**3. A Brief Introduction to dynamic analysis**

\*Dinamik Analize Kısa Bir Giriş:

Statik analiz, diskte farklı bileşenlerini inceleyerek kötü amaçlı yazılım hakkında yararlı bilgiler elde etmenin etkili bir yolu olsa da, kötü amaçlı yazılım davranışını gözlemlememize izin vermez.Bu bölümde, dinamik kötü amaçlı yazılım analizinin temellerini öğreneceksiniz. Kötü amaçlı yazılımın dosya biçiminde neye benzediğine odaklanan statik analizin aksine, dinamik analiz kötü amaçlı yazılımın nasıl davrandığını görmek için güvenli, kapalı bir ortamda çalıştırmayı içerir.. Dinamik analizi kullanarak, paketleme ve gizleme gibi yaygın statik analiz engellerini aşabilir ve belirli bir kötü amaçlı yazılım örneğinin amacına ilişkin daha doğrudan içgörü elde edebiliriz.Temel dinamik analiz tekniklerini, bunların kötü amaçlı yazılım veri bilimiyle ilgisini ve uygulamalarını keşfederek başlıyoruz. Dinamik analiz örneklerini iş başında incelemek için malwr.com gibi açık kaynaklı araçlar kullanıyoruz. Bunun, konuyla ilgili yoğun bir anket olduğunu ve kapsamlı olması amaçlanmadığını unutmayın. Daha eksiksiz bir giriş için Pratik Kötü Amaçlı Yazılım Analizine (No Starch Press, 2012) göz atın.

\*Why use Dynamic Analysis?

Dinamik analizin neden önemli olduğunu anlamak için, paketlenmiş kötü amaçlı yazılım sorununu ele alalım. Kötü amaçlı yazılım paketlemenin, programın kötü niyetli doğasını gizlemek için kötü amaçlı yazılımın x86 derleme kodunu sıkıştırmak veya gizlemek anlamına geldiğini hatırlayın. Paketlenmiş bir kötü amaçlı yazılım örneği, kodun çalıştırılabilmesi için bir hedef makineye bulaştığında kendi kendini açar. Bölüm 2'de tartışılan statik analiz araçlarını kullanarak paketlenmiş veya gizlenmiş bir kötü amaçlı yazılım örneğini sökmeye çalışabiliriz, ancak bu zahmetli bir süreçtir. Örneğin, statik analizde, önce kötü amaçlı yazılım dosyasında karmaşık hale getirilmiş kodun konumunu bulmamız gerekir. Ardından, çalışabilmesi için bu kodu gizlemeyi kaldıran gizleme kaldırma alt yordamlarının konumunu bulmamız gerekir. Alt rutinleri bulduktan sonra, kod üzerinde gerçekleştirmek için bu gizleme kaldırma prosedürünün nasıl çalıştığını bulmamız gerekir. Ancak o zaman kötü amaçlı koda ters mühendislik uygulama sürecine başlayabilirdik. Bu işlemin basit ama akıllıca bir alternatifi, kötü amaçlı yazılımı korumalı alan adı verilen güvenli ve kapalı bir ortamda yürütmektir. Bir sanal alanda kötü amaçlı yazılım çalıştırmak, gerçek bir hedefe bulaşırken olduğu gibi kendi kendini açmasına izin verir. Yalnızca kötü amaçlı yazılım çalıştırarak, belirli bir kötü amaçlı yazılım ikili programının hangi sunuculara bağlandığını, hangi sistem yapılandırma parametrelerini değiştirdiğini ve hangi cihaz G / Ç (giriş / çıkış) gerçekleştirmeye çalıştığını öğrenebiliriz.

\*Dynamic Analysis for Malware Data Science:

Dinamik analiz, yalnızca kötü amaçlı yazılım ters mühendisliği için değil, aynı zamanda kötü amaçlı yazılım veri bilimi için de yararlıdır. Dinamik analiz, bir kötü amaçlı yazılım örneğinin ne yaptığını ortaya çıkardığından, eylemlerini diğer kötü amaçlı yazılım örnekleriyle karşılaştırabiliriz. Örneğin, dinamik analiz kötü amaçlı yazılım örneklerinin diske hangi dosyaları yazdığını gösterdiğinden, bu verileri diske benzer dosya adları yazan kötü amaçlı yazılım örneklerini bağlamak için kullanabiliriz. Bu tür ipuçları, kötü amaçlı yazılım örneklerini ortak özelliklere göre sınıflandırmamıza yardımcı olur. Aynı gruplar tarafından yazılan veya aynı kampanyaların parçası olan kötü amaçlı yazılım örneklerini belirlememize bile yardımcı olabilirler. En önemlisi, dinamik analiz, makine öğrenimi tabanlı kötü amaçlı yazılım algılayıcıları oluşturmak için yararlıdır. Dinamik analiz sırasında davranışlarını gözlemleyerek kötü niyetli ve iyi huylu ikili dosyaları ayırt etmek için bir dedektörü eğitebiliriz. Örneğin, hem kötü amaçlı yazılımlardan hem de zararlı dosyalardan binlerce dinamik analiz günlüğünü gözlemledikten sonra, bir makine öğrenme sistemi, msword.exe, powershell.exe adında bir işlem başlattığında bu eylemin kötü amaçlı olduğunu ancak msword.exe'nin Internet Explorer'ı başlattığında bunu öğrenebilir. , bu muhtemelen zararsızdır. Bölüm 8, hem statik hem de dinamik analize dayalı verileri kullanarak kötü amaçlı yazılım algılayıcılarını nasıl oluşturabileceğimiz hakkında daha ayrıntılı bilgi verecektir. Ancak gelişmiş kötü amaçlı yazılım algılayıcıları oluşturmadan önce, dinamik analiz için bazı temel araçlara bakalım.

\*Basic Tools for Dynamic Analysis:

Çevrimiçi dinamik analiz için bir dizi ücretsiz, açık kaynaklı araç bulabilirsiniz. Bu bölüm malwr.com ve CuckooBox'a odaklanmaktadır. Malwr.com sitesi, dinamik analiz için ücretsiz ikili dosyalar göndermenize izin veren bir web arayüzüne sahiptir. CuckooBox, ikili dosyaları yerel olarak analiz edebilmeniz için kendi dinamik analiz ortamınızı kurmanıza izin veren bir yazılım platformudur. CuckooBox platformunun yaratıcıları ayrıca malwr.com'u ve malwr.com'un perde arkasında CuckooBox'ı çalıştırıyor. Bu nedenle, malwr.com'da sonuçları nasıl analiz edeceğinizi öğrenmek, CuckooBox sonuçlarını anlamanıza yardımcı olacaktır.

Not: Baskı sırasında, malwr.com’un CuckooBox arayüzü bakım nedeniyle kapalıydı. Umarım bu bölümü okuduğunuzda site tekrar açılır. Aksi takdirde, bu bölümde sağlanan bilgiler, https://cuckoosandbox.org/ adresindeki talimatları izleyerek kurabileceğiniz kendi CuckooBox örneğinizden çıktı almak için uygulanabilir.

pip install -U cuckoo

önce indirmeden önce gerekli ortam ayarlanmalı

\*Typical Malware Behaviors:Tipik Kötü Amaçlı Yazılım Davranışları

Aşağıda, bir kötü amaçlı yazılım örneğinin yürütme sırasında gerçekleştirebileceği başlıca eylem kategorileri verilmiştir:

Dosya sistemini değiştirme Modifying the file system  Örneğin, diske bir aygıt sürücüsü yazma writing a device driver to disk , sistem yapılandırma dosyalarını değiştirme, dosya sistemine yeni programlar ekleme ve programın otomatik olarak başlamasını sağlamak için kayıt defteri anahtarlarını değiştirme

  Sistem yapılandırmasını değiştirmek için Windows kayıt defterini değiştirme Modifying the windows registry to change the system configuration Örneğin, güvenlik duvarı ayarlarını değiştirme

Aygıt sürücülerini yükleme Loading device drivers  Örneğin, kullanıcı tuş vuruşlarını kaydeden bir aygıt sürücüsünün yüklenmesi

Ağ eylemleri  Network acitons Örneğin, alan adlarının çözümlenmesi ve HTTP isteklerinin yapılması

  Bu davranışları, bir kötü amaçlı yazılım örneği kullanarak ve malwr.com'daki raporunu analiz ederek daha ayrıntılı olarak inceleyeceğiz.

\*Loading  a File on [malwr.com](http://malwr.com)

Malwr.com üzerinden bir kötü amaçlı yazılım örneği çalıştırmak için https://malwr.com/ adresine gidin ve ardından bir ikili program yükleyip analiz için göndermek için Gönder düğmesine tıklayın. Bu bölüme eşlik eden veri dizininde bulabileceğiniz SHA256 hash'i d676d95 karakterleriyle başlayan bir ikili dosya kullanacağız. Bu ikili dosyayı malwr.com'a göndermenizi ve giderken sonuçları kendiniz incelemenizi tavsiye ederim. Gönder sayfası Şekil 3-1'de gösterilmektedir.

Örneğinizi bu form aracılığıyla gönderdikten sonra site,

genellikle yaklaşık beş dakika süren analizin tamamlanmasını beklemeniz gerekir.

Sonuçlar yüklendiğinde, dinamik analiz ortamında çalıştırıldığında yürütülebilir dosyanın ne yaptığını anlamak için onları inceleyebilirsiniz.

\*Analyzing Results on [malwr.com](http://malwr.com):

Bu dosyanın sonuçları, daha sonra keşfedeceğimiz dinamik analizin bazı temel yönlerini göstermektedir.

->Signatures Panel:İmzalar Paneli:

Sonuçlar sayfasında göreceğiniz ilk iki panel Analiz ve Dosya Ayrıntılarıdır. Bunlar, dosyanın çalıştırıldığı zamanı ve dosyayla ilgili diğer statik ayrıntıları içerir. Burada odaklanacağım panel, Şekil 3-3'te gösterilen İmzalar panelidir. Bu panel, dosyanın kendisinden ve dinamik analiz ortamında çalıştırıldığında davranışından türetilen üst düzey bilgileri içerir. Bu imzaların her birinin ne anlama geldiğini tartışalım.

Signatures:

File has been identified by at least one AntiVirus on VirusTotal as malicious

The binary likely contains encrypted or compressed data

The executable is compressed using UPX

Collects information to fingerprint the system (MachineGuid , DigitalProductId,SystemBiosDate)

Creates an Alternate Data Stream (ADS)

Installs itself for autorun at Windows startup

Dosya, VirusTotal'da en az bir AntiVirus tarafından kötü amaçlı olarak tanımlandı

İkili büyük olasılıkla şifrelenmiş veya sıkıştırılmış veriler içerir

Yürütülebilir dosya UPX kullanılarak sıkıştırılır

Sistemi parmak izi almak için bilgi toplar (MachineGuid, DigitalProductId, SystemBiosDate)

Alternatif Veri Akışı (ADS) oluşturur

Windows başlangıcında otomatik çalıştırma için kendini yükler

Şekil 3-3: Kötü amaçlı yazılım örneğimizin davranışıyla eşleşen malwr.com imzaları

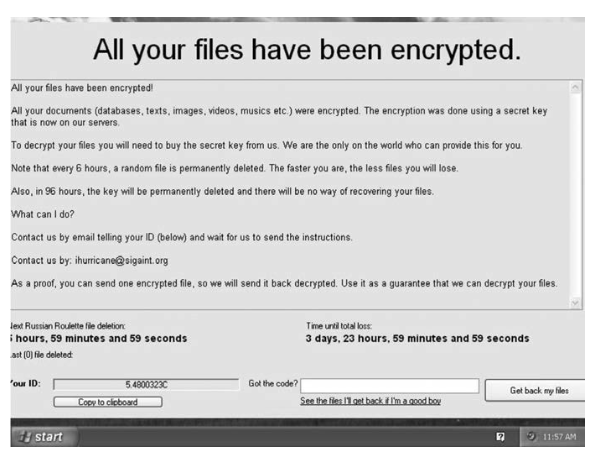
Şekilde gösterilen ilk üç imza, statik analizden kaynaklanır (yani bunlar, kötü amaçlı yazılım dosyasının eylemlerinden değil, kendisinin özelliklerinden kaynaklanır). İlk imza bize, popüler antivirüs toplayıcısı VirusTotal.com'daki bir dizi antivirüs motorunun bu dosyayı kötü amaçlı yazılım olarak işaretlediğini söylüyor. İkincisi, ikilinin sıkıştırılmış veya şifrelenmiş veriler içerdiğini gösterir, bu da karmaşıklığın yaygın bir işareti. Üçüncüsü bize bu ikilinin popüler UPX paketleyicisi ile sıkıştırıldığını söylüyor. Bu statik göstergeler kendi başlarına bize bu dosyanın ne yaptığını söylemese de, bize muhtemelen kötü niyetli olduğunu söylüyorlar. (Rengin statik ve dinamik kategorilere karşılık gelmediğini unutmayın; bunun yerine, her bir kuralın ciddiyetini temsil eder; burada kırmızı - burada daha koyu gri - sarıdan daha şüphelidir.)

Sonraki üç imza dosyanın dinamik analizinden kaynaklanmaktadır. İlk imza, programın sistemin donanımını ve işletim sistemini tanımlamaya çalıştığını gösterir. İkincisi, programın Alternatif Veri Akışları (ADS) olarak bilinen ve kötü amaçlı yazılımın, standart dosya sistemi tarama araçlarını kullanırken görünmez olacak şekilde diskteki verileri gizlemesine izin veren zararlı bir Windows özelliğini kullandığını gösterir. Üçüncü imza, dosyanın Windows kayıt defterini değiştirdiğini ve böylece sistem yeniden başlatıldığında, belirlediği bir programın otomatik olarak yürütüleceğini belirtir. Bu, kullanıcı sistemini her yeniden başlattığında kötü amaçlı yazılımı yeniden başlatır.

Gördüğünüz gibi, bu otomatik olarak tetiklenen imzalar düzeyinde bile dinamik analiz, dosyanın amaçlanan davranışı hakkındaki bilgilerimize önemli ölçüde katkıda bulunur.

->Screenshots Panel:

İmzalar panelinin altında Ekran Görüntüleri paneli bulunur. Bu panel, kötü amaçlı yazılım çalışırken dinamik analiz ortamı masaüstünün ekran görüntüsünü gösterir. Şekil 3-4 bunun neye benzediğinin bir örneğini göstermektedir.

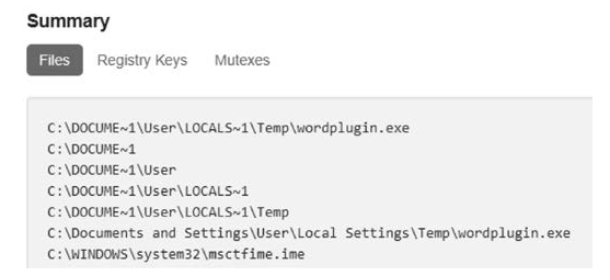


Şekil 3-4: Kötü amaçlı yazılım örneğimizin dinamik davranışının ekran görüntüsü

Karşılaştığımız kötü amaçlı yazılımın, hedefin dosyalarını şifreleyen ve verilerini geri almak istiyorlarsa ödeme yapmaya zorlayan bir tür kötü amaçlı yazılım olan fidye yazılımı olduğunu görebilirsiniz. Kötü amaçlı yazılımımızı basitçe çalıştırarak, tersine mühendisliğe başvurmadan amacını ortaya çıkarabildik.

->Modified System Objects Panel: Değiştirilmiş Sistem Nesneleri Paneli

Ekran görüntüleri altındaki bir satır başlık, kötü amaçlı yazılım örneğinin ağ etkinliğini gösterir. İkili programımız herhangi bir ağ iletişimi ile ilgilenmedi, ancak olsaydı, burada iletişim kurduğu ana bilgisayarları görürdük. Şekil 3-5, Özet panelini gösterir.



Şekil 3-5: Kötü amaçlı yazılım örneğimizin hangi dosyaları değiştirdiğini gösteren Özet bölmesinin Dosyalar sekmesi

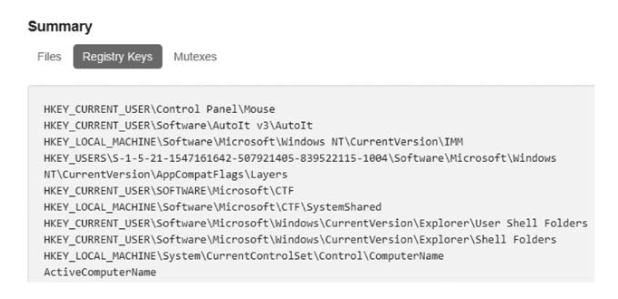
Bu, kötü amaçlı yazılımın dosyalar, kayıt defteri anahtarları ve muteksler gibi hangi sistem nesnelerini değiştirdiğini gösterir.

Şekil 3-6'daki Dosyalar sekmesine bakıldığında, bu kötü amaçlı fidye yazılımının diskteki kullanıcı dosyalarını gerçekten şifrelediği açıktır.



Şekil 3-6: Özet bölmesinin Dosyalar sekmesindeki dosya yolları, örneğimizin fidye yazılımı olduğunu gösterir.

Her dosya yolundan sonra, .locked uzantılı bir dosya gelir ve bunun, değiştirdiği dosyanın şifrelenmiş sürümü olduğu sonucuna varabiliriz. Sonra, Şekil 3-7'de gösterilen Kayıt Anahtarları sekmesine bakacağız.



Şekil 3-7: Kötü amaçlı yazılım örneğimizin hangi kayıt defteri anahtarlarını değiştirdiğini gösteren Özet bölmesinin Kayıt Defteri Anahtarları sekmesi

registery,  Kayıt defteri, Windows'un yapılandırma bilgilerini depolamak için kullandığı bir veritabanıdır. Yapılandırma parametreleri kayıt defteri anahtarları olarak depolanır ve bu anahtarların ilişkili değerleri vardır. Windows dosya sistemindeki dosya yollarına benzer şekilde, kayıt defteri anahtarları ters eğik çizgi ile sınırlandırılmıştır. Malwr.com bize kötü amaçlı yazılımımızın hangi kayıt defteri anahtarlarını değiştirdiğini gösterir. Bu Şekil 3-7'de gösterilmese de, malwr.com'da raporun tamamını görüntülerseniz, kötü amaçlı yazılımımızın değiştirdiği kayda değer bir kayıt defteri anahtarının bir kayıt defteri olan HKEY\_LOCAL\_MACHINE \ SOFTWARE Microsoft \ Windows \ CurrentVersion \ Run olduğunu görmelisiniz. Windows'a bir kullanıcı her oturum açtığında programları çalıştırmasını söyleyen anahtar. Kötü amaçlı yazılımımızın, sistem her açıldığında kötü amaçlı yazılımı yeniden başlatmasını söyleyecek şekilde bu kayıt defterini değiştirmesi çok olasıdır, bu da kötü amaçlı yazılım bulaşmasının yeniden başlatmadan yeniden başlatmaya kadar devam etmesini sağlar. Malwr.com raporundaki Muteksler sekmesi, Şekil 3-8'de gösterildiği gibi kötü amaçlı yazılımın oluşturduğu mutekslerin adlarını içerir.



Şekil 3-8: Kötü amaçlı yazılım örneğimizin oluşturduğu muteksleri gösteren Özet Summary bölmesinin Muteksler sekmesi

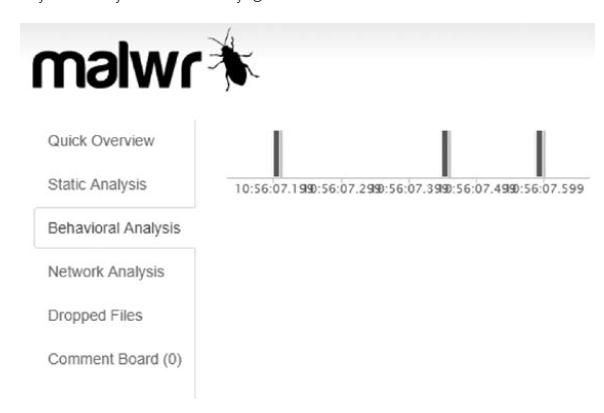
Muteksler, bir programın bazı kaynakları ele geçirdiğini gösteren kilit dosyalarıdır. Kötü amaçlı yazılım, bir sistemi iki kez etkilemesini önlemek için genellikle muteksleri kullanır. En az bir muteksin oluşturulduğu ortaya çıktı

(CTF.TimListCache.FMPDefaultS-1-5-21-1547161642-507921405-839522115-

1004MUTEX.DefaultS-1-5-21-1547161642-507921405-839522115-1004 ShimCacheMutex), güvenlik topluluğu tarafından kötü amaçlı yazılımla ilişkili olduğu bilinmektedir ve burada bu amaca hizmet ediyor olabilir.

->API Call Analysis:

Şekil 3-9'da gösterildiği gibi, malwr.com kullanıcı arayüzünün sol panelindeki Davranış Analizi sekmesine tıklamak, kötü amaçlı yazılım ikili programımızın davranışı hakkında ayrıntılı bilgi getirmelidir. Bu, kötü amaçlı yazılım tarafından başlatılan her işlem tarafından argümanları ve dönüş değerleri ile birlikte hangi API çağrılarının yapıldığını gösterir. Bu bilgileri incelemek zaman alıcıdır ve Windows API'leri hakkında uzman bilgisi gerektirir. Kötü amaçlı yazılım API çağrı analizi hakkında ayrıntılı bir tartışma bu kitabın kapsamı dışında olsa da, daha fazla bilgi edinmek istiyorsanız, etkilerini keşfetmek için tek tek API çağrılarına bakabilirsiniz.



Şekil 3-9: Dinamik yürütme sırasında API çağrılarının ne zaman yapıldığını gösteren kötü amaçlı yazılım örneğimiz için malwr.com raporunun Davranış Analizi bölmesi

Malwr.com, bağımsız kötü amaçlı yazılım örneklerini dinamik olarak analiz etmek için harika bir kaynak olsa da, çok sayıda örnek üzerinde dinamik analiz yapmak için harika değildir. Dinamik bir ortamda çok sayıda örneğin yürütülmesi, makine öğrenimi ve veri analizi için önemlidir, çünkü kötü amaçlı yazılım örneklerinin dinamik yürütme kalıpları arasındaki ilişkileri tanımlar. Dinamik yürütme modellerine göre kötü amaçlı yazılım örneklerini algılayabilen makine öğrenimi sistemleri oluşturmak, binlerce kötü amaçlı yazılım örneğinin çalıştırılmasını gerektirir.

Bu sınırlamaya ek olarak, malwr.com, XML veya JSON gibi makine tarafından ayrıştırılabilir biçimlerde kötü amaçlı yazılım analizi sonuçları sağlamaz. Bu sorunları çözmek için kendi CuckooBox'ınızı kurmalı ve çalıştırmalısınız. Neyse ki, CuckooBox ücretsiz ve açık kaynaklıdır. Aynı zamanda kendi dinamik analiz ortamınızı kurmanız için adım adım talimatlarla birlikte gelir. Http://cuckoosandbox.org/ adresine giderek bunu yapmanızı tavsiye ederim. Artık perde arkasında CuckooBox kullanan malwr.com'dan dinamik kötü amaçlı yazılım sonuçlarını nasıl yorumlayacağınızı anladığınıza göre, CuckooBox'ı kurup çalıştırdıktan sonra CuckooBox sonuçlarını nasıl analiz edeceğinizi de öğreneceksiniz.

!!!!CuckooBox kurmamı istedi

\*Limitations of Basic Dynamic Analysis:Temel Dinamik Analizin Sınırlamaları

Dinamik analiz güçlü bir araçtır, ancak her derde deva kötü amaçlı yazılım analizi değildir. Aslında ciddi sınırlamaları var. Bir sınırlama, kötü amaçlı yazılım yazarlarının CuckooBox ve diğer dinamik analiz çerçevelerinin farkında olmaları ve kötü amaçlı yazılımlarının CuckooBox'ta çalıştığını algıladığında çalışmamasına neden olarak bunları atlatmaya çalışmasıdır. CuckooBox geliştiricileri, kötü amaçlı yazılım yazarlarının bunu yapmaya çalıştıklarının farkındadır, bu nedenle kötü amaçlı yazılımların CuckooBox'tan kurtulma girişimlerini aşmaya çalışırlar. Bu kedi-fare oyunu, bazı kötü amaçlı yazılım örneklerinin dinamik analiz ortamlarında çalıştıklarını kaçınılmaz olarak algılayacağı ve çalıştırmaya çalıştığımızda yürütmeyi başaramayacağı şekilde sürekli oynar. Diğer bir sınırlama, herhangi bir atlatma girişimi olmasa bile, dinamik analizin önemli kötü amaçlı yazılım davranışlarını ortaya çıkarmayabilmesidir. Yürütme sırasında uzak bir sunucuya geri bağlanan ve komutların verilmesini bekleyen bir kötü amaçlı yazılım ikili dosyası durumunu düşünün. Bu komutlar, örneğin, kötü amaçlı yazılım örneğine kurban ana bilgisayardaki belirli dosya türlerini aramasını, tuş vuruşlarını kaydetmesini veya web kamerasını açmasını söyleyebilir. Bu durumda, uzak sunucu komut göndermezse veya artık çalışmıyorsa, bu kötü niyetli davranışların hiçbiri açığa çıkmayacaktır. Bu sınırlamalar nedeniyle, dinamik analiz kötü amaçlı yazılım analizi için tam bir çözüm değildir. Aslında, profesyonel kötü amaçlı yazılım analistleri, mümkün olan en iyi sonuçları elde etmek için dinamik ve statik analizi birleştirir.

\*Özet

Bu bölümde, sonuçları analiz etmek için malwr.com ile bir fidye yazılımı kötü amaçlı yazılım örneğinin dinamik analizini yaptınız. Ayrıca dinamik analizin avantajlarını ve eksikliklerini de öğrendiniz. Artık temel dinamik analizin nasıl yapılacağını öğrendiğinize göre, kötü amaçlı yazılım veri bilimine dalmaya hazırsınız. Bu kitabın geri kalanı, kötü amaçlı yazılım verilerini statik analize dayalı kötü amaçlı yazılım verileri üzerinde gerçekleştirmeye odaklanmaktadır. Statik analize odaklanacağım, çünkü dinamik analize kıyasla iyi sonuçlar almak daha basit ve daha kolay olduğundan, kötü amaçlı yazılım veri bilimiyle ellerinizi kirletmek için iyi bir başlangıç ​​noktasıdır. Bununla birlikte, sonraki her bölümde, veri bilimi yöntemlerini dinamik analize dayalı verilere nasıl uygulayabileceğinizi de açıklayacağım.