## IDA Pro ile Remote Linux Debugging

## written by Mert SARICA | 31 October 2010

Windows bağımlısı biri olarak benim dünyamda Linux, hep sanal makina içinde çalışmaya mahkum olmuştur. Her ne kadar Ubuntu'yu çok seviyor olsamda alışkanlık ve oyunlar nedeniyle Windows kullanmaya uzun bir süre daha devam edeceğim gibi duruyor. Windows üzerinde debug için Ollydbg, Immunity Debugger ve IDA Pro araçlarını sıkça kullanıyorum fakat sistem Linux olunca GDB kullanmak gerçekten grafik arayüzü olmaması nedeniyle can sıkıcı olabiliyor.

Fakat benim gibi sadece tek bir işletim sistemi kullanmıyor sanal makinalardanda faydalanıyorsanız Linux üzerindeki bir programı debug etmek için IDA Pro'nun uzaktan (remote) debug özelliğinden faydalanabilirsiniz. Remote debugging, yerel veya uzaktaki ağ üzerinde yer alan bir sistemde çalışan bir programı kendi sisteminiz üzerinden debug etmenizi sağlar bu sayede örnek olarak uzak sistemde çalışan zararlı bir yazılımı kendi sisteminize zarar vermeden analiz etme imkanınız olmuş olur.

Benim gibi ana sistem olarak Windows 7 kullanıyorsanız ve sisteminizde IDA Pro v5.x yüklü ise şu adımları izleyerek Linux üzerindeki renksiz GDB'ye güzel bir alternatifiniz olabilir :)

Yazının daha kolay anlaşılabilmesi için örnek olarak Linux için hazırlanmış bir crackme programını debug edeceğiz. Crackme, tersine mühendislik becerilerinizi geliştirebilmeniz için internet üzerindeki gönüllüler tarafından kırılmak üzere hazırlanmış programlara verilen isimdir.

İlk olarak Crackmes.de sitesinden cyrex's Linux CrackMe programını indirelim ve C:\Program Files\IDA\crackme klasörü içine arşivden çıkartılmış halini kopyalayalım.

Öncelikle uzaktan debug etmek için Windows sisteminiz üzerinde ida adında bir kullanıcı yaratmanızı tavsiye ederim. Kullanıcıyı yarattıktan sonra C:\Program Files\IDA klasörünü paylaşıma açalım ve IDA kullanıcısını bu paylaşım üzerinde yetkilendirelim. Daha sonra sanal makina içinde yüklü olan Ubuntu'ya geçerek (evet herkes Ubuntu kullanmalı :p) bulunduğumuz klasör altında ida klasörü yaratalım ve arından smbmount komutu ile az önce yaratmış olduğumuz paylaşıma bağlanarak bu klasör içinde yer alan linux\_server programını çalıştıralım.



Daha sonra Windows'a geçerek IDA'yı çalıştıralım ve File menüsünden Open'a basarak C:\Program Files\IDA\crackme klasörü içinde yer alan crackme programını açalım.

Image: State description       Image: State description         Image: State description	The interactive disassembler File Edit Jump Search View Options Windows Help	
Image: Control of the set of the se		
A In (20) 四常 (2 · + N ) (2 · H · Y S N K / H · / P · ; 法理 山地 A A 平 A A 一 : : 本田 山地 A A 平 A A · · · · · · · · · · · · · · ·	■ ● ● ■ ●   ※ ※ ▲ ● / ■   ● 喩 ■ ● □	
Image: State description       Image: State description         Image: State description       OK         State: State description       OK         State: State description       OK         State: State: State description       OK         State: State	X En   2001 0001 0X1 "s" + ★ N ×   255 + # + 'x' S N K → → Ø	::幸辛 Ц临 孟杰辛杰杰
Image: Second		
Codific CV-forgen FileVDA/cackme/cackme as  File CV-forgen FileVDA/cackme/c		Load a new file
Binay file Processor type Intel 80x68 processor: metapc Loading segment Loading segment Loading gfiret Control to able of debugging Control to able o		Load file C:\Program Files\DA\crackme\crackme as ELF (Executable) (elk Idw)
Dupot window       OK       Cancel       Heb         System DLL directory       C:Windows         System DLL directory       C:Windows         System DLL directory       C:Windows		Binary file
Processor type         Intel 80x86 processors: metapc         Loading segment         Loading gifset         Loading gifset         Dotons         Create base for debugging         Kernel options1         Loading options         Processor pions         Processor options		
Intel 80x86 processors: metapc       Set         Loading gfret       Madyis:         Uptot window       Processor options         Dytes       pages size         description       OK         26214       32 8322 allocating memory for bitres		Processor type
Loading segment       Madyis         Updation       P Enabled         Options       P Enabled         P Rename DLL entries       Kernel options2         P Rename DLL entries       Processor options         Dadrag options       Processor options         P Ender Size       OK         Date Size description       OK         25214       32 Size allocating memory for bitres		Intel 80x86 processors: metapc
Utgett window       OK       Cancel       Help         Dytes       pages size       description       OK       Cancel       Help         OK       Cancel       Help       Manual		Loading segment 0x00000000 Analysis
Output window       Image: Control in the point of the p		Loading offset 0x0000000 V Indicator enabled
Image: Duput window       Image: Cancel Help         Image: Duput Help: Duput Stating memory for Virtual array       Image: Cancel Help         Image: Duput Help: Duput Stating memory for Virtual array       Image: Cancel Help         Image: Duput Help: Duput Stating memory for Virtual array       Image: Cancel Help         Image: Duput Help: Duput Stating memory for Virtual array       Image: Cancel Help         Image: Duput Help: Duput Stating He		Options
Image: Control of the control of th		Create base for debugging Kernel options1
Image: Second		Image: Coord resources       Image: The coord resources       Image: The coord resources
Cutput window     Create FLAT group     Processor options       B) Output window     Create FLAT group     Processor options       b) tes     pages size description     OK     Cancel       262144     32 8192 allocating memory for b-tree     OK     Cancel       565366     8 8192 allocating memory for or interes     Free pages size allocating memory for options     Image: Club allocating memory for options		Manual load Kernel options2
Dutput window     Dutput window     Diverse pages size description     DK     Cancel     Help       262144     32 8192 allocating memory for b-tree     DK     Cancel     Help     A       262144     32 8192 allocating memory for oritrual array     262144     32 8192 allocating memory for oritrual array       262144     32 8192 allocating memory for oritrual array       262144     32 8192 allocating memory for oritrual array       5693624     total memory allocated       ording TUP module C1/Program ElleSigners/Stor.w22 for processor metanc0K		Loading options Processor options
System DLL directory     C.Windows       Dutput window     0K     Cancel       bytes     pages size description     0K     Cancel       262144     32 8192 allocating memory for b-tree     a       262144     32 8192 allocating memory for name pointers     a       589824     total memory allocated       opding TUP module C1/Program E1185/Stocky22 for processor metanc0K		
Dutput window       DK       Cancel       Help       Image: Concel C		System DLL directory C:\Windows
bytes pages size description the state of th	💹 Output window	DK Carrel Help
26214       32 8192 allocating memory for b-tree         65336       8 8192 allocating memory for virtual array         262144       32 8192 allocating memory for virtual array         5698024       total memory allocated         ording TDP module C:\Program Files/DAlproprise/suprava2 for processor metanc0K	bytes pages size description	
589824 total memory allocated	262144     32 8192 allocating memory for b-tree       65536     8 8192 allocating memory for virtual array       262144     32 8192 allocating memory for name pointers	
nading TDP module C:\Program Files\TDA\procs\pc.w32 for processor metapcOK	589824 total memory allocated	
Autoanalysis subsystem has been initialized. Possible file format: ELF (Executable) (C:\Program Files\IDA\loaders\elf.ldw)	Loading IDP module C:\Program Files\IDA\procs\pc.w32 for processor met Autoanalysis subsystem has been initialized. Possible file format: ELF (Executable) (C:\Program Files\IDA\loaders\o	tapcOK

Daha sonra üstteki menüden Debugger'ı seçelim ve daha sonra Remote Linux Debugger'ı seçelim. Ardından Debugger menüsünden Start process'i seçelim ve Hostname kısmına sanal makina içinde çalışan Ubuntu sisteminin IP adresini girelim.

IDA - C:\Program Files\IDA\crackme\crackm Elle Edit Jump Search View Debugger Options	eX	
	Treat I I I I I I I I I I I I I I I I I I I	
Fil Functions window	X III IDA Viewak X IIII Hex Viewak X X Supplies X En France X Kill Inports X III Frances	
Function name     A       Binit_proc     Binit_proc       Binit_proc     B	Debug application setup: inux         Application         Upt file         C:VPogram Files/DA/crackme         Upt file         C:VPogram Files/DA/crackme/crackme         Upt file         Directory         C:VPogram Files/DA/crackme         V         Directory         C:VPogram Files/DA/crackme         V         Passovid	
Dinit_dummy_0	call _print	
	100.00% (-46.40) (225.11) 00000450 08048450 main	
🗒 Output window		
Copyright (c) 2004-2009 Gergely Erdely1 - http://d-dome.net/idapython/		
Propagating type information Function argument information has been The initial autoanalysis has been finis	propagated C	
Python		
All'idle Down Disk: 4GB		

IP adresini girdikten sonra ise Debugger menüsünden Start process'i seçerek
debug işlemini başlatalım. (Her iki uyarı mesajınıda Yes diyerek
geçebiliriz.)

Crackme'yi kırmak için bizden doğru şifreyi bulmamız isteniyor. Debug işlemi başladıktan sonra Ubuntu'ya bakacak olursanız ekranda sizden doğru şifreyi girmenizi istediğini görebilirsiniz. Buraya rastgele bir şifre girdiğimizde (12345) hata mesajı ile karşılaşıyoruz ve debug işlemi sonlanıyor.

Amacımız doğru şifreyi bulmak olduğu için bunun için IDA'da Shift 12 tuşlarına basarak Strings penceresini açalım ve az önce karşılaştığımız hata mesajının üzerine iki defa basarak program üzerinde bu değişkenin tutulduğu ilgili bölüme gidelim.

Faremizin imlecini char aOhhhhYourSkill[] üzerine getirdikten sonra x tuşuna

basarak bu değişkeni çağıran kod parçasına gidelim.



Bu kodun üzerine hızlıca göz attığımızda kullanıcıdan alınan verinin yani şifrenin strcmp fonksiyonu yardımı ile 47ghf6fh37fbgbg değeri ile karşılaştırıldığını ve doğru olması durumunda Good ile başlayan mesaja aksi durumda Ohhh ile başlayan hata mesajına gittiğimizi görüyoruz ve şifrenin 47ghf6fh37fbgbg olduğunu öğrenmiş oluyoruz ve crackme başarıyla çözülmüş oluyor.

root@bt:~# mkdir ida root@bt:~# smbmount //192.168.1.3/ida ida -o username=ida,password=ida,rw root@bt:~# ./ida/linux\_server IDA Linux remote debug server(ST). Version 1.10. Copyright HexRays 2004-2009 Listening on port #23946... Accepting incoming connection... td\_ta\_new: application not linked with libthread td\_ta\_new: application not linked with libthread -[ Linux CrackMe (Level:2) by cyrex ]--[ TODO: You have to get the valid Password ]-Enter Password: 12345 -[ Ohhhh, your skills are bad try again later ]-Closing incoming connection... Accepting incoming connection... td\_ta\_new: application not linked with libthread td\_ta\_new: application not linked with libthread -[ Linux CrackMe (Level:2) by cyrex ]--[ TODO: You have to get the valid Password ]-Enter Password: 47ghf6fh37fbgbgj -[ Good, You're ready to begin linux reversing ]-Closing incoming connection...

Gördüğünüz üzere Windows üzerinde çalışan IDA ile Linux üzerindeki bir programı debug etmek GDB'nin aksine daha kolay ve eğlenceli olabiliyor.

Bir sonraki yazıda görüşmek dileğiyle herkese iyi haftalar dilerim.

Not: Geçtiğimiz aylarda yayınlanan v6 sürümü ile IDA Pro kullanıcıları Linux ve Mac OS X üzerinde GUI arayüzüne kavuştu.