nosso "foreach" inicia o nosso checagem de hosts com banco de dados Oracle.

```
foreach $porta (@portas) {
```

Criamos a variável "\$sock" usando o módulo "IO::Socket::INET", e conectando ao endereço alvo e a porta mencionada: 1521.

```
$sock = IO::Socket::INET->new("$ponto:$porta");
```

Na próxima linha, verificamos se a conexão com o servidor foi estabelecida, se foi estabelecida a conexão com o servidor alvo, capturamos o banner do servidor com banco de dados Oracle.

```
if($sock) {
```

```
    $remote = IO::Socket::INET -> new (Proto => "tcp", PeerAddr => $ponto,  
    PeerPort => $porta, Timeout => "7");
```

A nossa conexão, usou o módulo "IO::Socket::INET", o protocolo "TCP" e um Timeout de "7".

Agora, a linha com a variável "\$line" contém a resposta do servidor.

```
$line = <$remote>;
```

Checamos a resposta usando a expressão de condição "IF", se o banner contém os banners:

- 32-bit Windows: Version 8.1.7.0.0
- 32-bit Windows: Version 10.2.0.4.0
- 32-bit Windows: Version 8.1.7.0.0
- 32-bit Windows: Version 10.2.0.1.0

Se a captura do banner corresponder com a variável "\$line", temos um servidor vulnerável ao exploit do Metasploit.

Code desenvolvido por: Inj3cti0n P4ck3t
Nome: Fernando Henrique Mengali de Souza
E-mail to contact: fer_henrick@hotmail.com

Modo de uso: perl MetasploitScanOracle.pl listaDelps_ou_ListaDeSites.txt

```
};

exit();
}

open( SITE, "< $ARGV[0]" ) or die( "Nao foi possível abrir o arquivo: $!" );

our @array = <SITE>;

$numero = $#array;

for ($i = 0; $i <= $numero; $i++) {

    $Url = "$array[$i]";

    if($Url !~ /http:\\/\//) { $Url = "http://$Url"; }

    $Stop = index($Url,":");
    $Protocolo = substr($Url,0,$Stop);
    $Start = index($Url,"/") + 2;
    $Dominio = substr($Url,$Start);
    $Stop = index($Dominio,"/");
    $Dominio = substr($Dominio,0,$Stop);
    $Start = rindex($Url,"/") + 1;
    $NomeArq = substr($Url,$Start);
    $Compr_Url = length($Url);

    $ponto = "$Dominio \n";

    our @portas = "1521";

    foreach $porta (@portas) {

        $sock = IO::Socket::INET->new("$ponto:$porta");

        if($sock) {

            $remote = IO::Socket::INET -> new (Proto => "tcp", PeerAddr => $ponto,
            PeerPort => $porta, Timeout => "7");

            $line = <$remote>;

            if ($line =~ "32-bit Windows: Version 8.1.7.0.0") {
```



```
print "Server: $ponto vulnerável\n";

print "Exploit: tns_service_name \n\n";

}

if ($line =~ "32-bit Windows: Version 10.2.0.1.0" || $line =~ "32-bit Windows:
Version 10.2.0.4.0") {

print "Server: $ponto vulnerável\n";

print "Exploit: tns_auth_sesskey \n\n";

}

if ($line =~ "32-bit Windows: Version 8.1.7.0.0") {

print "Server: $ponto vulnerável\n";

print "Exploit: tns_arguments \n\n";

}

}

}

}
```

Agradecimentos aos amigos:

C00l3r - _MLK_ - s4r4d0 - DD3str0y3r - Sh0rtKiller - CODE RED - Forast - r0t3d
- dr4k3 - Archit3ct - D3UX - Believe - _Bl4ck9_f0x6 - fvox

Acesse: <http://www.botecounix.com.br>