ELSEVIER

Sistemler ve Yazılım Dergisi 67 (2003) 109–122

The Journal of Systems and Software

www.elsevier.com/locate/jss

Saldırı tespiti için hafif ajanlar

Q

Guy Helmer, Johnny SK Wong *, Vasant Honavar, Les Miller, Yanxin Wang

Bilgisayar Bilimleri Bölümü, Iowa Eyalet Üniversitesi, 225 Bilgisayar Bilimleri Binası, Ames, IA 50011-1041, ABD

24 Kasım 2000'de alındı; revize edilmiş haliyle 1 Mart 2001'de alındı; 14 Haziran 2002'de kabul edildi

Soyut

Mobil ajanlara dayalı bir saldırı tespit sistemi (IDS) prototipi tasarladık ve uyguladık. Ajanlarımız, dağıtılmış sistemler ağında izlenen sistemler arasında seyahat eder, veri temizleme ajanlarından bilgi alır, bilgileri sınıflandırır ve ilişkilendirir ve aracılar aracılığıyla bilgileri bir kullanıcı arayüzüne ve veritabanına iletir.

Hafif ajan desteğine sahip ajan sistemleri, ajanlara çalışma zamanında yeni özellikler eklenmesine olanak tanır. Çoklu ajan IDS'imizin tasarımını açıklıyoruz ve hafif ajan özelliklerinin IDS'mizdeki mobil ajanlara iletişim ve iş birliği yetenekleri eklememize nasıl olanak tanıdığını gösteriyoruz.

1. Giriş

Güvenli bir bilgisayar sistemi, nesnelerinin (veriler, süreçler veya hizmetler gibi) gizliliği, bütünlüğü ve erişilebilirliği konusunda garantiler sağlar. Ancak sistemler genellikle güvenlik açıklarına yol açan tasarım ve uygulama kusurları içerir. Bir saldırı, bir saldırgan veya saldırgan grubunun güvenlik açıklarından yararlanarak sistemin gizlilik, bütünlük veya erişilebilirlik garantilerini ihlal etmesiyle gerçekleşir. Saldırı tespit sistemleri (IDS'ler), bir dizi saldırıyı tespit eder ve bir saldırı tespit edildiğinde önceden belirlenmiş bir eylemi gerçekleştirir.

Dağıtık bir sistemde izinsiz girişleri tespit etmek (Denning, 1987) zor bir sorundur. IDS'ler, izlenen sistemlere ve ağlara önemli bir ek yük bindirmeden büyük miktarda veriyi analiz etmelidir. Veriler, bilgi işlem sistemine dağıtılmış kaynaklardan elde edilmelidir. İzinsiz girişler, fiziksel bileşenlerden uygulamalara kadar dağıtık sistemin tüm seviyelerinde gerçekleşebilir; her seviyenin izlenmesi gerekebilir. Çeşitli kaynaklardan gelen veriler toplanmalı ve ilişkilendirilmelidir. Bir izinsiz giriş olup olmadığını belirlemek için. İzinsiz girişler mümkün olan en kısa sürede tespit edilmeli, böylece anında ve etkili karşı önlemler alınabilmelidir.

Araştırma grubumuz, dağıtılmış bilgi ağlarını (Honavar vd., 1998) ve veri ambarı tekniklerini bilgi dağıtım sistemlerine uygulamaktadır. Dağıtılmış bilgi ağları, bilgi alma ve çıkarma, veri dönüştürme ve bilgi keşfi için aracılar kullanır. Veri ambarı teknolojileri, veri ve bilginin düzenlenmesi ve heterojen fiziksel olarak dağıtılmış veri ve bilgi kaynaklarından özümsenmesi için kullanılır.

Yazılım ajanları, program kodu ve durumdan oluşur ve bir kullanıcı adına belirli bir özerklikle görevleri yerine getirmek için vardır. Bir yazılım ajanının hedefi, ortamına tepki vermesini, hedefine ulaşmak için planlar yapmasını, faydasını en üst düzeye çıkarmasını ve/veya zaman içinde davranışını değiştirmesini sağlayan bir miktar zeka gerektirebilir (Honavar, 1998). Yazılım ajanları, veri kaynaklarına ulaşmak ve görevlerini uzaktan yürütmek için mobiliteyi kullanabilir, bu da işin doğal bir şekilde dağıtılmasını ve iletişim yükünün azalmasını sağlar.

Hafif ajanlar, temel görevlerini minimum kodla gerçekleştiren ajanlardır. Dinamik olarak güncellenebilir ve yükseltilebilirler, daha küçük, daha basit ve daha hızlı taşınabilirler (daha küçük boyutları nedeniyle). Örneğin, dağıtılmış bilgi işlem sistemlerinde birçok işletim sistemi bulunur. Bir saldırı tespit ajanının bu sistemlerde etkili bir şekilde çalışabilmesi için, geçeceği sistemdeki bilgileri işleyebilmesi ve

Q Kısmen Savunma Bakanlığı tarafından finanse edildi.

* Sorumlu yazar. Tel.: +1-515-294-2586; faks: +1-515-294-0258.

E-posta adresleri: ghelmer@cs.iastate.edu (G. Helmer), wong@cs.iastate.edu (JSK Wong), honavar@cs.iastate.edu (V. Honavar), lmiller@cs.iastate.edu (L. Miller), wangyx@cs.iastate.edu (Y. Wang).

0164-1212/03/$ - ön sayfaya bakınız 2002 Elsevier Inc. Tüm hakları saklıdır.
doi:10.1016/S0164-1212(02)00092-4

sistemde geçerli olacak tespit kurallarını taşıyın
göç edeceği yer. Eğer ajan tasarlanmamışsa
Hafif ajan konseptini kullanarak, her zaman her şeyi taşıyacaktır
tüm sistemler için gerekli olan ve olacak kurallar
daha büyük ve israf kaynakları.

Hafif ajan konseptini akılda tutarak, tasarım
sadelik ve minimalizm üzerine kurulu olacak. Bir ajan
sadece onu hafif yapan temel özellikleri taşıyacak; hedef sisteme
ulaştıktan sonra,
duruma göre gerekli güncelleme ve yükseltmeler yapılır.
Bu tasarım amacını kullanarak, sistem birçok aracının göç etmesi
gerektiğinden kaynak tasarrufu sağlayacaktır.
izlenen sistem. Hafif maddeler,
taşımak için gereken ağ bant genişliği ve CPU süresi
Voyager, tek ticari mobil ajandır
platform şu anda dinamik yükseltmeyi destekliyor
ajan, bu yüzden hafif ajanımız olarak Voyager'ı kullanıyoruz
platform. Hafif ajan tasarımı ayrıca bize
IDS'imize hızlıca özellikler eklemek için.

Mobil aracı, bir kuruluş adına hareket eden bir programdır.
kullanıcı veya başka bir program ve geçiş yapabilme yeteneğine sahip
kendi kontrolü altındaki bir ağda ana bilgisayardan ana bilgisayara.
ajan ne zaman ve nereye göç edeceğini seçer ve
kendi yürütmesini durdurabilir ve başka bir yerde devam edebilir
ağda. Aracı sonuçları ve mesajları döndürür
eş zamanlı olmayan bir biçimde (Ottawa Üniversitesi,
2000). Mobil aracıların, onlarla etkileşim kurmak için uzak hizmetlerle
ağ bağlantısına ihtiyacı yoktur ve
ağ bağlantıları tek seferlik iletim için kullanılır
Veri (aracı ve muhtemelen durumu ve verileri) biçimindeki sonuçlar,
mutlaka veriye geri dönmez.
aynı iletişim yörüngesini kullanan kullanıcı, eğer gerçekten
sonuçların kaynak siteye geri gönderilmesi gerekmektedir.
Alternatif olarak, aracı kendisini başka bir ara düğüme gönderebilir
ve kısmi sonuçlarını da beraberinde götürebilir. Sonuçlar, adresinin
bulunduğu kullanıcıya geri iletilir.
Ajan bilir. Mobil ajan dağıtımı uygular
mimaridir ve uygulamanın en yaygın yolu olan istemci-sunucu
paradigmasından daha iyidir
Dağıtık uygulamalar. Bu modelde, istemci ve sunucu arasında bir ağ
bağlantısı kurulmalıdır.
Bu Paradigma, aşağıdaki durumlarla başa çıkmak için bozulur:
çok dağıtılmış sorunlar, yavaş ve/veya zayıf
kaliteli ağ bağlantıları ve özellikle
Sürekli değişen uygulamaların bakımı.
tek bir merkezi sunucu ve çok sayıda istemciye sahip sistem,
ölçeklenebilirlik sorunu var. Birden fazla sunucu olduğunda
dahil olunca ölçekleme sorunları hızla çoğalıyor,
her müşterinin bağlantıları yönetmesi ve sürdürmesi gerektiğinden
Birden fazla sunucuyla. Son olarak, bir istemci ve bir sunucunun
üzerinde anlaştığı protokol, doğası gereği özel ve statiktir. Genellikle,
sunucudaki belirli prosedürler
Sunucu protokolde kodlanır ve bir parçası haline gelir
arayüz. Belirli veri türü sınıfları,
bu prosedürler ve sonuç özel bir ağdır
Bir uygulama programı arayüzünün sürümü.
yüz genişletilebilir, ancak yalnızca yeniden kodlamanın yüksek maliyetiyle
Protokol sürüm uyumluluğu ve yazılım yükseltmelerini sağlamak için
uygulama.

Mobil aracılar tüm bu doğal sınırlamaların üstesinden gelir
istemci-sunucu paradigmasında. Her şeyden önce,
mobil ajan paradigması, müşteri kavramını paramparça ediyor
ve sunucu. Mobil aracılarla, kontrol akışı aslında ağ üzerinden hareket eder,
istemci-sunucu paradigmasının istek/yanıt mimarisi.
Aslında, her düğüm aracı ağındaki bir sunucudur ve
ajan, bulabileceği yere hareket eder
yürütülmesinin her noktasında çalıştırması gereken hizmetler.

Sunucuların ve bağlantıların ölçeklenmesi daha sonra
birden fazla cihaz arasında gerekli olan karmaşık üstel ölçekleme
olmadan, basit bir kapasite sorunu
sunucular. Kullanıcılar ve sunucular arasındaki ilişki
her bir ajana kodlanmış olarak, parçalara ayrılmak yerine
istemciler ve sunucular. Aracı, sistemi kendisi oluşturur,
ağ veya sistem yöneticilerinden ziyade.
Sunucu yönetimi basit bir mesele haline geliyor
sistemlerin yönetimi ve yerel yükün izlenmesi.

Sağlam ağların sorunu büyük ölçüde azaldı
Çeşitli nedenlerden dolayı. Bağlantı için bekleme süresi, yalnızca aracı
içeri veya dışarı taşımak için gereken süreye düşürülmüştür.
Makineden çıktı. Çünkü ajan kendi
kimlik bilgileri, bağlantı yalnızca bir kanaldır ve
kullanıcı kimlik doğrulamasına bağlı veya sahteciliğe karşı savunmasız. Hayır
istekler bağlantı boyunca akar; aracı kendisi
sadece bir kez hareket eder.

Son olarak ve en önemlisi, aracıların kullanımıyla uygulama
düzeyinde bir protokol oluşturulmaz. Bu nedenle, aracı tabanlı tüm
uygulamalar için uyumluluk sağlanır.
Tam yukarı doğru uyumluluk, ele alınması gereken bir sorun olmaktan
çıkıp norm haline gelir ve yükseltme veya
Bir uygulamanın yeniden yapılandırılması, istemci dağıtımına
bakılmaksızın yapılabilir. Sunucular yükseltilebilir,
hizmetler taşındı, yük dengeleme devreye girdi ve güvenlik
politikanın kesintiye uğramadan veya revizyon yapılmadan uygulanması
ağ ve müşteriler.

Bu nedenle mobil ajanların dağıtılmış mimari için birçok avantajı
vardır. Dağıtılmış mobil otonom
ajanlar saldırı tespitinde kritik sorunları çözer,
bant genişliği, kullanıcı bilgisayarlarındaki işlem döngüleri, verimlilik,
güvenilirlik gibi genel bir bakış açısı sağlarlar
''bileşenleri'' eklemek ve bütünleştirmek için mimari
sisteme. Monolitik, merkezi sistemler
bir kullanımıyla üstesinden gelinebilecek birkaç hata
dağıtılmış mimari. Mobil aracılar,
istemci-sunucu tarafından eklenebilecek zayıflıklar
dağıtılmış mimariye yönelik bir paradigma.

Ağ saldırı dedektörleri genellikle ağ segmentlerine bağlı tek
sensörler kullanır. Ancak, yerel alan
ağlar anahtarlı mimarilere doğru ilerledi
tüm ağa tekli yayın çerçeveleri yayınlamayan
segmentler. Merkezi sensörler, sensörün anahtarlı bir şekilde
bağlanmadığı segmentlerdeki trafiği kaçıracaktır.

çevre. Dağıtılmış aracılar bu sorunu şu şekilde çözer:
her ana bilgisayarda ağ etkinliğinin izlenmesi.

Ağ saldırı dedektörlerinin de sorunları var
yüksek veri hızları. Merkezi sistemler paketleri kaçırabilir
Hızlı veya Gigabit Ethernet'te ağır yük durumlarında
ağlar. Dağıtılmış aracılar, işlemeyi dağıtır
Ağ sistemleri arasındaki çaba ve muhtemelen
izinsiz girişlerin tespit edilme şansını artırmak
merkezi bir IDS tarafından özlenecektir.

Ajan tabanlı IDS'nin modüler mimarisi, diğer projelerden saldırı
tespit bileşenlerinin entegrasyonuna olanak tanır. Örneğin, SNORT

ağ IDS (Roesch, 1999) küçük, hızlı, düşük maliyetli bir
İmza yapılarıyla eşleşen ağ paketlerini izleyen sensör. Ajan tabanlı
IDS, SNORT'u içerebilir
ve her ana bilgisayarda ağ kötüye kullanımı tespiti sağlayın.
Benzer şekilde, çeşitli küçük programlar port tarama, yayın pingleri
(smurf'ler) gibi belirli saldırıları tespit eder.
alışılmadık zamanlarda oturum açma ve dosyalarda değişiklik yapma. Bunlar şunlar olabilir:
Dağıtılmış IDS'ye sarmalayıcı ajanlar aracılığıyla dahil edilmiştir.

Merkezi veya merkezi olarak yönetilen bir IDS,
tek bir başarısızlık noktası ve tek bir saldırı hedefi.
Dağıtılmış aracı mimarisi merkezi bir noktadan kaçınır
başarısızlık. Otonom ajanlar çalışmaya devam edebilir
diğer aracı sunucuların veya diğer arızaların arızalanmasına rağmen
tümün tehlikeye atılmasını önleyen bir sistemde
Bir bileşen arızalansa veya saldırıya uğrasa bile IDS (Mell ve
McLarnon, 1999). Mobil aracılar ayrıca yetenekli olabilir
saldırganlardan kaçmanın ve kendilerini yeniden diriltmenin
öldürüldü.

Dağıtılmış bir aracı tabanlı IDS, mekansal sorunları çözmeye yardımcı olur
birden fazla kişinin saldırı tespitinde sorun yaşadığı durumlarda
ana bilgisayar bir saldırıda yer alıyor. Örneğin, bir
''FTP sıçrama'' saldırısında, bir saldırgan bir ana bilgisayardaki anonim bir
FTP sunucusunu kullanarak bir komutu başka bir ana bilgisayara taklit edebilir.
Hedef ana bilgisayarda uzak kabuk. Her ikisindeki aracılar
anonim FTP sunucusu ve hedef ana bilgisayar bunu tespit edebilir
saldırıdaki mekânsal olarak ayrı olaylar ve bunların korelasyonu
Olayları neredeyse gerçek zamanlı olarak izleyin.

Kumar (1995), IDS'lerin eksikliklerini sıralıyor.
Farklı bir şekilde, eksiklikler bir IDS'de olması istenen özelliklerin bir
listesini sağlar.

Genel Mimari. Yaygın saldırı tespiti

Çerçeve (CIDF, Porras ve diğerleri, 1999), bir IDS için genel bir
mimari belirler ve bir IDS'nin bileşenlerini sınıflandırır. Dağıtılmış

mobil bir sistem

ajanlar IDS'yi CIDF mimarisiyle uyumlu esnek bir şekilde uygular.

Verimlilik. Dağıtılmış bir Çoklu Ajan Sistemi şunları elde eder:

dağıtılmış verilerin uygun düzeylerde denetlenmesi

sistem ve bilgi işlemeyi dağıtır

ve saldırı tespit çabası.

Taşınabilirlik. IDS'ler genellikle bir

Bir kuruluşun güvenlik politikasına yönelim. Kuruluşlar arasındaki

güvenlik politikalarındaki farklılıklar

IDS'nin taşınabilirliğinin olmamasıyla sonuçlanır. Farklı bir şekilde

IDS'nin işletim açısından taşınabilirliği anlamında

sistemleri ve bilgisayar mimarisi de bir sorundur.

Perl ve Java, iki yorumlanan dildir

IDS'ler için taşınabilirlik sağlamak amacıyla kullanılır. Otonom

saldırı tespiti için ajanlar (AAFID, Balasubra-maniyan ve diğerleri,

1998) ve kendi projemiz MAIDS,

Bu tür iki IDS vardır.

Yükseltilebilirlik. Ajan tabanlı bir sistemde mevcut olana benzer

bileşen tabanlı bir mimariye dayalı bir IDS

sistem yükseltilebilirlik ve iyileştirmeyi karşılar

endişe. Yeni özellikler böyle bir sisteme kolayca eklenebilir

sistem.

Bakım. Öğrenilenlerin sürdürülmesi ve güncellenmesi

Bir IDS'nin bileşenleri tarafından kullanılan bilgi, bileşenlerin

mimarisine bağlı olacaktır.

Performans kıyaslamaları. Mevcut IDS'lerin gerçek dünyadaki kapsamlı

nicel performans değerlendirmeleri

ortamlar mevcut değildir. Güvenlik açığı kapsamı

tedarikçiler tarafından güvenlik açığı değerlendirme projelerinin

yardımıyla ele alınmaya başlanıyor, buna

Ortak Güvenlik Açığı Sayımı (Mann,

1999).

Test etme. Kumar, ''saldırı tespit sistemlerini test etmenin kolay bir

yolu yoktur.'' diyor. (Kumar, 1995) MIT

Lincoln Labs saldırı tespit değerlendirmesi (Lippmann ve diğerleri,

1998) ilk büyük testlerden biriydi

araştırma IDS'leri. Saldırıları simüle etmek hâlâ zor,

Sistemlerin etkinliğini değerlendirmek zor

Gerçek dünya yükleri altında çalıştırıldığından, sistemleri

çalıştırmak için önemli ayarlama ve uzmanlık gerekebilir.

Çoklu ajan IDS, diğer saldırılara benzer
tespit sistemlerinin etkinliğinin gerekli olması nedeniyle
gerçek dünya koşullarında gösterilebilir.

Modüler ve genişletilebilir bir yapı inşa etme yaklaşımı
sistem, bir IDS'deki karmaşık sorunların çözülmesine yardımcı olur.
Sorunu bilgi alma, sınıflandırma, işbirliği ve derleme yönlerine ayırır.

Sistemimiz için geri alma yapan ajanlar geliştirildi
dağıtılmış sistemlerden gelen bilgileri sınıflandırır
(gömülü uzman kurallarını veya makine öğrenimini kullanarak)
(teknikler) kullanır ve verileri bir veritabanında depolar.

Bu makalede, tasarımımızın kısa bir özetini inceliyoruz.
dağıtılmış IDS ve ardından ajanın eklenmesini açıklayın
Dinamik toplama kullanarak sisteme iş birliği sağlıyoruz. Dinamik
toplamanın nasıl fayda sağladığını gösteriyoruz.
Mevcut nesneleri genişletmek için kullanışlı bir mekanizmadır ve bize
yeni özellikleri hızla ekleme olanağı sağlar
sistem.

IDS'mizin ilk tasarımında, aracı iletişimi
dikeydi: veri toplama aracı ile ilişkili mobil aracı arasında ve

Kullanıcı arayüzüne kadar her şey var. Ancak bu tasarımda,

ajanlar yatay olarak iletişim kuramadı
Birbirleriyle işbirliği yaparak saldırıları tespit edebilirler.
Yatay iletişim için konsepti ekledik
ajan duyarlılığının, bir ajanın daha fazla olacağı yerde
müdahaleci olaylar yaşandığında olağandışı olaylara karşı duyarlı
Diğer temsilciler tarafından fark edilmiştir. Hassasiyete örnek olarak,
başarısız oturum açma işlemlerini ele alalım. Birkaç oturum açma hatası
tek bir ana bilgisayar normal olabilir, örneğin bir kullanıcı unuttuğunda
şifresini belirledi. Ancak, bir saldırgan şifreyi belirlediğinde
(Belki de sistemimiz tarafından müdahaleci olarak kabul edilen port
tarama yoluyla) bir hedef ana bilgisayara bağlanabilir.
Hedef ana bilgisayara gidin ve birkaç tipik parola deneyin. Bu durumda,
ilk olay ile ilk olay arasında gevşek bir zamansal ilişki vardır
(port tarama) ve ikinci olay (başarısız oturum açma girişimleri). Aracı
hassasiyet seviyeleri gerçek zamanlı bir
normal izin verirken ilgili müdahalelerin korelasyonu,
bireysel olayların alarm tetiklemeden geçmesini sağlıyoruz.
Ajanlara hassasiyet ve yatay iletişim eklemek için dinamik toplama
kullanıldı.

Sistem uygulamamız şu anda şunları içermektedir:

- Bilgiyi elde eden statik veri temizleme maddeleri
  Sistem kayıtlarından, denetim verilerinden ve operasyonel
  istatistiklerden bilgi toplayıp, bilgileri ortak bir formatta sunmak;
- Devam eden işlemleri izleyen ve sınıflandıran düşük seviyeli aracılar
  faaliyetleri yürütmek, olayları sınıflandırmak ve bunlara ilişkin bilgileri
  arabuluculara iletmek;
- İşbirliğini artıran düşük seviyeli aracılar için yönler
  acentelere;
- Saldırı tespiti için tahmini kurallar elde etmek amacıyla makine
  öğrenimini kullanan veri madenciliği ajanları
  sistem kayıtları ve denetim verileri.

Çoklu Aracılı Saldırı Algılama Modeli temel alınmıştır
dağıtılmış bilgi ağları fikri üzerine.
Şekil 1'de gösterilen mimari aşağıdaki katmanları kullanır:

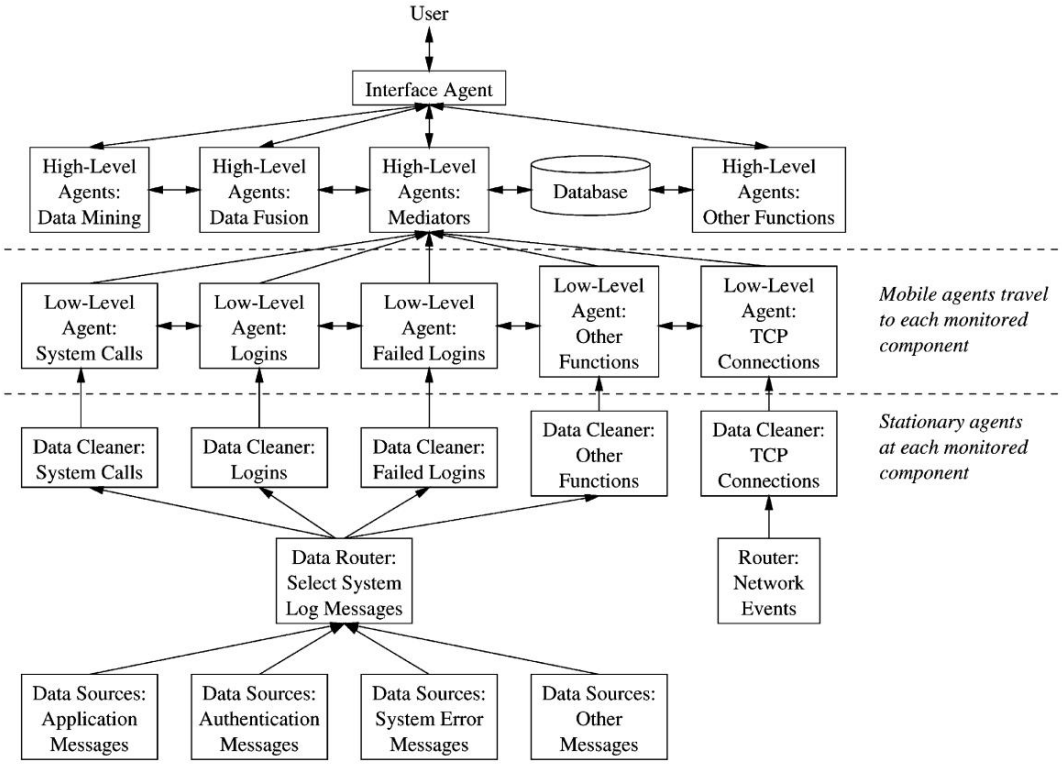
Kullanıcı Arayüzü. Kullanıcı arayüzü, aşağıdakilerin kontrol edilmesini sağlar:
ajanlar, izlenen sistemlerin listesinin yönetimi,
ve izinsiz girişleri bildirir.

Veritabanı. Veritabanı, eğitim ve çevrimdışı saldırı tespiti için verileri
depolar.

Veri birleştirme ve veri madenciliği. Bu düzeydeki aracılar şunları yapabilir:
Alt ajanlardan gelen bilgileri birleştirin ve veritabanından bilgi
çıkarın.

Aracılar. Düşük seviyeli aracılar, aracıların ziyaret ettiği sistemleri
kontrol eden aracılar tarafından yönetilir.
ajanlardan gizli verileri elde edin ve yönlendirin
verileri yerel veritabanına ve kullanıcı arayüzüne aktarır. Sistem
geliştirildikçe, aracılar
bireysel olayları tutarlı bir şekilde birbirine bağlamak için
veritabanındaki verilere veri madenciliği algoritmaları uygulayın
Bir saldırıya karışan unsurların görünümü.

Veri toplama, temel sınıflandırma ve temel veri madenciliği. Mimarinin
ortasında, düşük seviyeli,
Mobil ajanlar, saldırı tespitinde ilk hattı oluşturur. Periyodik olarak
her bir ilişki noktasına seyahat ederler.



Şekil 1. Mobil Ajan Saldırı Algılama Sisteminin Mimarisi.

ated veri temizleme maddeleri, yakın zamanda toplanan verileri elde edin
bilgileri ve verileri belirlemek için sınıflandırmak
tekil müdahalelerin olup olmadığı.
Veri temizleme ve biçimlendirme. Alt kısımda
katmanlı mimari, sistem günlük yönlendiricileri ve sistem
Etkinlik ajanları günlük dosyalarını okur ve sistemlerin işleyişini
izler. Yönlendiriciler, daha önce dağıtılmış veri temizleme
ajanlarına veri besler.
Belirli etkinliklere olan ilgilerini kaydettiler. Hedeflenen
veri temizleme maddeleri, elde edilen verileri işler
yönlendiriciler ve etkinlik aracıları. Verileri şu şekilde işlerler:
ortak veri formatları.

IDS aracı sistemimiz için hiyerarşik mimari
aşağıdaki avantajlara sahiptir:

1. Aracın uygulanması verimlidir. Düşük seviyeli bir aracı izlenen
sisteme ulaştığında, aracı
parçaların seyahat etmesine gerek yoktur. Birçok düşük seviyeli ajan
oluşturulan ve izlenen sisteme aktarılan
aracı parçanın birçok kez oluşturulmasına gerek yoktur
ve geçiş yapmanıza gerek yok. Çok fazla ağ bant genişliği
ve CPU zamanından tasarruf edilir.
2. Katmanlı sistemin tasarımı ve değiştirilmesi kolaydır. Aracıların
açık bir şekilde düzenlenmesi, sistemin bakımını kolaylaştırır.
3. Platform ahlaksızlığı sağlar. Daha düşük seviyeler
Sistem günlükleriyle iletişim kurma gereksinimi platforma
bağlıdır. Yeni bir işletim sistemi eklendiğinde, yalnızca en
düşük seviyedeki aracı veri toplayan araçların eklenmesi gerekir.
eklendi.

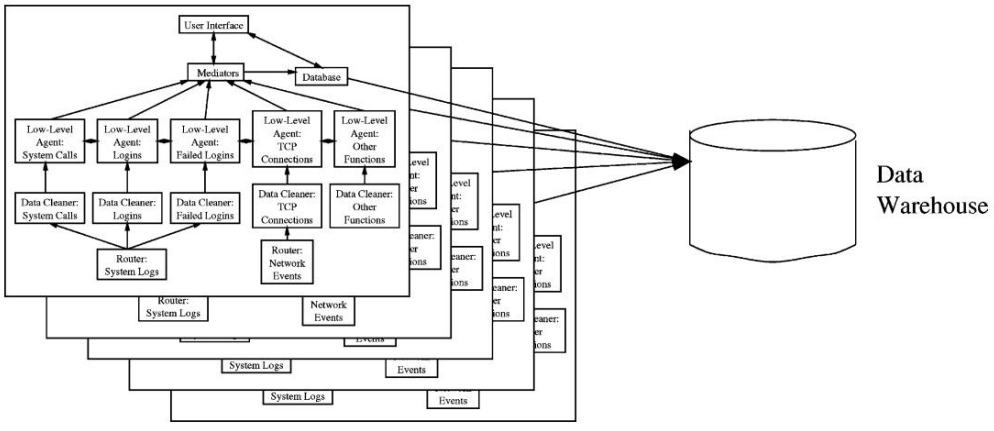
Hiyerarşik mimaride izlenen bileşenlerin ihlalinin
raporlanmasındaki gecikme, hiyerarşik mimariden yararlandığımız
avantajlarla karşılaştırıldığında önemsiz kalacaktır.
mimari. Aracılar hiyerarşik sistem mimarisinde hafif olabilir ve
performans büyük ölçüde
iyileştirildi.

Sistemi daha da geliştirdikçe, birden fazla departman
düzeyindeki sistemler izlenebilecek. Veri ambarı
konut, bilgi ve verileri birleştirmek için kullanılabilir
Saldırıların bireysel departmanlardan kuruluş çapında bir
görünümüne geçişini sağlar. Şekil 1'deki aracı sistemi
Bir kuruluşun departman düzeyinde çalıştırılmak üzere
hedeflenmiştir. Kurum çapında bilgi sağlamak için
saldırılar, her departman temsilcisinden gelen veriler
sistem, gösterildiği gibi bir veri ambarına veri aktaracaktır
Şekil 2.

Çünkü veri ambarı küresel bir veri tabanı sağlayacaktır.
Saldırı tespit sistemlerinin görünümü, desteklemez
sadece saldırıların tespiti değil aynı zamanda:

• yöneticilerin yeni saldırıları keşfetmesine yardımcı olur,
• Sistem yöneticilerine saldırıların nasıl engellendiği konusunda eğitim verir
sistemlerine monte edilmiş ve
• kurumsal bilgideki zayıf noktaları belirler
sistemleri.

Desteklemek için bir dizi aracı altyapısı mevcuttur
Aracı sistemleri. Genellikle, aracı altyapıları şunları sağlar:
aracı sunucular, aracı arayüzleri ve aracı brokerlar.
Aracı sunucular hareketlilik ve kimlik doğrulama sağlayabilir.
Aracı arayüzleri uygulama programları tarafından kullanılır.
Acenteler oluşturmak ve onlarla iletişim kurmak için. Acente brokerleri
adlandırma ve konum hizmetleri sağlar. Prototip
Mobil ajan IDS, Java dilinde oluşturulmuştur
Voyager Nesne İstek Broker'ını (ObjectSpace) kullanarak
Inc., 1999). Voyager, hareketlilik, arayüz ve
ve ajanları uygulamak için gereken adlandırma hizmetleri
IDS. Voyager, kullanımı kolay bir platformdur.
web'de serbestçe kullanılabilir ve %100 Java'dır, bu nedenle
Voyager temel alınarak geliştirilen uygulamalar taşınabilirdir ve
birden fazla işletim sistemiyle uyumludur
ve donanım platformları. Voyager ayrıca tek
ticari mobil aracı platformu şu anda kullandığımız özellik olan
dinamik toplamayı destekliyor
Hafif ajan sistemimizi tasarlamak için.
Bu avantajlardan yararlanmak için ajan platformumuz için Voyager'ı
kullandık.



Şekil 2. Saldırı tespiti için kurumsal veri deposu.

2. İlgili çalışma

2.1. Dağıtılmış saldırı tespit sistemleri

Dağıtılmış saldırı tespit sistemi (DIDS)
Kaliforniya Üniversitesi–Davis (Mukherjee ve diğerleri,
1994) ana bilgisayar ve LAN monitörlerinin bir kombinasyonunu kullandı
Sistem ve ağ etkinliğini gözlemleyin. Merkezi bir yönetici, monitörlerden
bilgi alarak tespit etti.
saldırılar. DIDS, sistemimize benzer şekilde kullanılmıştır
birden fazla, dağıtılmış monitör, ajanlarımıza benzer ve
anormallikleri keşfetmek için yapay zeka algoritmaları
olaylar. DIDS, istihbaratın tamamen merkezi olması ve DIDS'in bunu
kullanmaması bakımından sistemimizden farklıdır.
herhangi bir ajan teknolojisinin.

CIDF (Reilly ve Stillman, 1998) bizimkine benziyor
Proje genel mimarisi. CIDF, Savunma İleri Teknolojiler Ofisi'nden
oluşan bir grup tarafından geliştirilen önerilen bir standarttır.

Projeler Ajansı, Kaliforniya Üniversitesi–Davis, Bilgi Bilimleri Enstitüsü,
Odessey Araştırma ve
diğerleri. CIDF isimlendirmesi, keşif ajanlarını da kapsayacak şekilde
bizimkinden farklıdır.
Veri toplama ajanlarımız, analiz ajanlarımız düşük seviyeli ajanlarımıza
ve yönlerimize karşılık gelir ve karar-yanıt ajanlarımız yüksek seviyeli
ajanlarımıza karşılık gelir
ajanlar. Genel olarak, CIDF bir model sağlar
Sistemimizi karşılaştırabileceğimiz ancak özellikle ajanlarla ilgili
olmayan bir sistem.

2.2. Saldırı tespit sistemlerindeki ajanlar

Çeşitli projeler, ajanların kullanımını araştırdı
Saldırı tespit sistemleri. Bilgisayar İmmünoloji Projesi, meta öğrenme
için Java ajanları (JAM) ve
AAFID projelerinin her biri sorunu farklı şekillerde inceledi
yollar.

AAFID projesi (Balasubramaniyan ve diğerleri, 1998)
Purdues COAST projesinde esnek, dağıtılmış bir IDS bulunmaktadır
MAIDS tasarımına benzer bir altyapı. Hızlı erişim için Perl aracıları
kullanan bir IDS geliştirildi.
prototipleme ve platformlar arası uyumluluk.
Aracılar arasındaki iletişim tesadüfidir.
Proje ayrıca, saldırı tespitine yönelik ajan tabanlı yaklaşımı da analiz
etti. Yaklaşımımız, vurgulama açısından farklılık göstermektedir.
öğrenme algoritmalarının kullanımı, veri ambarı,
ve mobil ajanlar. Sistemimiz Java dilinde uygulanmıştır.

Üniversitedeki Bilgisayar İmmünoloji Projesi
New Mexico'nun (Warrender ve diğerleri, 1999; Forrest ve diğerleri,
1997, 1996) fikirlere dayalı IDS tasarımlarını araştırdı
hayvanların bağışıklık sistemlerini inceleyerek elde edilen bilgiler. Küçük,
bireysel ajanlar dağıtılmış bir sistemde dolaşacaktı,
izinsiz girişleri tespit edin ve izinsiz girişleri çözün.
Projenin bir kısmı, güvenlikle ilgili bilgisayar programlarını
gözlemleyerek bir öz benlik duygusu geliştirdi
programlar tarafından yürütülen normal sistem çağrı kümeleri.
Bu benlik duygusu, izinsiz girişleri tespit etmek için kullanılabilir
bir programın alışılmadık bir diziyi çalıştırdığını keşfetmek
sistem çağrıları. Bilgisayar İmmünoloji Projesi farklıdır
projemizden, bireysel aracılara odaklanmalarıyla
Çoklu ajanların işbirliği yaptığı entegre bir sistemden ziyade.

Columbia Üniversitesi'ndeki JAM Projesi (Lee ve
Stolfo, 1998; Stolfo ve diğerleri, 1997) akıllı, dağıtılmış Java ajanlarını
ve veri madenciliğini kullanarak modellerin öğrenilmesini sağlar.
paylaşılabilen dolandırıcılık ve müdahaleci davranışlar
Projemiz, odaklandığı konu bakımından farklılık göstermektedir.
tek bir organizasyon ve onun içinde saldırı tespiti
departman verilerini bir araya getirmek için veri ambarı kullanımı
ve kuruluş çapında güvenlik bilgilerine ilişkin görünümler sağlar.

3. Hafif ajanlar

Dinamik toplama, çalışma zamanında yeni öğelerin eklenmesine olanak tanır
Aracılara yetenekler. Bu bölüm, eklemeyi açıklıyor
Dinamik toplama kullanılarak sisteme ajan işbirliğinin sağlanması.

Küçük, minimal ajanlara hafif diyoruz çünkü
aracılar, asgari düzeyde işlevselliği uygular,
tüm işlevleri içeren ağır siklet ajanlara karşı
ihtiyaç duyulabilecek bir şey. Ağır sıklet ile karşılaştırıldığında
ajanlar, hafif ajanlar şunlardır:

- Daha küçük,
- Daha basit,
- Daha hızlı taşınırlar (daha küçük boyutlarından dolayı),
- Dinamik olarak güncellenebilir ve yükseltilebilir.

Dinamik toplama, hafif mobil aracılara işbirliği yetenekleri
eklememize olanak sağladı ve
Sisteme hızla yeni özellikler ekleyin. Özel
Dinamik toplamanın kullanımı, IDS'mizdeki ajanların
Birbirimizi müdahaleci faaliyetler hakkında bilgilendirmek.

Her bir aracı, kendi yönetimi için dinamik toplamayı kullanır
Hassasiyet seviyesi. Hassasiyet seviyesi,
Bir ajanın normal koşullar altında müdahaleci olarak kabul edilmeyebilecek
ancak müdahaleci olarak kabul edilmeyen olaylara karşı hassas olması
İlgili müdahalelerin varlığında müdahaleci olabilir.

Hassasiyet sorununa bir örnek, şu sorundur:
Başarısız oturum açma girişimleri. Tek bir oturum açma girişiminde birkaç oturum açma hatası
Örneğin, bir kullanıcı ana bilgisayarını unuttuğunda, ana bilgisayar normal olabilir.
şifre. Ancak, bir saldırgan bir şifreyi tespit ettiğinde
hedef ana bilgisayara (belki de sistemimiz tarafından müdahaleci
olarak kabul edilen bağlantı noktası taraması yoluyla) bağlanabilir.
Hedef ana bilgisayara gidin ve birkaç tipik parola deneyin. Bu durumda,
ilk olay ile ilk olay arasında gevşek bir zamansal ilişki vardır
(port tarama) ve ikinci olay (başarısız oturum açma girişimleri). Aracı
hassasiyet seviyeleri gerçek zamanlı bir
normal izin verirken ilgili müdahalelerin korelasyonu,
bireysel olayların alarm tetiklemeden geçmesini sağlar.

3.1. Dinamik toplama ile aracı yeteneklerinin eklenmesi

Nesnelerin dinamik olarak toplanması üç avantaj sağlar (ObjectSpace Inc., 1999).

- Yalnızca ikili bir nesnenin davranışı şu şekilde olabilir: ilgilendi.
- Bir nesne belirli şekillerde özelleştirilebilir.
- Bir nesnenin davranışı, çalışma zamanında genişletilebilir. Derleme zamanında belirtilmesi gerekmeyen yollar.

Bu faydalar listesine şunları da ekliyoruz:

- Bir nesne, yeni bir işlevselliğe ihtiyaç duyulana kadar olabildiğince küçük olmalıdır.

Dinamikte kullanılan nesneler için Voyager terminolojisi Toplama, ''yöntemlerdir''. Voyager, birincil nesneyi ve onun yönlerini, yönetilen bir ''toplam'' olarak tanımlar Voyagers facet seçim kuralları, belirli bir nesneye özel bir facet uygulamasını kolayca seçer. Bu, bir nesnenin belirli bir şekilde özelleştirilmesine olanak tanır. nesne.

Voyager, bir facet tabanlı uygulamanın bir versiyonunu seçer Birincil nesnenin facet adı ve sınıf adına göre. Belirli bir sınıfın bir faceti yeniden talep edildiğinde, Voyager, onu uygulayan bir facet arar. facet sınıflarının adının önüne I eklenmiş bir arayüz. Örneğin, AFacet adlı faset olsaydı İstendiğinde, Voyager, IAFacet arayüzünü uygulayan bir sınıf arayacaktır.

Voyager, nesneleri uygulamalar için arar birincil nesnenin adını kullanarak facets arayüzü Facet adı eklenir. Eşleşen bir uygulama bulunamazsa, Voyager aramayı şu şekilde yapar: birincil nesneler üst sınıflardır. Örneğin, birincil nesne aClass sınıfından olsaydı (ki bu da java.lang.Object) ve AFacet faceti talep edildi, Voyager ilk önce şu adı taşıyan bir yönü arayacaktır: aClassAFacet, ardından ObjectAFacet ve son olarak AFaset.

Voyager bir uygulama aradığında facet, birincil nesne paketini ve facets paketi.

3.2. Dinamik toplamanın kullanımı

Çoklu Ajan Saldırı Algılama Sistemindeki akıllı ajanlar, yönlendirilmiş olmaları bakımından hafiftir. veri toplama ve sınıflandırmaya yöneliktir. Düşük seviyeli ajanlar yalnızca doğrudan kendileriyle iletişim kurarlar ilgili veri toplama aracıları ve aracıları. Ekleme IDS'deki düşük seviyeli aracı iletişimi, aracıların İlgili verileri gerçek zamanlı olarak birleştirmek ve bunlardan yararlanmak Sistemdeki ilgili bileşenlerin güvenlik durumu hakkında bilgi sağlar. Dinamik toplama, Düşük seviyeli aracılar arasında, aracılara fazla yük bindirmeden iletişim eklemenin kullanışlı bir yolu kendileri.

Aracılar arası iletişim, duyarlılık yönlerinin düşük seviyeli bir bütün olarak kullanılması ajanlar. Hassasiyet yönleri, bir nesne ailesidir kendi aralarında saldırı bilgilerini iletmek ve ilgili izinsiz girişler hakkında bilgi kullanarak etki yaratmak Gelecekteki ihlallerle ilgili kararlar.

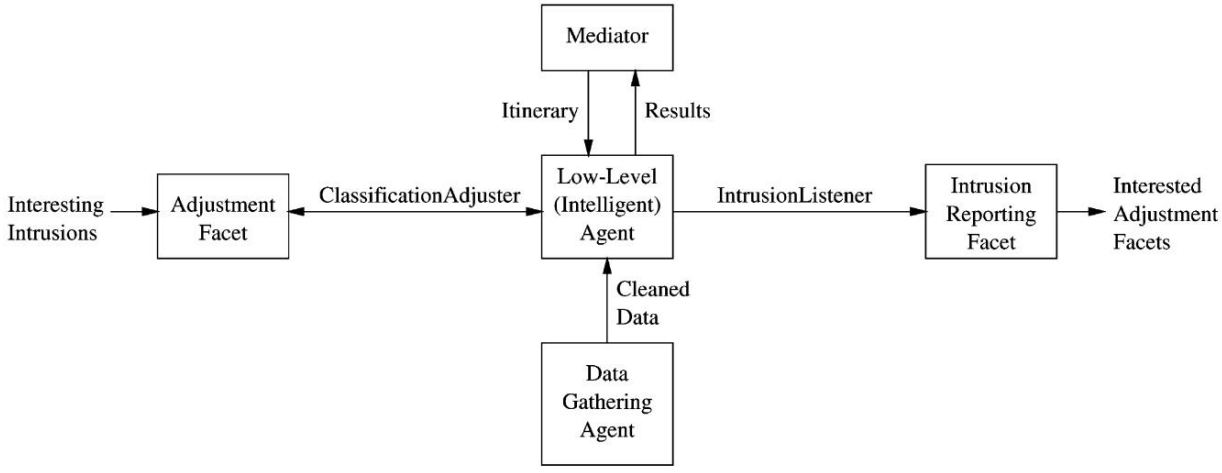
Başarısız giriş denemeleri sorunu iyi bir sorun teşkil ediyor Hassasiyet yönlerinin nasıl kullanıldığına dair bir örnek. Birkaç örnek Dağıtılmış bir sistemde başarısız oturum açmalar, kullanıcıların şifreleri unutmaya eğilimli, yanlış şifreleri denemeye eğilimli Giriş yapmaya çalışırken yanlış sistemler veya yanlış yazımlar kullanıyorlar. Ancak, Bir saldırı karşısında, birkaç başarısız oturum açma işlemi sinyal verebilir Bir saldırganın, tanımladıktan sonra yaygın olarak kullanılan veya varsayılan parolaları denemesiyle gerçekleşen bir saldırının bir sonraki adımı Ağdan sanal terminal veya dosya aktarım bağlantılarına izin veren sistemler. Başarısız oturum açma duyarlılığı Facetler, alışılmadık ağ bağlantı olaylarını dinler (saldırganın hedeflerini belirlemesiyle ortaya çıkan) kaynaklar, bu olayları hatırlar ve bir süreliğine başarısız oturum açmalara karşı duyarlılıklarını artırır.

IDS'de ajan işbirliği geliştirilmedi ajanlar tasarlanıp geliştirilinceye kadar ve test edildi. Dinamik toplama, bir yol sağladı ajanlar aşırı yüklenmeye gerek kalmadan genişletilebilir yeni özelliklere sahip bir ajan. Ancak, ajanların ihtiyaç duyduğu Hassasiyet yönlerini karşılamak için biraz yeniden tasarım. Her bir ayarlama yönü, olayların gerçekleşip gerçekleşmeyeceğine ilişkin kararları etkilemek için kendisini birincil etkenine entegre etmelidir. müdahaleci. Aracılar, yeni elde edilen olayları bir dinleyiciye (yani ayarlama) sağlamak üzere değiştirildi Olayların izinsiz giriş sınıflandırmasının ayarlanması için (yönetim) kullanıldı. Aracılar ayrıca bir dinleyiciyi bilgilendirmek üzere de değiştirildi (raporlama yönü) bir olayın tespit edilmesi durumunda müdahaleci. Aracı mimarisine yapılan bu ayarlama acenteleri çok daha esnek ve geleceğe açık hale getirir toplama yoluyla geliştirme.

Raporlama yönleri, birincil aracılarından diğer ilgili yönlere izinsiz giriş mesajlarını yayınlamak üzere tasarlanmıştır. Ayarlama yönleri ise izinsiz girişleri dinler. raporlama yönlerinden gelen mesajlar. Şekil 3, ajanlar ve yönler arasındaki bilgi akışı.

3.3. IDS'de İletişim

Denetim bilgileri acenteler arasında değiştirilmektedir Audit sınıfının alt sınıflarını kullanarak. Audit sınıfı, bir denetim nesnesini bir tabloda serileştirmek için yöntemler sağlar. anahtarlar ve değerler, bir denetim nesnesinin seri durumdan çıkarılması, bir bir veritabanındaki denetim nesnesini ve bir denetim nesnesini elde etme Bir veritabanından. Audit sınıfının alt sınıfları, oturum açma, önemli dosyalardaki değişiklikler gibi olayları tam olarak açıklar. ağ bağlantıları ve yürütülen işlemler. Düşük seviyeli Temsilciler, denetim nesnelerini veri toplama temsilcilerinden alır



Şekil 3. Düşük seviyeli ajanlar için yönler.

ve saldırı tespitinin ilk seviyesini kullanarak uygulayın
yapay zeka veya uzman kuralları. Denetim nesneleri
IDS'den dikey olarak geçirilir (bkz. Şekil 1).

Saldırı bilgisi, yönler arasında değiştirilir
ASN.l sözdiziminde tanımlayıcı dizeler kullanarak.
Tim Bass (Bass, 1999) tarafından savunulan, izinsiz girişler hakkındaki
bilgileri kapsüllemek için küçük bir yönetim bilgi tabanı (MIB)
tasarlanmıştır. MIB,
saldırı bilgilerini iletmek için düşük seviyeli ajanlar
daha yüksek, daha soyut bir düzeyde ve daha fazlasını tanımlayın
Saldırı hakkında bilgi. ASN.l örneği.
saldırıların hiyerarşisi şöyledir.

Ana bilgisayarda tespit edilen saldırılar
host.priv_prog Ayrıcalıklı bir sunucuya yapılan izinsiz girişler
gram
host.auth Kimlik doğrulamayla ilgili saldırılar
Ağ hizmetleriyle ilgili net saldırılar
net.ip IP ağ protokolü üzerinden yapılan izinsiz girişler
net.ip.icmp ICMP protokolü üzerinden yapılan izinsiz girişler
net.ip.udp UDP protokolü üzerinden yapılan izinsiz girişler
net.ip.udp.service ¼ nfs NFS üzerinden izinsiz girişler
net.ip.tcp TCP protokolü üzerinden yapılan izinsiz girişler
net.ip.tcp.service ¼ login rlogin üzerinden yapılan izinsiz girişler
protokol

Dinleyiciler, herhangi bir bölüme ilgi gösterebilirler.
bir önek belirterek hiyerarşiyi genişletin. Örneğin, bir dinleyici
tümünü dinlemek için net.ip.tcp önekini kaydedebilirsiniz
TCP ile ilgili saldırılar.

3.4. Yönler arası iletişim

IDS'deki her bir ajan tarafından tespit edilen saldırılar
İlgili müdahaleler açısından incelendi. Bu tür ilişkilere örnekler Tablo 1'de gösterilmektedir.

Raporlama yönleri, daha önce açıklanan MIB'de saldırı bilgilerini
kodlamak için oluşturulmuştur ve
Raporları ilgili taraflara gönderin. Tarafları ayarlayın.
İlgili müdahalelere olan ilgiyi kaydederek müdahaleler arasındaki
ilişkileri kodlayın. Ayarlama yönleri
Alınan izinsiz giriş mesajlarını yorumlayarak kendi etki alanlarını etkilemek
hassasiyet seviyesini ayarlayın ve hassasiyet seviyelerini kullanarak ayarlayın
Olayların ilgili düşük seviyeli aracı aracılığıyla izinsiz giriş
sınıflandırmaları.

3.5. Düşük seviyeli aracı tarafından kullanılan makine öğrenimi yöntemi

Bu projede, dağıtılmış akıllı aracı kullanıyoruz
saldırı tespiti. Daha düşük seviyeli bir ajan katmanı, sadece
Sistem mimarisindeki veri temizleme ajanlarının üstünde, saldırı
tespitinin ilk seviyesini oluştururlar.
Hafif mobil ajan teknolojisi, bu ajanlar seyahat ediyor
her birine ilişkin veri temizleme maddelerini toplayın
son bilgileri ve verileri belirlemek için sınıflandırmak
şüpheli bir faaliyetin gerçekleşip gerçekleşmediğini tespit etmek için. Temsilciler
çeşitli sınıflandırma algoritmalarını kullanabilme yeteneği,
hangisinin seçileceği veriye bağlı olacaktır.

Projemizde çeşitli veri temizleme ve düşük seviyeli
ajanlar uygulandı. Bu bölümdeki araştırmamız
ayrıcalıklı programları izleyen aracılara odaklanır

Tablo 1
Ajanlar ve bunlarla ilgili müdahaleler

Ajan	İlgili izinsiz giriş	Gerekçe
Güvenilmeyen bağlantı; NetTCP	Ağ daemon'una saldırı; Yapılandırma dosyası değiştirildi	Saldırgan genellikle ağ üzerinden oturum açmayı dener bir ağ daemon'una saldırmak veya bir yapılandırma dosyasını değiştirmek
Kritik dosyalar	Ağ daemon'una saldırı; NFS	Saldırgan, daha fazla saldırıya olanak sağlamak için kritik bir dosyayı değiştirebilir
Başarısız girişler	Port taraması	Saldırgan, makineleri bulmak için kullanılabilir ağ bağlantı noktalarını tarayabilir hangisine giriş yapmayı deneyebilirim

ve günlük kaydına dayalı saldırıyı tespit etmek için algoritmalar
ayrıcalıklı programların verileri. Ayrıcalıklı programların verileri.
dağıtılmış bilgi işlem sistemlerinde ağ hizmeti genellikle
özel ayrıcalıklarla çalıştırın. Örneğin, popüler
sendmail mail transfer programı süper kullanıcı ile çalışır
UNIX sistemlerinde ayrıcalıklar. Forrests projesi
New Mexico Üniversitesi (Forrest ve diğerleri, 1996) normal ve

Send-mail gibi ayrıcalıklı programların anormal kullanımları. Forrest
sistem çağrısı verileri, bir dizi dosyadan oluşur.
bir işlem kimlik numarası ve sistem çağrısı veren satırlar
Dosyalar, dosya olup olmadıklarına göre bölümlere ayrılır.
normal veya anormal kullanım davranışı göstermek
SunOS 4.1 üzerinde çalışan ayrıcalıklı sendmail programı.
Forrest, bağlam sağlamak için sistem çağrı izlerini dizi pencerelerine
düzenledi ve bir veritabanının
bilinen iyi dizi pencereleri bir
makul büyüklükte müdahaleci olmayan sendmail yürütme kümesi.
Forrest daha sonra müdahaleci davranışın
sistem çağrısının yüzdesinin bulunmasıyla belirlenebilir
bilinen iyi dizilerden hiçbirine uymayan diziler
diziler. Forrests ile aynı veri setini kullanıyoruz
İlgili makalelerde kullanılan tekniklerle karşılaştırmayı mümkün
kılar (Lee ve Stolfo, 1998; Warrender ve diğerleri, 1999).

Geliştiren bir özellik vektör tekniği sunuyoruz
Forrest'ın tekniğine bağlı çünkü bu bir şeye bağlı değil
anormal dizilerin eşik yüzdesi. Özellik vektör tekniğimiz, geniş kapsamlı

her süreçten elde edilen veriler, daha uzun vadeli
Verilerin referans ve analiz için depolanması. Diğer kural öğrenme
teknikleriyle karşılaştırıldığında, tekniğimiz
kolayca taşınabilen kompakt bir kural kümesi oluşturur
Hafif maddeler. Tekniğimiz ayrıca madenciliği de destekleyebilir
Verilerden analiz edilebilecek bir şekilde bilgi elde etmek
uzmanlar tarafından.

Bu konudaki çalışmalarımız hakkında ayrıntılı bilgi için lütfen bkz.
Bu makalelerimize (Helmer ve diğerleri, 1998, 1999)

3.6. Dağıtılmış saldırı tespiti için hafif ajanlar

Dağıtılmış bilgi ağı, bilgiye erişim, bilgiyi organize etme,
dönüştürme ve işleme için hesaplamalı araçlar içerir.
heterojen, dağıtılmış içeriklerin analizi
veri ve bilgi kaynakları ve dağıtılmış problemler için
çözme ve karar verme. Mevcut tasarımımız
dağıtılmış bilgi ağı aşağıdakilerden oluşur
bileşenler: mobil bir aracı altyapısı; akıllı
Çoklu ajan sistemleri için bilgi çıkarma, saldırı sınıflandırması,
koordinasyon ve kontrol mekanizmaları ve dağıtılmış saldırı tespiti
için ajan
yazılım hata ağacı (SFT) ve renkli petri ağı (CPN) üzerine
modeli.

Heterojen veri kaynaklarından gelen saldırı verileri, birden fazla
donanım platformunda ve işletim sisteminde bulunur.
farklı coğrafi konumlardaki sistemler. Bu,
birlikte çalışabilirlik için sağlam ve esnek bir çerçeve
çeşitli veri kaynakları ve istemciler arasında iyi
zamansal ve mekansal olanı anlama yöntemleri
Dağıtılmış verilerin ilişkisi.

Dağıtılmış saldırı bilgilerinin özümsenmesi ve işlenmesi, dağıtılmış
verilerin zamansal ve mekansal ilişkisinin anlaşılması ve ilişkilendirilmesi
için
Saldırı olaylarında iki yöntem kullanıyoruz:

İlk yöntem, iletişimi eklemektir
düşük seviyeli aracılar, aracıların ilgili bilgileri birleştirmesine ve
sistemdeki ilgili bileşenlerin güvenlik durumu hakkındaki bilgiden
yararlanmasına olanak tanır
dinamik toplama ve hassasiyet yönleri. Bu,
Yukarıda 3.1–3.4. Bölümlerde ayrıntılı olarak tanıtılmıştır.

İkinci yöntem, bilgiyi kullanarak birleştirmektir
Anlamayı ve anlamayı mümkün kılan SFT ve CPN modelleri
Saldırı tespiti gereksinimlerini doğru bir şekilde tanımlamak için
gereken alan bilgisinin yakalanması. SFT kullanıyoruz
olayların kombinasyonlarını ve dizilerini modellemek için
hangi izinsiz girişlerin meydana gelebileceğini belirler. Ardından SFT modelleri
IDS'deki dedektörler için CPN tasarımları oluşturmak amacıyla izinsiz
girişler kullanılır. CPN dedektör modelleri daha sonra
mobil aracılar olarak uygulamaya eşlendi
Dağıtılmış IDS. Ayrıntılar için lütfen bu dokümanlarımıza bakın.
makaleler (Helmer ve ark., 2001, yayımlanmak üzere sunulmuştur).

Mobil ajan saldırı tespit mimarisi
verimli, dağıtılmış bilgi füzyonunu ve müdahaleyi mümkün kılar
Olay korelasyonunu sağlar ve yanlış alarmları etkili bir şekilde azaltır.

4. Uygulama

Bu bölümde, genel uygulama
prototip IDS tartışılıyor. Ardından,
iletişimleri eklemek için dinamik toplama
ajanlar sunulmaktadır.

4.1. IDS uygulaması

Prototip IDS, Suns kullanılarak uygulandı
Java Geliştirme Kiti sürüm 1.1 (Sun Microsystems,
2000) ve ObjectSpaces Voyager Nesne İstek Broker sürüm 3
(ObjectSpace Inc., 1999).

Platform bağımsızlığı, güvenliği ve hızı nedeniyle geliştirme dili
olarak Java seçildi
geliştirmenin. Java'nın platform bağımsızlığı, tamamen paylaşılan
kodu derlememize ve çalıştırmamıza olanak sağladı
Silicon Gra-phics IRIX ve Hewlett-Packards HP-UX ticari
platformları da dahil olmak üzere çeşitli platformlardan herhangi birinde
işletim sistemleri ve ücretsiz işletim sistemleri dahil
Ücretsiz BSD ve Linux. Java'nın güvenlik özellikleri şunları içerir:
güvenilmeyen kodu çalıştırmak için deneme ortamları, sıkı tipleme,
sınır denetimi, bayt kodu doğrulaması ve kod
imzalama. Java'nın katı tiplemesi ve nesne yönelimi, yapıyı zorlayarak
geliştirmeye yardımcı oldu ve
mevcut nesnelerin genişletilmesiyle işlevselliğin eklenmesi,
ve hata ayıklama için gereken süreyi azaltarak

C dilinde uygulanmış benzer karmaşıklıktaki projeler.

Java kodunun çalışma zamanı performansı bir endişe kaynağıdır, ancak IDS projesinin önünde bir engel teşkil etmemiştir. Performans, prototip IDS'nin çalışmasını göstermek için yeterli olmuştur. Ek performans gerektiğinde, tam zamanında derleyiciler

Java bayt kodunu yerel makine koduna derleyin gereksinimleri karşıladı.

Voyager Nesne İstek Aracısı seçildi Prototip IDS, kullanılabilirliği, ana dili olarak Java'da geliştirilmesi ve desteği nedeniyle mobil ajanlar. Voyager mobilite, mesajlaşma ve (hedef nesnenin konumundan bağımsız olarak) geçen ve Java nesneleri için adlandırma hizmetleri, bunların hepsi prototip IDS'miz için gereklidir. Voyager ayrıca şunları da destekler: dinamik toplama yoluyla hafif ajanları genişletmek, veya facetler. Facetler, IDS'ye yetenekler eklememize olanak tanır Aracı iletişimi ve iş birliği de dahil olmak üzere aracılar. Ayrıca dinamik olarak karşı önlemler de ekleyebiliriz Aracılara yetenekler. Bu, çalışan bir IDS'ye izin verir yeni saldırılara uyum sağlamak için yeni sistemler uygulamak karşı önlemler.

IDS'deki ajanların alt kademesi, Sabit veri temizleme ajanları. Sabit ajanlar sistem günlükleri de dahil olmak üzere kaynaklardan bilgi edinin, denetim verileri ve operasyonel istatistikler. Sabit ajanlar bilgileri ortak bir biçime dönüştürür daha üst düzey ajanların kullanımı için.

Sabit bir acentenin bir örneği şudur: Sistem günlüklerini okuyan DGFailedLoginAgent Başarısız oturum açma raporları için. Sistem oturumundan gelen başarısız oturum açma mesajlarını Oturum Açma nesnelerine ayrıştırır. başarısız oturum açma işleminin ne zaman gerçekleştiğini, hangi kullanıcının oturum açtığını açıklayın hesap kullanıldı ve hangi bilgisayarlar dahil oldu başarısız girişte.

IDS'deki aracıların orta katmanı, düşük seviyeli aracılar kümesidir. Düşük seviyeli aracılar, mobil aracılardır. Sabit aracılardan bilgi toplar. Düşük seviyeli aracılar, toplanan bilgileri izlemek için işler. ve olayları sınıflandırır. Daha sonra düşük seviyeli ajanlar, arabulucularına bilgi verirler.

Düşük seviyeli bir aracıya örnek olarak, her ana bilgisayarın DGFailedLo-ginAgent'ını ziyaret ederek başarısız oturum açma işlemlerinin son listesini alan FailedLo-ginAgent verilebilir. dağıtılmış tüm ağdaki başarısız oturum açma sayısı sistem kısa bir süre içerisinde bir eşiği aşarsa, Başarısız oturum açmalar, saldırı girişimi olarak işaretlenir.

Prototip, girişimler de dahil olmak üzere saldırıları tespit eder birden fazla makinede birden fazla parola denemesi, alışılmadık TCP ağ bağlantıları, kritik dosyalardaki değişiklikler (Tripwire'ın (Kim ve Spafford) yardımıyla, 1994)), sendmail posta aktarım aracısına yönelik saldırılar, ve güvenli olmayan ağ servislerine bağlantıları reddetti.

Ancak bu bilgi akışının tasarım tamamen dikeydir (Şekil 1 açısından) ve ajanlar her seviye birbiriyle işbirliği yapmıyor veya koordine olmuyor diğer. Mevcut düşük seviyeli ajanlar daha sonra dikkate alınır ''hafif'' ifadesinin, birbirleriyle doğrudan iletişim kurma yeteneğine sahip olmadıkları görüşünden kaynaklandığı belirtiliyor.

4.2. Aracılara işbirliği ekleme

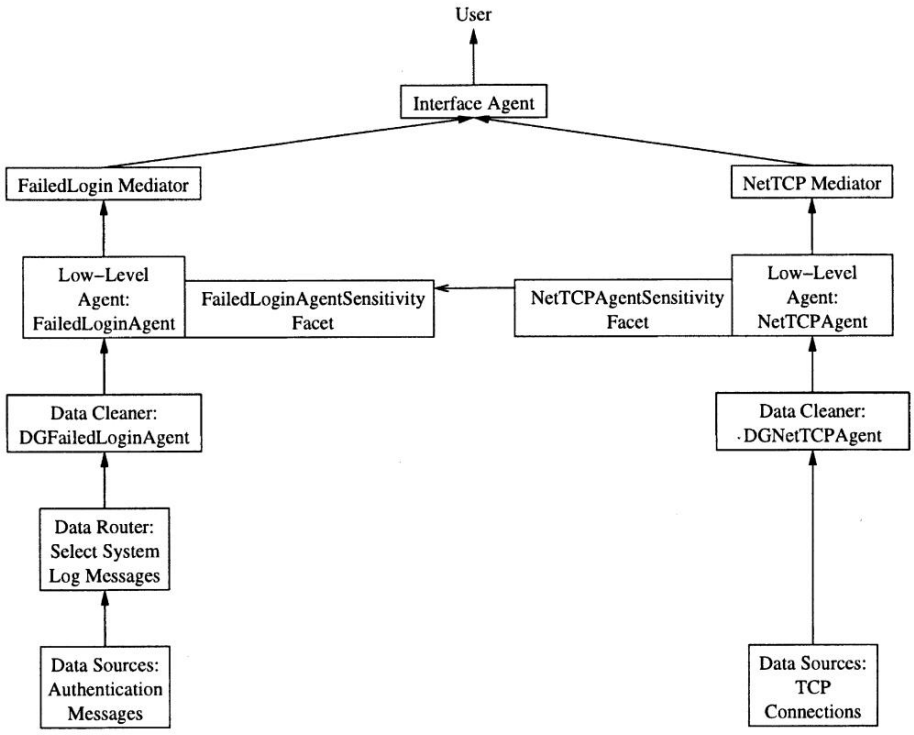
Duyarlılık yönleri, düşük seviyeli aracılara iletişim yetenekleri eklemek için oluşturulmuştur. Duyarlılık Facets, ayarlama ve izinsiz giriş raporlamasını uygular Şekil 3'te gösterilen işlevler. Aracılar aracıları oluşturur ve ajanlara Hassasiyet yönleri ekleyin. Hassasiyet Facets raporlama ve ayarlama işlevlerini uygular Bölüm 3.3'te açıklanmıştır. Aracılar bilgi toplarken, bilgi Hassasiyet yönlerinden geçirilir raporlama ve ayarlama için.

ISensitivity arayüzü şu şekilde tanımlandı: yöntemler:

Kayıt ( ) İlgili etkinliklere abone olmak için ajan,
setIntrusionClasses( ) Diğer ajanlardan gelen ilginç izinsiz girişlere abone olmak için,
getSensitivity( ) Mevcut duyarlılığı elde etmek için ity seviyesi,
sendIntrusionMessage( ) Mesajları yayınlamak için izinsiz girişler hakkında ve
recvIntrusionMessage( ) Saldırı mesajlarının facet tarafından alındığı yer.

Arayüzün varsayılan bir uygulaması olan Sensi-tivity oluşturuldu. register( ) yöntemi uygulaması, bağlı olduğu uygulama tarafından görülen tüm olaylara abone olur. ajan. setIntrusionClasses( ) yöntemi uygulaması, hangi izinsiz girişlerin ilgi çekici olduğunu tanımlar bu facet. getSensitivity( ) yöntemi uygulaması, facet'in ne kadar hassas olduğuna dair bir gösterge döndürür şüpheli olaylara yöneliktir. sendIntrusionMes-sage( ) yöntemi uygulaması, müdahaleci bir olay hakkında diğer dinleme cihazlarına bilgi yayınlar. Hassasiyet yönleri. recvIntrusionMessage( ) yöntem uygulaması, diğer Hassasiyet yönlerinden gelen müdahaleci olaylar hakkında mesajlar alır ve ayarlar Önemine göre yön duyarlılık düzeyi etkinlik.

Şekil 4, Hassasiyetli IDS'nin bir dilimini göstermektedir fasetleri dahil edildi. Belirli bir Hassasiyet faseti, NetTCPAgentSensitivity, temel Dosyalardaki değişiklikler ve arabellek taşma saldırıları dahil olmak üzere müdahaleci etkinlikleri dinlemek için hassasiyet yönü. Diğer aracılar bu müdahaleci faaliyetleri bildirirse, NetTCPAgent'ın hassasiyeti artırılmalıdır. hedef, anormal TCP bağlantıları alabilir yakın gelecekte bu müdahalenin bir parçası olabilir. Çünkü Voyagers'ın bir ajana faset ekleme kuralları, NetTCPAgent'a bir Hassasiyet yönü eklendi,



Şekil 4. Fasetleri olan bir IDS bölümü.

NetTCPAgentSensitivity yönü NetTCPAgent'a eklendi.

FailedLoginAgentSensitivity yönü, NetTCP-Agent'tan gelen izinsiz etkinlik raporlarını dinler. NetTCPAgent, kullanılabilir telnet portlarını taramak gibi bir etkinlik algılarsa, FailedLoginAgentSensitivity yönü şüphe seviyesini yükseltir. Bir saldırgan, kullanılabilir telnet portlarını tarayabilir, ardından keşfedilen telnet portlarına bağlanabilir ve sıklıkla kullanılan parolalarla bilinen hesaplara giriş yapmaya çalışabilir. Bu durumda, FailedLoginAgentSensitivity yönü, kabul edilebilir başarısız oturum açma eşiğini düşürerek saldırının tespit edilmesine yardımcı olur.

4.3. Sonuçlar

Sadece ihtiyaç duyulduğunda hassasiyet yeteneği eklendiğinde, normal durumda ağ üzerindeki yük ve ajanların iletim hızı çok daha iyi olmaktadır.

Özellikle, FailedLoginAgent sınıfı için Java bayt derlenmiş kodu toplam 5665 bayttır (2298 + 3367). Fail-edLoginAgentSensitivity sınıfı için bayt derlenmiş kodu toplam 5547 bayttır (1540 + 4007). FailedLoginAgent'a kalıcı olarak duyarlılık özelliği eklemek, etkenin boyutunu %96 oranında artıracaktır. Yalnızca kodun boyutu dikkate alındığında, FailedLoginAgent, duyarlılık yönü olmadan duyarlılık yönüyle olacağı boyutun yaklaşık yarısı kadardır. Sabit bir iletim hızında, Hafif etkenin tek başına iletilmesi için gereken süre, etkenin faset ile iletilmesi için gereken sürenin yaklaşık yarısı kadardır.

NetTCPAgent için tasarruflar daha az dramatiktir. NetTCPAgent sınıfı için Java bayt derlenmiş kodun toplam boyutu 7434 bayttır (4067 + 3367). NetTCPAgentSensitivity sınıfı için bayt derlenmiş kodun toplam boyutu 4493 bayttır (486 + 4007). NetTCPAgent'a duyarlılık özelliğinin kalıcı olarak eklenmesi, boyutunu %60 artıracaktır.

IDS sistemimiz yalnızca iyi bir verimliliğe ve hızlı tepki süresine sahip olmakla kalmıyor, aynı zamanda yüksek doğruluk ve düşük yanlış alarm oranına da sahip. Birkaç deney gerçekleştirdik.

Öncelikle, alt seviye bir aracı tarafından sendmail sistem çağrılarının sınıflandırılmasını test etmek için bir deney gerçekleştirdik. Temsilcimizi eğitmek için New Mexico Üniversitesi'ndeki Forrests projesinden sistem çağrıları veritabanını kullandık. Sendmail sistem çağrıları için, yöntemimizi kullanarak ortalama yanlış alarm oranı yalnızca %0,83 ve öğrenilen hipotezlerin karmaşıklığı ortalama 8,6 kuraldır. Deneyin ayrıntıları için lütfen diğer makalelerimize (Helmer ve ark., 1998, 1999) bakın.

İkinci olarak, IDS'nin korelasyon yeteneğini test etmek için deneyler yürüttük. Yanlış alarm oranını ve korelasyon yeteneğini test etmek için bazı dağıtılmış saldırılar kullanıldı.
Dağıtık saldırılardan biri FTP sıçrama saldırısıydı.
FTP sıçrama saldırısı, bir saldırganın normalde erişimi olmayan bir ağ bağlantı noktasına veri aktarmak için kullanılabilir. Bu sorundan yararlanmanın bir yolu, verileri FTP sunucusuna güvenen uzak bir kabuk sunucusuna göndermektir.

FTP sunucusu. Hedef FTP sunucusuna güveniyorsa, rsh daemon'ı verileri kullanıcı girişiymiş gibi kabul eder ve verilen komutu yürütür. Deneyi gerçekleştirmek için, bir FTP sunucusuna veri gönderen bir saldırgan makinesi, savunmasız FTP sunucusunu çalıştıran bir aktarma makinesi ve aktarma makinesine güvenen bir rsh daemon'ı çalıştıran bir makine kullandık. Tüm FTP sıçrama saldırıları başarıyla tespit edildi ve bu deneyde hiçbir yanlış alarm bildirilmedi (Helmer ve ark., 2002, yayınlanmak üzere gönderildi).

4.4. Tartışma

Projemizde, hafif dağıtılmış bir aracı kullanarak bir IDS'yi nasıl uygulayabileceğimizi inceledik. Bu makalede, tartışmamızı aracı mimarisi ve aracının yükseltilebilirliği üzerine yoğunlaştırıyoruz.

Güvenlik sorunu, ajan sistemi için büyük bir sorundur. Sistemimiz şu anda güvenlik sorununu dikkate almıyor: Aracı sisteme erişebilen herkes kontrolü ele geçirebilir. Gelecekte bu sorunu ele alacağız.

Yeni keşfedilen dağıtık saldırılar, dağıtık saldırı tespiti için modellenmeli ve IDS'ye entegre edilmelidir. Tekniğimizi kullanarak, yeni dağıtık saldırıyı tanımlamak için bir SFT analizi gerçekleştirilir. SFTA daha sonra tespit araçlarına çevrilir. Elle çeviri zahmetli ve yorucu olduğundan, SFTA'yı tespit araçlarına otomatik olarak çeviren bir sistem tasarladık. Daha fazla bilgi için lütfen şu makalelerimize bakın (Helmer ve ark., 2001, yayınlanmak üzere sunulmuştur).

5. Sonuçlar ve gelecekteki çalışmalar

Hafif ajanların genişletilmesi, IDS'mizde yeni bir iletişim biçimi uygulamak için kullanışlı bir mekanizma sağlar. Dinleme ve raporlama işlevlerini uygulayan özellikler geliştirilerek, mevcut ajan tasarımını veya işleyişini olumsuz etkilemeden IDS'ye önemli bir yeni özellik eklenmiştir. Herhangi bir müdahale olmadığında ajanların boyutundan kaynaklanan sistem yükü azaldığından, sistemin normal çalışma koşullarında işleyişi iyileştirilmiştir.

Aracıların genişletilmesi, IDS'yi genişletmek için birçok olanak sunar. Hafif aracı yeteneklerinin potansiyel bir kullanımı, IDS'deki aracılara veri madenciliği yetenekleri eklemek olabilir. Çeşitli veri madenciliği algoritmalarını uygulayan bir grup faktör oluşturulabilir.

Aracılar daha sonra uygun yönleri ekleyerek daha üst düzey saldırı tespiti için veri madenciliği algoritmalarını dinamik olarak ekleyebilirler.

Saldırılar için prototip MIB, saldırıların bir sınıflandırmasına dayalı olarak resmileştirilmelidir. Saldırıları tanımlamak için ASN.l sözdiziminin kullanılması, IDS'lerin ve iletişimin entegrasyonu için umut verici görünmektedir.

Saldırı tespit ajanları arasında. Ayrıca, saldırılar arasındaki ilişkiler resmi olarak tanımlanmalı ve facetlere kodlanmalıdır. Prototip saldırı ilişkileri tablosu, hassasiyet facetlerimizi geliştirmenin ilk adımıydı. Önceki IDS'ler bu ilişkileri bir uzman sistemindeki kurallar olarak kodluyordu (Mukherjee ve ark., 1994).

Uzman bir sistemde kuralları kodlamaktan ziyade, aracıları ve yönleri kullanmak bu ilişkilerle başa çıkmanın daha dinamik ve esnek bir yolunu sunuyor gibi görünüyor.

IDS sisteminin genişletilmesi, bir saldırının kaynağını gerçek zamanlı olarak belirlemek için veri birleştirmeyi uygulamak üzere facet'ler kullanılarak mümkün olabilir. Bazı saldırılar, saldırganın kaynak IP adresini belirler ve bazen saldırganın kimliğini belirlemeye yardımcı olacak daha fazla bilgi kaynaktan toplanabilir.

Facetlerin kullanımıyla yapılabilecek olası bir diğer genişletme, saldırılara karşı önlem almak olabilir. Saldırı bilgilerini gerçek zamanlı olarak birleştiren yüzeyler tasarlanabilir. Birleştirilen bilgiler belirli kriterleri karşıladığında, yüzeyler, izlenen sisteme saldırıyı caydırmak veya durdurmak için düzeltici veya savunmacı önlemler almasını yönlendirebilir. Saldırının kaynağı biliniyorsa (daha önce bahsedilen tanımlama yüzeyleri aracılığıyla), daha fazla saldırıyı önlemek için karşı önlemler alınabilir.

Facetler, ajanlarıyla bir araya getirildiğinden, karşı önlem facetleri, eylemlerin gerçekleştirilmesi gereken sistemde çalışacaktır. Bir saldırıya karşı koymak için hedef sisteme komut göndermek için ek bir iletişim yükü gerekmeyecektir.

Referanslar

Balasubramaniyan, J., Garcia-Fernandez, JO, Isacoff, D., Spafford, EH, Zamboni, D., 1998. Otonom ajanlar kullanılarak saldırı tespiti için bir mimari. Teknik Rapor COAST TR 98-05, Purdue Üniversitesi, Bilgisayar Bilimleri Bölümü.

Bass, T., 1999. Yeni nesil dağıtılmış saldırı tespit sistemleri için çoklu sensör veri füzyonu. Bildiriler Kitabı, 1999 IRIS Sensör ve Veri Füzyonu Ulusal Sempozyumu, Mayıs.

Denning, DE, 1987. Bir saldırı tespit modeli. IEEE İşlemleri ve Yazılım Mühendisliği Dergisi SE-13 (2), 222–232.

Forrest, S., Hofmeyr, SA, Somayaji, A., 1997. Bilgisayar immünolojisi. CACM Dergisi 40 (10), 88–96.

Forrest, S., Hofmeyr, SA, Somayaji, A., Longstaff, TA, 1996. Unix süreçleri için bir öz benlik duygusu. In: Pro-ssp96, Los Alamitos, CA, ABD, 1996, pub-IEEE-CSP, s. 120–128.

Helmer, G., Wong, J., Slagell, M., Honavar, V., Miller, L., Lutz, R., Wang, Y., 2002. Yazılım hata ağacı ve renkli Petri ağı tabanlı belirtimi, tasarımı ve ajan tabanlı saldırı tespit sistemlerinin uygulanması. IEEE Yazılım Mühendisliği İşlemleri, sunulmuştur.

Helmer, G., Wong, J., Slagell, M., Honavar, V., Miller, L., Lutz, R., 2001. Bir saldırı tespit sisteminin gereksinim analizine yönelik bir yazılım hata ağacı yaklaşımı. Bildiriler, Bilgi Güvenliği için Gereksinim Mühendisliği Sempozyumu. Purdue Üniversitesi, Bilgi Güvencesi ve Güvenliği Eğitim ve Araştırma Merkezi, Mart.

Helmer, G., Wong, JSK, Honavar, V., Miller, L., 1998. Saldırı tespiti için akıllı ajanlar. Bildiriler, IEEE Bilgi Teknolojileri Konferansı, Syracuse, NY, ABD, Eylül, s. 121–124.

Helmer, G., Wong, JSK, Honavar, V., Miller, L., 1999. Saldırı tespiti için genetik algoritma kullanılarak özellik seçimi. Genetik ve Evrimsel Hesaplama Konferansı Bildirileri, Orlando, FL, ABD, Temmuz, s. 1781.

Honavar, V., 1998. Williams, J., Sochats, K. (Ed.), Bilgi Teknolojileri Ansiklopedisi, bölüm Akıllı Aracılar. Marcel Dekker, New York, ABD.

Honavar, V., Miller, L., Wong, JSK, 1998. Dağıtılmış bilgi ağları. Bildiriler, IEEE Bilgi Teknolojileri Konferansı, Syracuse, NY, ABD, Eylül 1998, s. 87–90.

Kim, GH, Spafford, EH, 1994. Tetikleyicilerle ilgili deneyimler: Saldırı tespiti için bütünlük denetleyicilerinin kullanımı. USENIX Derneği, editör, pro-sans94, pub-USENIX:adr, Nisan, pub-USENIX, s. 89–101.

Kumar, S., 1995. Bilgisayar Saldırılarının Sınıflandırılması ve Tespiti. Doktora tezi, Purdue Üniversitesi, West Lafayette, IN, ABD, Ağustos.

Lee, W., Stolfo. S., 1998. Saldırı için veri madenciliği yaklaşımları algılama. In: Pro-ss98, pub-USENIX.

Lippmann, RP, Cunningham, RK, Fried, DJ, Garfinkel, SL, Gorton, AS, Graf, I., Kendall, KR, McClung, DJ, Weber, DJ, Webster, SE, Wyschogrod, D., Zissman, MA,1998. 1998 DARPA/AFRL çevrimdışı saldırı tespit değerlendirmesi. Pro-raid98, Louvain-la-Neuve, Belçika'da.

Mann, DE, Christey, SM 1999. Güvenlik Açıklarının Ortak Bir Şekilde Belirlenmesine Doğru. Güvenlik Açığı Veritabanlarıyla Araştırma Üzerine İkinci Çalıştay, Purdue Üniversitesi, West Lafayette, IN, ABD, Ocak.

Mell, P., McLarnon, M., 1999. Mobil ajan saldırılarına dayanıklı dağıtılmış hiyerarşik saldırı tespit sistemleri. In: Pro-raid99, Purdue, IN, ABD, Eylül.

Mukherjee, B., Todd Heberlein, L., Levitt, KN, 1994. Ağ ihlali tespiti. IEEE Ağı Dergisi, 8 (3), 26–41.

ObjectSpace, Inc., Dallas, TX. ObjectSpace Voyager Temel Teknolojisi Kullanıcı Kılavuzu, 1999. Sürüm 3.0.0.

Porras, P., Schnackenberg, D., Stuart, S.-C., Maureen, S., Wu, F., 1999. Ortak saldırı tespit çerçevesi mimarisi. Çevrimiçi, http://www.gidos.org/drafts/architecture.txt .

Reilly, M., Stillman, M., 1998. Ölçeklenebilir saldırı tespiti için açık altyapı. Bildiriler, 1998 IEEE Bilgi Teknolojileri Konferansı, Syracuse, NY, ABD, IEEE Press, s. 129–133.

Roesch, M., 1999. Snort: Ağlar için hafif saldırı tespiti. In: Pro-lisa99, Seattle, WA, ABD, Kasım, pub-USENIX.

Stolfo, SJ, Prodromidis, AL, Tselepis, S., Lee, W., Fan, D., Chan. PK, 1997. JAM: Dağıtık veritabanları üzerinde meta öğrenme için Java aracıları. Üçüncü Uluslararası Bilgi Keşfi ve Veri Madenciliği Konferansı Bildirileri, Newport Beach, CA, ABD, Ağustos, s. 74–81.

Sun Microsystems. Java Geliştirme Kiti Sürüm 1.1.x. Çevrimiçi, Mayıs 2000. Şuradan edinilebilir: <http://java.sun.com/products/jdk/1.1/ >.

Ottawa Üniversitesi. Akıllı mobil aracılar araştırması. Çevrimiçi, Mayıs 2000. Şuradan edinilebilir: <http://deneb.genie.uottawa.ca/webdata/research/>.

Warrender, C., Forrest, S., Pealmutter, B., 1999. Sistem çağrılarını kullanarak saldırıları tespit etme: alternatif veri modelleri. In: Pro-ssp99, IEEE Press, s. 133–145.

Dr. Guy Helmer, Palisade Systems, Inc.'de Kıdemli Yazılım Mühendisidir ve 2001 Ar-Ge 100 Ödülü sahibi PacketHound da dahil olmak üzere ağ güvenliği ve uygulama protokolü yönetim cihazları tasarlayıp üretmektedir. Bilgisayar Bilimleri alanında doktora derecesine sahiptir. 2000 yılında Iowa Eyalet Üniversitesi'nden, 1998 yılında Iowa Eyalet Üniversitesi'nden Bilgisayar Bilimleri alanında Yüksek Lisans derecesini ve 1989 yılında Güney Dakota Maden ve Teknoloji Okulu'ndan Bilgisayar Bilimleri alanında Lisans derecesini aldı. Guy, 1998, 1999 ve 2000 yıllarında Yahoo!'nun En Çok Bağlantılı Üniversiteleri arasında yer alan Madison, Güney Dakota'daki Dakota Eyalet Üniversitesi'nde yedi yıl boyunca sistem programcısı, ağ mühendisi ve sistem yöneticisi olarak çalıştı. Burada birden fazla kampüsün ağ bağlantısını kurma, Güney Dakota eyalet hükümetinin internete ilk bağlantısını kurma ve bölgedeki ilk kapsamlı yüksek hızlı yurt ağlarından birinin mühendisliğini yapma gibi başarılar elde etti. Araştırma ilgi alanları arasında işletim sistemi ve ağ güvenliği, yüksek performanslı hesaplama, yazılım mühendisliği ve yazılım güvenliği yer almaktadır.

Dr. Johnny SK Wong, ABD, Iowa, Ames'teki Iowa Eyalet Üniversitesi Bilgisayar Bilimleri Bölümü'nde tam profesördür. Araştırma ilgi alanları arasında İşletim Sistemleri, Dağıtık Sistemler, Telekomünikasyon Ağları, Geniş Bant Entegre Hizmetler Dijital Ağları, Eşzamanlılık Kontrolü ve Kurtarma, Multimedya ve Hipermedya Sistemleri, Akıllı Çoklu Ajan Sistemleri ve Saldırı Tespit yer almaktadır. 1983-1986 yılları arasında Tele-com Australia ile araştırma sözleşmeleri kapsamında araştırmacı olarak çalışmış ve ISDN ağ protokollerinin performansını incelemiştir. Bu süre zarfında ISDN iletişim mimarisi ve protokollerinin incelenmesine ve değerlendirilmesine katkıda bulunmuştur. 1989-1990 yılları arasında Des Moines, Iowa'daki Microware Systems Corporation ile bir araştırma sözleşmesi kapsamında Baş Araştırmacı olarak çalışmıştır. Bu sözleşme, ISDN'de Koordineli Multimedya İletişimi çalışmalarını içermektedir. Dr. Wong, 1991 ve 1992 yazlarında Rochester'daki IBM şirketi tarafından desteklendi. IBM'deyken, Uygulama Sistemleri için dağıtılmış bilgi işlem ortamı (DCE) üzerinde çalıştı. Bu, iletişim protokollerinin ve dağıtılmış veritabanı kavramlarının entegrasyonunu içeriyordu. Dr. Wong ayrıca, mühendislik eğitimini geliştirmek amacıyla NSF Sentez Koalisyonu Projesi tarafından finanse edilen Courseware Matris Yazılım Projesi'nde koordineli multimedya sistemi (COMS) üzerinde de yer almaktadır. 1993-1996 yılları arasında, Iowa Enerji Merkezi tarafından finanse edilen, multimedya iletişim teknolojisini kullanarak enerji tasarrufu eğitimi için bilgiye dayalı bir sistem üzerine bir araştırma projesi üzerinde çalışmaktadır. 1995-1996 yılları arasında, Enerji Bakanlığı'na (DOE) bağlı Ames Laboratuvarı tarafından desteklenerek Çoklu Veritabanları için Ara Yazılım sistemi üzerinde çalışmıştır.

Savunma Bakanlığı (DoD) tarafından finanse edilen Saldırı Tespiti ve Önlemler için Akıllı Çoklu Ajanlar ve Mayo Vakfı tarafından finanse edilen Dünya Çapında Ağ Uygulamaları üzerinde veritabanı oluşturma ve X-ışını görüntüleme projelerinde yer aldı. Şu anda, her ikisi de Ulusal Bilim Vakfı (NSF) tarafından finanse edilen CISE Eğitim İnovasyonu: Entegre Güvenlik Müfredat Modülleri ve NSF Bilgi Güvencesi Programı üzerinde çalışmaktadır.

Dr. Vasant Honavar, Hindistan Bangalore Üniversitesi'nden Elektronik Mühendisliği alanında lisans derecesi, Drexel Üniversitesi'nden Elektrik ve Bilgisayar Mühendisliği alanında yüksek lisans derecesi ve Wisconsin Üniversitesi, Madison'dan Bilgisayar Bilimleri alanında yüksek lisans ve doktora derecesi almıştır. Yapay Zeka Araştırma Laboratuvarı'nı (www.cs.ias-tate.edu/~honavar/aigroup.html) kurmuş ve yönetmektedir. Iowa Eyalet Üniversitesi (ISU) Bilgisayar Bilimleri Bölümü'nde doçent olarak görev yapmaktadır. Honavar ayrıca Lawrence E. Baker Bioenformatik ve Biyolojik İstatistik Merkezi, Sanal Gerçeklik Uygulama Merkezi ve ISU'da Biyoenformatik ve Hesaplamalı Biyoloji öğretim üyesidir. Araştırma ve öğretim ilgi alanları arasında Yapay Zeka, Makine Öğrenmesi, Biyoenformatik ve Hesaplamalı Biyoloji, Dilbilgisel Çıkarım, Akıllı Aracılar ve Çoklu Aracı Sistemleri, Dağıtık Akıllı Bilgi Ağları, Saldırı Algılama, Sinirsel ve Evrimsel Hesaplama, Veri Madenciliği, Bilgi Keşfi ve Görselleştirme, Bilgi Tabanlı Sistemler ve Uygulamalı Yapay Zeka bulunmaktadır. Hakemli dergilerde, konferanslarda ve kitaplarda 90'dan fazla araştırma makalesi yayınlamış ve üç kitabın ortak editörlüğünü yapmıştır. Elsevier tarafından yayınlanan Bilişsel Sistemler Araştırma Dergisi'nin ortak editörüdür. Araştırmaları kısmen Ulusal Bilim Vakfı, Ulusal Güvenlik Ajansı, Savunma İleri Araştırma Projeleri Ajansı (DARPA), ABD Enerji Bakanlığı, John Deere Vakfı, Carver Vakfı, Pioneer Hi-Bred Inc. ve IBM'den aldığı hibelerle finanse edilmiştir. Prof. Honavar, ACM, AAAI, IEEE ve New York Bilimler Akademisi üyesidir.

Les Miller, Iowa Eyalet Üniversitesi'nde Bilgisayar Bilimleri profesörü ve bölüm başkanıdır. Araştırma ilgi alanları arasında veritabanları, veri ambarları, heterojen dağıtık veri kaynaklarının entegrasyonu ve mobil aracılar yer almaktadır. 1989-1999 yılları arasında ISMM'nin Uluslararası Mikro Bilgisayar Uygulamaları Dergisi'nin editörlüğünü yapmıştır. Ayrıca Uluslararası Bilgisayarlar ve Uygulamaları Derneği'nin (ISCA) başkan yardımcılığını da yapmıştır.

Yanxin Wang, Bilgisayar Bilimleri Bölümü'nde doktora adayıdır.
Iowa Eyalet Üniversitesi'nde Bilgisayar alanında yüksek lisans derecesi aldı.
1996 yılında Çin'deki Tsinghua Üniversitesi'nden Fen Bilimleri alanında lisans derecesi aldı.
Pekin Havacılık Üniversitesi'nden Bilgisayar Bilimleri alanında lisans derecesi
ve Astronotluk, Çin'de 1993 yılında doktorasını tamamladı. Şu anki araştırma ilgi alanları şunlardır:
ağ ve ağ güvenliği, saldırı tespiti ve işletim sistemi
tems.