Guest'i Hazırlamak

Bu noktada Cuckoo ana bilgisayar bileşenini yapılandırmış olmalısınız ve kötü amaçlı yazılım yürütme için kullanacağınız sanal makinelerin sayısını ve adlarını tasarlamış ve tanımlamış olmalısınız. Şimdi bu makineleri oluşturmanın ve uygun şekilde yapılandırmanın zamanı geldi.

- Sanal Makineyi Oluşturmak
- Python Kurulumu
- Ek Yazılımlar
- Network Konfigürasyonu
- Agent Kurulumu
- Sanal Makineyi Kaydetmek
- Sanal Makinenin Klonlanması
- Linux Host Kurulumu

Sanal Makineyi Oluşturmak

Sanallaştırma yazılımınızı doğru bir şekilde kurduktan sonra, ihtiyacınız olan tüm sanal makineleri oluşturabilirsiniz.

Sanallaştırma yazılımınızı kullanmak ve yapılandırmak, bu kılavuzun kapsamı dışındadır.

> Sanallaştırılmış ortamınızı nasıl tasarlayıp oluşturacağınıza dair bazı ipuçları ve düşünceleri Sandboxing bölümünde bulabilirsiniz.

> 64-bit Windows 7 veya Windows XP sanal makineleri önerilir. Windows 7 için Kullanıcı Erişim Kontrolü'nü devre dışı bırakmanız gerekecektir. 2.0-rc2 sürümünde değiştirildi: Eskiden Windows XP bir konuk VM olarak önerilirdi, ancak günümüzde 64-bit Windows 7 makinesi çok daha iyi sonuçlar vermektedir.

> KVM Kullanıcıları - Kesinlikle snapshot'ları destekleyen bir hard disk görüntü formatı seçtiğinizden emin olun. Daha fazla bilgi için Sanal Makineyi Kaydetme bölümüne bakın.

Sanal makine oluşturulurken, Cuckoo'nun belirli bir yapılandırmaya ihtiyacı yoktur. İhtiyaçlarınıza en uygun seçenekleri seçebilirsiniz.

Cuckoo'yu sanallaştırılmış Windows sisteminizde düzgün çalıştırmak için bazı gerekli yazılımları ve kütüphaneleri kurmanız gerekecektir.

Python Kurulumu

Cuckoo konuk bileşeninin düzgün çalışabilmesi için Python kesin bir gerekliliktir.

Resmi web sitesinden uygun Windows yükleyicisini indirebilirsiniz. Bu durumda da Python 2.7 tercih edilir.

Bazı Python kütüphaneleri ise Cuckoo konuk bileşenine bazı ek özellikler sağlamak için isteğe bağlıdır. Bunlar şunları içerir:

- Python Pillow: Analiz sırasında Windows masaüstünün ekran görüntülerini almak için kullanılır.

Bunlar, Cuckoo'nun düzgün çalışması için kesin olarak gerekliliği olmayan ancak tüm mevcut özelliklere erişmek istiyorsanız bunları kurmanızı önerdiğimiz kütüphanelerdir. Python sürümünüze göre doğru paketleri indirip kurduğunuzdan emin olun.

Ek Yazılımlar

Bu noktada, Cuckoo'nun düzgün çalışması için gerekli olan her şeyi kurmuş olmalısınız.

Hangi tür dosyaları analiz etmek istediğinize ve hangi türde bir kumlu Windows ortamında kötü amaçlı yazılım örneklerini çalıştırmak istediğinize bağlı olarak tarayıcılar, PDF okuyucular, ofis paketleri vb. gibi ek yazılımları kurabilirsiniz. Ek yazılımların 'otomatik güncelleme' veya 'güncellemeleri kontrol et' özelliğini devre dışı bırakmayı unutmayın.

Bu tamamen size ve ihtiyaçlarınıza bağlıdır. Sandboxing bölümünü okuyarak bazı ipuçları alabilirsiniz.

Network Konfigürasyonu

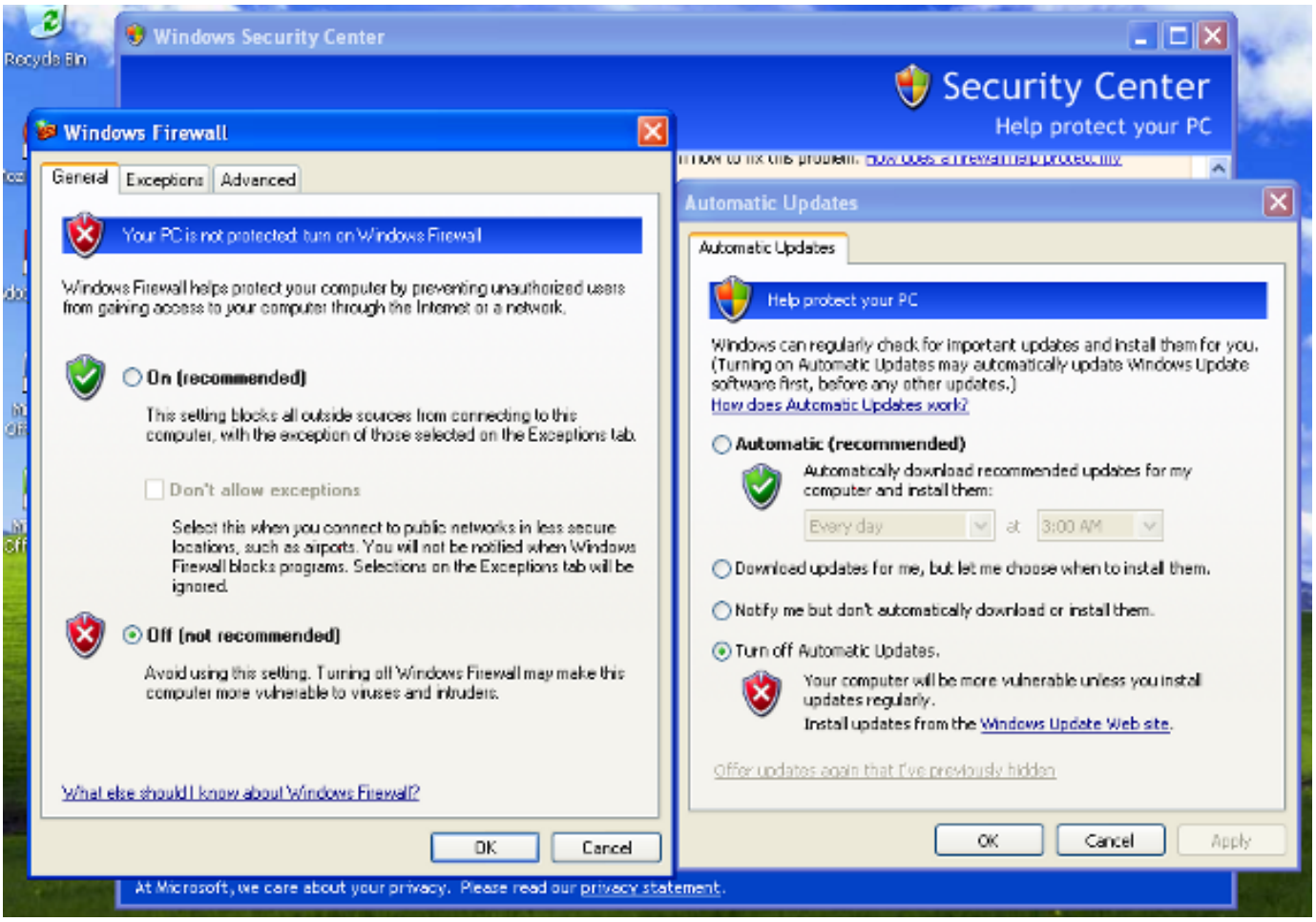
Şimdi sanal makineniz için ağ kurma zamanı geldi.

Windows Ayarları

Sanal makinenin temel ağ yapılandırmasını yapılandırmadan önce, Windows içinde bazı ayarları düzenlemek isteyebilirsiniz.

Yapılması gereken en önemli şeylerden biri, Windows Güvenlik Duvarı'nı ve Otomatik Güncellemeler'i devre dışı bırakmaktır. Bunun nedeni, bunların normal koşullar altında kötü amaçlı yazılımın davranışını etkileyebileceği ve Cuckoo tarafından gerçekleştirilen ağ analizini, bağlantıları bırakarak veya gereksiz istekleri içerecek şekilde kirletebileceğidir.

Bu ayarları Windows'un Denetim Masası'ndan aşağıdaki resimde gösterildiği gibi yapabilirsiniz:



Virtual Networking

Şimdi sanal makinenizin İnternet'e veya yerel ağınıza erişebilmesi için nasıl yapılandırılacağına karar vermeniz gerekiyor.

Önceki sürümlerde Cuckoo, Ana Bilgisayar ve Konuklar arasında veri alışverişi için paylaşılan klasörleri kullanırken, 0.4 sürümünden itibaren basit bir XMLRPC protokolü kullanarak ağ üzerinden çalışan özel bir ajanı benimsemektedir.

Bu düzgün çalışabilmesi için makinenizin ağını öyle yapılandırmanız gerekecek ki Host ve Guest iletişim kurabilsin. Sanal ağın doğru bir şekilde kurulup kurulmadığını kontrol etmek için bir konuğu ping ile test etmek iyi bir uygulamadır. Konuğunuz için yalnızca statik IP adresleri kullanın, çünkü Cuckoo DHCP'yi desteklemez ve kullanmak kurulumunuzu bozacaktır.

Bu aşama, kendi gereksinimlerinize ve sanallaştırma yazılımınızın özelliklerine çok bağlıdır.

> Sanal ağ, Cuckoo için hayati bir bileşendir, ana bilgisayar ile konuk arasında bağlantı almak için gerçekten emin olmalısınız. Kullanıcılar tarafından bildirilen sorunların çoğu, sanal ağlarının yanlış yapılandırılmasıyla ilgilidir. Eğer bundan emin değilseniz, sanallaştırma yazılımınızın dokümanını kontrol edin ve bağlantıyı ping ve telnet ile test edin.

Tavsiye edilen kurulum, doğru yönlendirme ile bir Host-Only ağ düzeni kullanmaktır. Bu tür ağ yönlendirmeleri hakkında daha fazla bilgi, ana makine kurulumunun bir parçası olan Per-Analysis Network Routing bölümünde bulunabilir.

Agent Kurulumu

Sürüm 0.4'ten itibaren Cuckoo, Gast içinde çalışan ve iletişimi ve veri alışverişini Host ile yöneten özel bir ajan kullanır. Bu ajanın çeşitli platformlarda çalışabilmesi için tasarlanmış olması, bu nedenle Windows, Android, Linux ve Mac OS X üzerinde kullanabilirsiniz. Cuckoo'nun düzgün çalışabilmesi için bu ajanı kurmanız ve çalıştırmanız gerekecek.

Oldukça basit.

`$CWD/agent/` dizininde `agent.py` dosyasını bulacaksınız. Bu dosyayı Guest işletim sistemine kopyalayın (istediğiniz şekilde, belki geçici bir paylaşılan klasör veya ana bilgisayardaki bir web sunucusundan indirerek, bu sonuncusunu önerilir) ve çalıştırın. Ajan, ana bilgisayarın iletişim kurabileceği küçük bir API sunucusunu başlatacaktır.

Windows üzerinde basitçe komut dosyasını başlatmak, aynı zamanda bir Python penceresi de başlatacaktır; eğer bunu gizlemek istiyorsanız dosyayı `agent.py`'den `agent.pyw` olarak yeniden adlandırabilirsiniz, bu da konsol penceresinin başlamasını engeller.

Eğer Windows'un başlangıcında betiğin başlatılmasını istiyorsanız, dosyayı Başlangıç klasörüne yerleştirin.

Sanal Makineyi Kaydetmek

Şimdi sanal makinenizi bir anlık görüntü durumuna kaydetmeye hazır olmalısınız.

Bunu yapmadan önce, onu nazikçe yeniden başlattığınızdan ve şu anda çalıştığından, Cuckoo'nun ajanının çalıştığından ve Windows'un tamamen başladığından emin olun.

Şimdi makineyi kaydetmeye devam edebilirsiniz. Bunun nasıl yapılacağı açıkça kullandığınız sanallaştırma yazılımına bağlıdır.

Eğer aşağıdaki adımları düzgün bir şekilde takip ederseniz, sanal makineniz Cuckoo tarafından kullanılmaya hazır olacaktır.

Virtualbox

VirtualBox'u seçiyorsanız, anlık görüntüyü grafik arayüzden veya komut satırından alabilirsiniz:

```
$ VBoxManage snapshot "<Name of VM>" take "<Name of snapshot>" --pause
```

Anlık görüntü oluşturma işlemi tamamlandıktan sonra makineyi kapatıp geri yükleyebilirsiniz:

```
$ VBoxManage controlvm "<Name of VM>" poweroff
$ VBoxManage snapshot "<Name of VM>" restorecurrent
```

KVM

Eğer KVM'i kullanmaya karar verdiyseniz, öncelikle sanal makineleriniz için anlık görüntüler destekleyen bir disk formatı kullandığınızdan emin olmalısınız. Libvirt araçları varsayılan olarak RAW sanal diskler oluşturur, ve çünkü bizim anlık görüntülere ihtiyacımız var, QCOW2 veya LVM kullanmanız gerekecek. Bu kılavuzun kapsamı için QCOW2'yi benimsemekteyiz, ki bu LVM'den daha kolay kuruludur.

Bu tür bir sanal diski doğru bir şekilde oluşturmanın en kolay yolu, libvirt paketi tarafından sağlanan araçları kullanmaktır. Virsh'i tercih ediyorsanız komut satırı arayüzünü veya güzel bir GUI

için virt-manager'ı kullanabilirsiniz. QCOW2 formatında doğrudan oluşturabilirsiniz, ancak RAW bir diskiniz varsa bunu şu şekilde dönüştürebilirsiniz:

```
$ cd /your/disk/image/path
$ qemu-img convert -O qcow2 your_disk.raw your_disk.qcow2
```

Şimdi VM tanımınızı aşağıdaki gibi düzenlemeniz gerekiyor:

```
$ virsh edit "<Name of VM>"
```

Disk bölümünü bulun, şu şekilde görünüyor:

```
<disk type='file' device='disk'>
   <driver name='qemu' type='raw'/>
   <source file='/your/disk/image/path/your_disk.raw'/>
   <target dev='hda' bus='ide'/>
   <address type='drive' controller='0' bus='0' unit='0'/>
</disk>
```

“type”'ı qcow2 olarak değiştirin ve “source file”'ı qcow2 disk görüntünüze değiştirin, şu şekilde:

```
<disk type='file' device='disk'>
   <driver name='qemu' type='qcow2'/>
   <source file='/your/disk/image/path/your_disk.qcow2'/>
   <target dev='hda' bus='ide'/>
   <address type='drive' controller='0' bus='0' unit='0'/>
</disk>
```

Şimdi sanal makinenizi test edin, eğer her şey çalışıyorsa Cuckoo Agent çalıştırırken sanal makineyi anlık olarak hazırlayın. Bu, sanal makineyi anlık alırken çalışır durumda olması gerektiği anlamına gelir. Ardından, onu kapatabilirsiniz. Aşağıdaki komutla nihayet bir anlık görüntü alabilirsiniz:

```
$ virsh snapshot-create "<Name of VM>"
```

Birkaç anlık görüntü almak hatalara neden olabilir:

```
ERROR: No snapshot found for virtual machine VM-Name
```

VM anlık görüntüleri aşağıdaki komutlarla yönetilebilir:

```
$ virsh snapshot-list "VM-Name"
$ virsh snapshot-delete "VM-Name" 1234567890
```

Vmware Workstation

Eğer VMware Workstation kullanmaya karar verdiyseniz, anlık görüntüyü grafik kullanıcı arayüzünden veya komut satırından alabilirsiniz:

```
$ vmrun snapshot "/your/disk/image/path/wmware_image_name.vmx"
your_snapshot_name
```

Burada your_snapshot_name, anlık görüntü için seçtiğiniz addır. Ardından makineyi GUI veya komut satırından kapatın:

```
$ vmrun stop "/your/disk/image/path/wmware_image_name.vmx" hard
```

Sanal Makinenin Klonlanması

Eğer birden fazla sanal makine kullanmayı planlıyorsanız, şimdiye kadar yapılan tüm adımları tekrarlamaya gerek yok: onu klonlayabilirsiniz. Böylece, tüm gereksinimleri zaten yüklenmiş olan orijinal sanal Windows'un bir kopyasına sahip olacaksınız.

Yeni sanal makine aynı zamanda orijinalin tüm ayarlarını içerecektir, ki bu da iyi değildir. Şimdi, bu yeni makine için Network Konfigürasyonu, Agent Kurulumu ve Sanal Makineyi Kaydetmek adımlarını tekrarlayarak devam etmeniz gerekiyor.

Linux Host Kurulumu

Öncelikle, makine platformunuz için ana bilgisayar tarafındaki ağ yapılandırmasını hazırlayın. Örneğin, VirtualBox'ı host-only arabirimleriyle kullanıyorsanız ve vboxnet0 arabiriminiz varsa, ek bağımlılıkları kurmanıza gerek yoktur.

QEMU kullanıyorsanız, ana bilgisayarda ek bağımlılıkları kurmanız gerekebilir:

```
$ sudo apt install uml-utilities bridge-utils
```

Ardından, arabirimini yapılandırmak için sanal makinelerin listesini conf/qemu.conf'dan alın. Örneğin, ubuntu_x32, ubuntu_x64, ubuntu_arm, ubuntu_mips, ubuntu_mipsel vb. Her bir VM için, kök olarak başlatmak zorunda kalmamak için ana bilgisayarda bir ağ tap arabirimi önceden yapılandırın, örneğin:

```
$ sudo tunctl -b -u cuckoo -t tap_ubuntu_x32
$ sudo ip link set tap_ubuntu_x32 master br0
$ sudo ip link set dev tap_ubuntu_x32 up
$ sudo ip link set dev br0 up

$ sudo tunctl -b -u cuckoo -t tap_ubuntu_x64
$ sudo ip link set tap_ubuntu_x64 master br0
$ sudo ip link set dev tap_ubuntu_x64 up
$ sudo ip link set dev br0 up
```

> Cuckoo'yu farklı bir kullanıcı olarak çalıştırıyorsanız, -u'dan sonraki "cuckoo" ifadesini kendi kullanıcı adınızla değiştirin.

x32/x64 Ubuntu 18.04 Linux Guest’i Hazırlamak

Agent’ın otomatik olarak başladığından emin olun. En kolay yol, crontab'a eklemektir:

```
```

```
$ sudo crontab -e
@reboot python /path/to/agent.py
```

Sanal makine içerisinde bağımlılıkları yükleyin:

```
$ sudo apt-get install systemtap gcc patch linux-headers-$(uname -r)
```

Kernel debugging symbols kurun:

```
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
C8CAB6595FDFF622

$ codename=$(lsb_release -cs)
$ sudo tee /etc/apt/sources.list.d/ddebs.list << EOF
  deb http://ddebs.ubuntu.com/ ${codename}          main restricted universe
multiverse
  #deb http://ddebs.ubuntu.com/ ${codename}-security main restricted
universe multiverse
  deb http://ddebs.ubuntu.com/ ${codename}-updates  main restricted universe
multiverse
  deb http://ddebs.ubuntu.com/ ${codename}-proposed main restricted universe
multiverse
EOF

$ sudo apt-get update
$ sudo apt-get install linux-image-$(uname -r)-dbgsym
```

(Debian 9 amd64 için) Kernel debugging symbols kurun:

```
$ sudo apt-get install linux-image-$(uname -r)-dbg
```

SystemTap tapset'i düzeltin, böylece Cuckoo analyzer çıktıyı düzgün bir şekilde çözebilir:

```
$ wget
https://raw.githubusercontent.com/cuckoosandbox/cuckoo/master/stuff/systemt
ap/expand_execve_envp.patch
```

```
$ wget
https://raw.githubusercontent.com/cuckoosandbox/cuckoo/master/stuff/systemt
ap/escape_delimiters.patch
$ sudo patch /usr/share/systemtap/tapset/linux/sysc_execve.stp <
expand_execve_envp.patch
$ sudo patch /usr/share/systemtap/tapset/uconversions.stp <
escape_delimiters.patch
```

Kernel extension'ı derleyin:

```
$ wget
https://raw.githubusercontent.com/cuckoosandbox/cuckoo/master/stuff/systemt
ap/strace.stp
$ sudo stap -p4 -r $(uname -r) strace.stp -m stap_ -v
```

Derleme işlemi tamamlandığında aynı klasörde stap_.ko dosyasını görmelisiniz. Şimdi STAP kernel extension'ı aşağıdaki gibi test edebilirsiniz:

```
$ sudo staprun -v ./stap_.ko
```

Çıktı aşağıdaki gibi olmalıdır:

```
staprun:insert_module:x Module stap_ inserted from file path_to_stap_.ko
```

`stap_.ko` dosyasını `/root/.cuckoo` dizinine yerleştirmelisiniz:

```
$ sudo mkdir /root/.cuckoo
$ sudo mv stap_.ko /root/.cuckoo/
```

Eğer varsa, sanal makinedeki güvenlik duvarını devre dışı bırakın:

```
$ sudo ufw disable
```

Sanal makine içinde NTP'yi devre dışı bırakın:

```
$ sudo timedatectl set-ntp off
```

İsteğe bağlı - önceden yüklenmiş yazılım ve konfigürasyonları kaldırın:

```
$ sudo apt-get purge update-notifier update-manager update-manager-core
ubuntu-release-upgrader-core
$ sudo apt-get purge whoopsie ntpdate cups-daemon avahi-autoipd avahi-
daemon avahi-utils
$ sudo apt-get purge account-plugin-salut libnss-mdns telepathy-salut
```

Linux guest sistemini statik IP adresleriyle yapılandırmak önerilir. Konfigürasyondaki makine girişinin doğru IP adresine sahip olduğundan ve platform değişkeninin linux olarak ayarlandığından emin olun. VM yapılandırıldıktan sonra bir anlık görüntü oluşturun. Şimdi analiz için hazırız.