BİLGİSAYAR AĞLARINDA SALDIRI TESPİTİ İÇİN İSTATİSTİKSEL YÖNTEM KULLANIMI VE BİR KARMA SALDIRI TESPİT SİSTEMİ TASARIMI

Muhammed Ali AYDIN[1] **Bülent ÖRENCİK[2]**

[1]Bilgisayar Mühendisliği Bölümü
Mühendislik Fakültesi
İstanbul Üniversitesi, 34320, Avcılar, İstanbul
[2] Bilgisayar Mühendisliği Bölümü
Elektrik-Elektronik Fakültesi
İstanbul Teknik Üniversitesi, 80626, Maslak, İstanbul

[1]e-posta: aydinali@istanbul.edu.tr [2] e-posta: orencik@cs.itu.edu.tr

Anahtar sözcükler: Bilgisayar Ağları, Bilgisayar Güvenliği, Saldırı Tespiti, İstatistiksel Yöntem, Karma Sistem

ABSTRACT

Intrusion Detection Systems (IDS) are hardware and software systems that monitor computer networks and systems for violations of security policy. IDS can be designed as signature-based or anomaly-based. Signature-based systems can only detect attacks that are known before whereas anomaly-based systems are able to detect unknown attacks. In this study a hybrid-IDS is developed. Snort, an open-source software, is chosen as the signature-based element of the newly developed system. Statistical method of anomaly detection approach is improved and built into Snort to realize a new hybrid system. The improved statistical method's and the hybrid system's performance are tested on the network traffic data (IDEVAL) that is used in MIT Lincoln Laboratories IDS evaluation in 1999. This test purposes to improve statistical methods and to find how many attacks are additionally detected with newly developed hybrid system compared to signature-based system on its own. In this study; PHAD (Packet Header Anomaly Detection), a statistical method of anomaly based IDS, is experimentally improved and added to Snort. The performance of the resulted hybrid IDS is significantly better than Snort.

1. GİRİŞ

Bilgisayar ağlarına yapılan saldırıları önlemek için kullanılan güvenlik duvarları yerleştirildiği ağlar arasında belirlenen bir politika çerçevesinde yalıtım sağlayan ağ bileşenleridir. Ancak güvenlik duvarının yerel ağ dışından gelen saldırıları önlemesine rağmen, saldırıların çoğunun ağ içerisinden düzenlendiği göz önüne alındığında yetersiz kaldığı açıkça görülmektedir.

Bunların önüne geçmek, en az zararla kurtarmak ya da güvenlik sorunlarını bularak aynı hataların tekrarlanmasını önlemek için Saldırı Tespit Sistemleri (Intrusion Detection Systems) kullanılmaktadır. Saldırı Tespit Sistemleri (STS'ler), bilgisayar sistemlerinde veya bilgisayar ağlarında oluşan olayları otomatik olarak belirleyerek güvenlik sorunları oluşturabilecek durumları analiz eden yazılım veya donanım sistemleridir. "2003 CSI/FBI Computer Crime And Security Survey" incelemesinde 1999 yılı itibariyle STS kullanımı %42 iken, 2003 yılında bu oran %73'lere ulaşmıştır.

STS'ler analiz yaklaşımlarına göre kural-temelli (imza-temelli) ve anormallik-temelli (davranış-temelli) olmak üzere ikiye ayrılmaktadır. Bu çalışmada anormallik-temelli yaklaşımların bir uygulaması olan istatistiksel yöntem incelenerek deneysel olarak geliştirilmiştir. Aynı zamanda her iki sistemin (imza-temelli ve anormallik-temelli) avantajlarını birleştirerek daha iyi bir sistem elde edilmeye (karma STS) çalışılmıştır.

2. İMZA TEMELLİ BİR SALDIRI TESPİT SİSTEMİ : SNORT

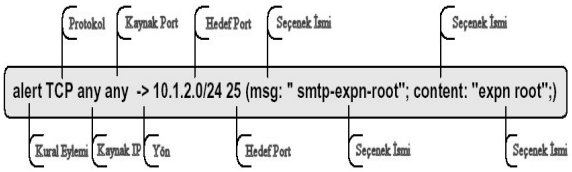
Kötüye kullanım detektörleri sistem etkinliğini analiz eder, olayları veya bilinen bir saldırıyı tanımlayan olayların önceden tanımlanmış modeliyle benzeşen olaylar dizisini arar. Bilinen saldırıları modelleyen şablona imza adı verildiğinden dolayı bu yönteme "imza-temelli tespit" de denir [1, 2, 3].

Snort, IP ağları üzerinde kötüye kullanım tespiti ve gerçek-zamanlı trafik analizi yapabilen yakın zaman önce geliştirilmiş bir ağ-temelli saldırı tespit sistemidir [4, 5].

Kaynak kodu ile birlikte dağıtılan ve kısa sürede son derece popüler olan Snort yazılımının artan işlevselliği ve becerileri nedeniyle pek çok firma Snort temelli ticari STS çözümleri geliştirmekte ve satmaktadır. Snort günümüzde çok sayıda büyük kuruluş tarafından tercih edilen bir STS haline gelmiştir.

Snort, şablon eşleme tekniğine dayanır ve içerik analizi yapar. Önceden tanımlanmış kötüye kullanım kurallarına göre alarm verir. Snort kural tabanlıdır ve kullanılan dil yeni kurallar tanımlamaya elverişlidir. Bu sayede kullanıcılar, varolan kuralları kendilerine göre düzenleyip kendi kurallarını ekleyebilmektedirler. Gerekli tüm özelleştirmeler düz metin dosyaları üzerinde yapılmaktadır.

Snort kural tanımlama dilinde her bir kural iki kısımdan oluşur: *Kural başlığı* ve *Kural seçenekleri*. Kural başlığı beş bölümdür; kural tepkisi (saldırı tespit edildiğinde verilecek tepki), uçlar arasındaki kaynak ve hedef bilgisi (protokole özgü kaynak ve hedef IP adresleri ve port numaraları), trafik akış yönü bilgisi ve protokol türü (TCP, UDP veya ICMP).



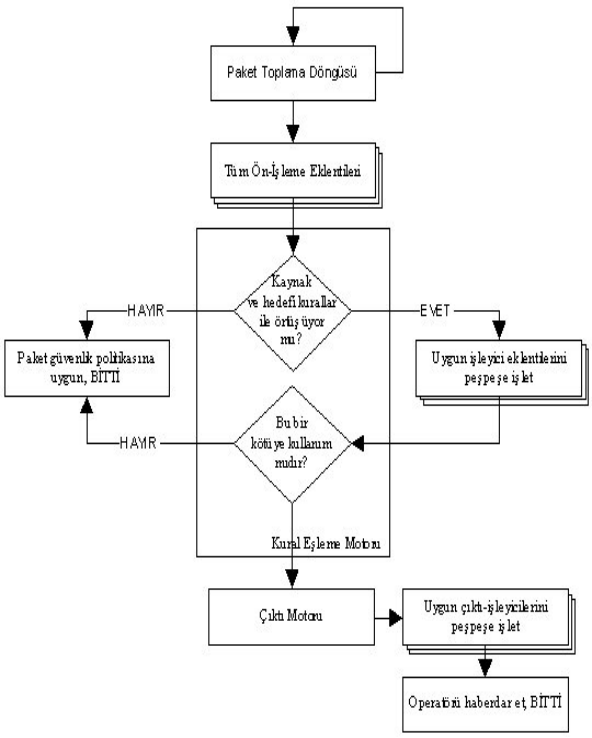
Şekil-1. Snort Kural Yapısı

Kural seçenekleri, belirtilen kötüye kullanım işleminin gerçekleşip gerçekleşmediğine karar vermede kullanılan çeşitli koşullardan oluşur. Örnek bir Snort kuralı Şekil-1'de verilmektedir. Her kuralın ilk alanı eylemdir. Şekil-1'deki kuralda seçilen eylem 'alert'tir. Bunun anlamı kuralda belirtilen kriterle eşleşen bir giriş geldiğinde, bir alarm oluşturulacağıdır. Sonraki alan protokol bilgisini göstermektedir. Örnek kuraldaki protokol TCP'dir. Üçüncü ve dördüncü alanlar kaynak adreslerden oluşur; ilk kısım IP adresi, ikinci kısım kaynak port numarasıdır. Eğer bu alanda "any any" şeklinde değerler bulunuyorsa bu, paketlerin herhangi bir IP adresinden ve herhangi bir TCP portundan gelebileceğini gösterir. Beşinci alan bilgi akış yönünü göstermektedir. Altıncı ve yedinci alanlar hedef adreslerden oluşur. Örnek kuraldaki hedef IP adresi 10.1.2.0/24 olarak verilmiştir ve ilgili bir ağdaki bütün IP adreslerini eşler. Bu örnekte TCP hedef portu 25 olarak ayarlanmıştır. 25 numaralı port, Simple Mail Transfer Protocol (SMTP) için kullanılmaktadır [6]. Hedef adresi takiben parantez içinde seçenekler listesi bulunmaktadır. Her seçenek bir seçenek ismi, varsa seçenek değeri ve seçeneğin bitişini gösteren bir noktalı virgülden oluşur. Şekil-1'de gösterilen kuralda ilk seçenek 'msg'dir ve eylem mesajını belirtmek için kullanılmıştır. İkinci seçenek olan 'content', bir şablon eşleşme kriterini belirtmektedir. Örnekte girişin veri alanı kısmında 'expn root' karakter dizisi aratılmaktadır. TCP veri alanında bu karakter dizisine rastlandığı zaman, koşul gerçeklenmiş olur. Bu kriterlerden birinin bile sağlanmaması halinde alarm üretilmez.

Snort'un veri yakalama motoru, Lawrence Berkeley National Laboratuarında geliştirilen *libpcap* paket yakalama kütüphanesini [7] kullanır. Sistem on taneden fazla UNIX türevini ve MS-Windows'u desteklemektedir. Ayrıca libpcap kütüphanesini kullanmasından ötürü, farklı ağ ortamlarında çalışabilmektedir. Snort'un başlıca beş bileşeni vardır [8]:

- Paket yakalama / ayıklama (dekoder) motoru
- Önişlemci eklenti-yazılımları (plug-ins)
- Tespit motoru
- Kayıt ve Alarm Sistemi
- Çıkış eklenti-yazılımları

Snort'un paketleri işlemesi Şekil-2'de gösterilmektedir [9].



Şekil-2. Snort'un Paketleri İşleme Döngüsü

3. İSTATİSTİKSEL YAKLAŞIMLA ANORMALLİK TEMELLİ STS: PHAD

Bu çalışmada istatistiksel temelli bir saldırı tespit sistemi olan anormallik tespit yaklaşımlarından Paket Başlığı Davranış Tespiti (PHAD)[10] üzerinde durulmuş ve modelin deneysel olarak iyileştirilmesi sağlanarak Snort'a önişlemci olarak eklenmiştir. PHAD diğer ağ temelli davranış tespit sistemlerinden iki yönüyle farklıdır. Bu farklardan ilki PHAD'ın kullanıcı davranışlarından ziyade protokolleri modellemesidir. Çünkü birçok saldırı protokol uygulamalarındaki açıklardan faydalanır ve ancak sıra

dışı girdi ve çıktıların tespit edilmesi ile anlaşılabilir. Diğer fark da PHAD'ın ağ istatistiklerinin kısa süre içerisindeki hızlı değişiminden uyarılan zaman-temelli bir model kullanmasıdır. Bu modellerin temel özellikleri şöyle özetlenebilir :

Protokol Modeli: Davranış tespiti sistemlerinin çoğu, yetkili ve yetkisiz kullanıcıları ayırt etmek için tasarlanmıştır. Örneğin yetkili bir kullanıcı ağ topolojisini bildiği için port taramasının yaptığı gibi var olmayan sunuculara ve hizmetlere bağlanmaya çalışmaz. Ayrıca şifre isteyen sunucular (TelNet, FTP, POP3, ...) kaynak IP adresleri ile tanımlı yetkili istemcilerden ve/veya günün belirli zamanlarında gelen isteklerden normal davranışı anlar. O halde bu hizmetlere erişmeye çalışan farklı kaynak adresleri için yetkisiz erişim uyarısı verilebilir. Bu tür STS'ler kullanıcı modellemeye dayanmaktadır. Diğer bir yaklaşım ise PHAD'ın da benimsediği protokol modellemesidir. Bilindiği gibi birçok saldırı, protokollerin uygulamalarındaki açıklarından yararlanır. Örneğin bu tip saldırılar sendmail, imap ve named protokollerin hatalı uygulamalarını kullanabilirler. Teardrop ve ping of death saldırıları, IP Protokolünün hatalı uygulamalarını deşerler. Bu saldırılar sırasında ağdaki etkinlik bir protokol anormalliğini işaret edebilir. Protokol anormalliklerindeki diğer bir etmen saldıran kodun hatalarından gelir. Aynı sunucu veya istemciyi yazan programcının protokolün tüm ayrıntılarını doğru uygulayamaması gibi saldırgan da her şeyi doğru yapamaz. Saldırganın, TTL, başlık uzunluğu, doğrulama biti, parçalanma göstergesi gibi IP başlık alanlarını doldururken yaptığı çeşitli hatalar veya alışılmamış uygulamaları ağda olağandışılıklara yol açabilir.

Zaman-Temelli Model: Birçok ağ olayı kendine benzerdir ve değişik peryotlarda kendini tekrarlayan bir yapıdadır. Ağ olayları birbirinden bağımsız değildir. Tersine uzun vadede bir bağımlılık vardır. Zaman-temelli modeli anormallik tespitine uygulamak için eğitim ve test aralıklarında "tn/r" ile bir anormallik skoru hesaplanır; burada n (herbir alan için uygun türden paketlerin sayısı) ve r (normal değerlerin sayısı) eğitim aralığı boyunca sayılır ve t en son anormalliğin görüldüğü zamandan bu yana geçen süredir [10].

Bu modelde eğitim aşamasında normal olan değerler bulunarak test sırasında normalden sapmalar belirlenir. Örneğin şu eğitim ve test verileri için: Eğitim Safhası (zaman 0-19):00000000000000001111 Test Safhası (zaman 20-24):01223. Eğitim sırasında izin verilen değerler kümesi kayıt edilir{0,1}; bu kümenin eleman sayısı, r = 2, ve gözlem sayısı, n = 20'dir. Eğer gözlemler 0 ile başlayan birim aralıklarla yapılırsa eğitim sırasında görülen en son değer olan "1", 16 zamanında gerçekleşir ve zaman değeri test aşamasında kullanılmak üzere tutulur. Test safhasındaki 22, 23, ve 24. zamanlardaki "2", "2", ve "3" anormalliktir çünkü bunlar eğitim setinde bulunmamaktadır. Görülen ilk "2"nin anormallik skoru tn/r = (22-16)*20/2 = 60 olarak hesaplanır. İkinci görülen "2"nin anormallik skoru (23-22)*20/2 = 10 olarak hesaplanır. "3"ün anormallik skoru (24-23)*20/2 = 10 olarak hesaplanır. "0" ve "1" in anormallik skorları 0'dır çünkü bunlar eğitim safhasında en az bir defa görülmüştür. Bu örnekteki hesaplar tek bir değer baz alınarak yapılmıştır. Birden fazla anormallik özelliğine sahip bir örnek(paket) için anormallik skoru $\sum$ tn/r dir ve burada toplam, anormal özelliklerin üzerinde hesaplanır. [10]

PHAD'ın Anormallik Tespitinde Kullandığı Özellikler: PHAD, ağ paketlerini tespit etmek için kullanılan bir zaman-temelli protokoldür. Her paket için bir skor hesaplar ve gelen ve giden trafik arasında ayrım yapmaz. Paket başlığındaki ilk 4 bayt alanlarına karşılık gelen 33 özelliği modeller. Bir bayttan küçük olan alanlar (TCP bayrakları gibi) bir bayt içinde birleştirilir. 4 bayttan büyük olan alanlar (6 baytlık Ethernet adresleri gibi) bölünür. Özellikler şunlardır:

- Ethernet Başlığı (bütün paketlerde yer alır)
- IP Başlığı
- TCP Başlığı
- UDP Başlığı
- ICMP Başlığı

PHAD, anormal özelliklerin üzerinde $\sum tn/r$ kullanarak bir anormallik skoru hesaplar.

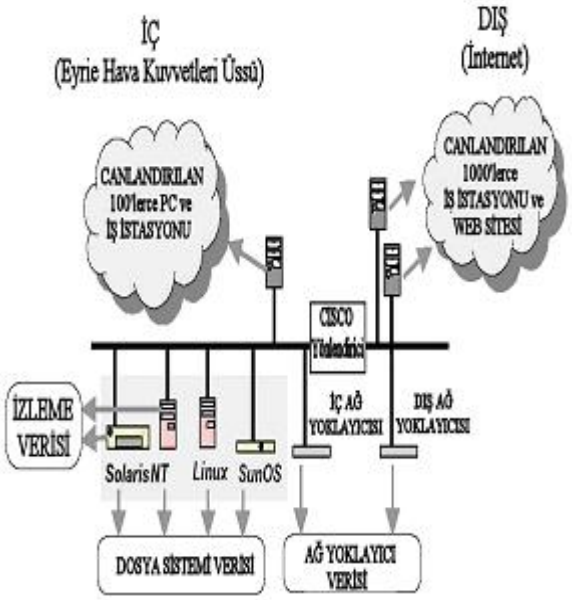
4. IDEVAL DEĞERLENDİRME VERİSİ

Yapılan bir bilimsel çalışmanın, varolan çalışmaları geçip geçmediğinin anlaşılabilmesi için diğerleri ile karşılaştırılması gerekir. Bu karşılaştırma işleminin gerçekleşebilmesi ise sonuçların her değerlendirme sonunda tekrar üretilebilir ve güvenilir olması ile mümkündür. Saldırı tespit araçlarının değerlendirilmesi için kullanılan veriler genellikle kişiye aittir ve bu nedenle değerlendirmenin yeniden yapılması halinde aynı sonuçlar alınamamaktadır. Kişisel verilerin başka ellere geçmesi gizlilik ilkesini çiğnediğinden istenmeyen bir durumdur. Bu sorunların üstesinden gelebilmek için Lincoln Laboratuarı (LL), DARPA'nın sponsorluğunda STS'ler için bir karşılaştırma ortamı sunan IDEVAL veri setini oluşturmuştur [11].

Değerlendirmeler 1998 ve 1999 yıllarında yapılmıştır. 1998 DARPA çalışmasının amacı, saldırı tespit sistemlerinin değerlendirilmesi için ilk standart yapıyı oluşturmaktı. Saldırı tespit sistemlerinin değerlendirilmesi için en uygun durum bilgilerin işleyen bir ağdan alınmasıdır. Ancak bu veriler, kişisel ve gizli bilgiler içerdiği için kullanılamamaktadır. Bu durumdan dolayı Amerikan Hava Kuvvetlerindeki bir yerel ağın simülasyonu gerçeklenmiştir. Bu ağda dns, finger, http, ident, ping, pop, smtp, snmp, telnet, time

ve x servislerini içeren yirmiden fazla ağ servisi otomatik olarak oluşturulmuştur. Ağ üzerinde farklı kategorilerde saldırılar denenmiştir [12].

1999 değerlendirmesi önceki yıla ek olarak skor hesaplama yönteminin basitleştirilmesi, anormallik tespit sistemlerinin eğitilmesi için kullanılabilen saldırı-içermeyen trafik verisi sağlanması, bir çok yeni saldırının dahil edilmesi ve 1998'de yer alan üç UNIX tabanlı hedef sunucuya bir adet Windows NT sunucusunun eklenmesi gibi iyileştirmeler getirmektedir. Kullanılan ağ simülasyonu Şekil-3'de görülmektedir [13].



Şekil-3. IDEVAL Değerlendirme Verisinin 1999 Test Ortamı Blok Diyagramı

Simülasyonu yapılan ağ içerisinde dört ana "kurban" makine bulunmaktadır. Bunların üzerinde SunOS, Solaris, Linux, ve Windows NT koşmaktadır. Trafik oluşturucular yüzlerce sunucuyu ve çeşitli uygulamalar çalıştıran ve İnternet bağlantısına sahip kullanıcıların simülasyonunu yapmaktadırlar. Ağ üzerinden toplanan veri, dört kurban makineden veya yönlendirici ile kurbanlar arasındaki "iç" ağ yoklayıcısı ve yönlendirici ile İnternet arasındaki "dış" ağ yoklayıcısından toplanmıştır. Saldırılar; İnternet'ten, yerel ağ içerisindeki güvenilen sunuculardan veya yerel ağ ve kurbanlara fiziksel erişime sahip saldırganlardan gelecek biçimde tasarlanmıştır. 1999 değerlendirmesi iki aşamadan oluşmaktadır. Bunun ilk aşamasında katılımcılara üç haftalık eğitim verisi dağıtılmıştır. Bu verilerden birinci ve üçüncü haftalar saldırı içermemektedir ve anormallik tespiti yapan sistemleri eğitmek üzere kullanılmaktadır. İkinci aşamada katılımcılara sistemlerini test edebilmeleri için iki haftalık test verisi dağıtılmıştır. Bunlar 58 çeşit saldırının 201 tekrarını içerir.

Saldırılar; Bilgi Tarama(probe), Hizmet Engelleme(DoS), Yönetici Hesabı ile Yerel Oturum Açma(R2L) ve Kullanıcı Hesabının Yönetici Hesabına Yükseltilmesi(U2R) gibi kategorilerine göre, incelenen veri türüne göre, kurbanın işletim sistemine(SunOS, Solaris, Linux, veya NT) göre ve saldırının yeni olup olmadığına göre sınıflandırılmaktadır [14, 15, 16].

5. PHAD İLE İLGİLİ YAPILAN ÇALIŞMALAR

PHAD ile ilgili deneysel çalışmalar IDEVAL Değerlendirme Veri Seti üzerinde test edilmiştir. Ancak bu testlerde orijinaldekinden farklı olarak anormallik skorunun hesabında $\sum t^p n / r$ ifadesi temel alınmıştır (p'nin 1/2, 1, 2, 3, 4 ve 5 değerleri ile deneme yapıldı); oysa orijinal çalışmada p=1 sabit alınmıştır. Burada *n;* her bir alan için uygun türden paketlerin sayısıdır, *r* ise eğitimdeki normal değerlerin sayısı olarak hesaplanır, yani *n* ve *r* değerleri eğitim periyodu sırasınca tespit edilir ve daha sonra test aşamasında $\sum t^p n / r$ bu değerlere ve *t'*ye bağlı olarak paketlerin anormallik skorları hesaplanır. *t* ise en son anormalliğin görüldüğü zaman üzerinden geçen süredir.

Deneysel çalışmalarda hesaplanan anormallik skorları FA'lara göre değerlendirmeye alınmıştır. *FA*; hatalı alarm sayılarıdır.

Tablo-1. p-FA Değişimine Göre Tespit Edilen Saldırı Adedi

p \ FA	10	50	100	200
1/2	6	17	24	24
1	7	26	37	45
2	6	26	40	55
3	5	28	50	64
4	1	23	36	56
5	0	13	15	15

Tablo-1'de görüldüğü gibi PHAD üzerinde yapılan deneysel çalışmalar sonucunda en yüksek STS başarımı, anormallik skorunun $\sum t^3 n / r$ (iyileştirilmiş PHAD) ile hesaplandığında elde edilmektedir.

6. PHAD'IN ÖNİŞLEMCİ OLARAK SNORT'A EKLENMESİ

Anormallik tespiti yaklaşımları olan iyileştirilmiş PHAD'ı Snort'a eklemek için Snort'un önişlemci mimarisinden faydalanıldı. Önişlemciler, paketlerin Snort'un ana tespit motoruna ulaşmadan önce alarm vermesi, atılması ve değiştirilmesi amacıyla bir ortam sağlamak için kullanılan yapıdır.

Projenin gerçekleştirildiği ortam MS Windows 2000 Professional TR kurulu, Pentium IV 2.0 GHz işlemcili, 256 MB RAM'e sahip masaüstü bilgisayarıdır. İlk işlem olarak www.snort.org İnternet sitesinden Snort 2.0.0 sürümünün kaynak kodu indirilerek bu haliyle MS Visual Studio kullanılarak derlendi. Proje, Snort'un kaynak kodu içerisindeki "\src\win32\WIN32-Prj" klasöründe bulunan bir "MS Visual Studio" projesi olan "snort.dsw" kullanılarak geliştirilmiş ve tüm eklentiler ve düzenlemeler bu proje kapsamında gerçekleştirilmiştir.

Snort'a bir önişlemci eklenmesi için izlenmesi gereken standart işlemler mevcuttur. Proje kapsamında PHAD 'ın Snort'a eklenmesi için bu işlemler gerçekleştirilmiştir. İyileştirilmiş PHAD'ın Snort'a önişlemci olarak eklenmesi şu şekilde gerçekleştirilmiştir:

- Önişlemcinin kaynak kodunun yer aldığı dosya olan "spp_phad.cpp", "snort.c" dosyası ile aynı klasöre eklendi.
- PHAD'ın tanımlanması için gereken "spp_phad.h" başlık dosyası, tüm önişlemcilerin çalışma sıralarının belirlendiği "plugbase.h" dosyasına eklendi. #define PP_PHAD 131072. Burada 131072, 2^{17} işleminden hesaplanmış bir değerdir ve 0. kuvvetle bildirilen önişlemci ilk çalıştırılacak olmak üzere PHAD'ın çalıştırılacak 18. önişlemci olduğunu gösterir.
- PHAD'ın başlatılması için gereken "SetupPhad()" fonksiyonu "plugbase.c" dosyasında yer alan "InitPreprocessors()" fonksiyonu içerisinden çağrılmalıdır.
- Son adım olarak proje baştan derlenerek Snort'a PHAD'ın önişlemci olarak eklenmesi gerçekleştirilmiştir.

PHAD'ın anormallik skorları ile tespit ederek alarm ürettiği paketler için değerlendirme EVAL yazılımı tarafından gerçekleştirilir. EVAL, PHAD'ın anormal olarak nitelendirdiği paketleri, IDEVAL veri seti içinde yer alan saldırılar ile karşılaştırarak hangilerinin doğru hangilerinin hatalı (FALSE ALARM- FA) olduğunu ayıran bir yazılımdır. EVAL varsayım olarak 100 FA için çalışır. Bunun anlamı 100. FA'dan sonra PHAD'ın tespit etmiş olduğu paketlerin geri kalan bölümünü değerlendirmeye almayacağıdır. Burada 100 FA değeri, deneysel olarak elde edilmiştir ve ortalama 200FA'nın üzerindeki değerler için tespit edilen saldırı adedinde kayda değer bir artış gözlenmemektedir. 100FA'da 10 günlük test verisi için günde ortalama 10 FA öngörülmüştür.

EVAL, Snort'a çıkış ek-yazılım (output plug-in) mimarisinden faydalanılarak eklenmiştir. EVAL'in kaynak kodunu içeren "spo_eval.cpp" dosyası projedeki "Output Source Files" kısmına eklenmiştir. Önişlemci olarak eklenen PHAD'ın tespit ettiği saldırıların listelenmesi için EVAL kullanılmaktadır. Snort'un bu haliyle yeniden derlenmesiyle Snort'un çıkışı olan "alert.ids"e ek olarak "evaloutp.txt" dosyası elde edilir. "evaloutp.txt" içeriğinde hedef IP adresine veya diğer başlıklara göre değerlendirilmeye alınmamış olan paketler, tespit edilen saldırıların ayrıntılı sınıflandırmasını içeren tablo (Tablo-2), tespit edilen saldırı türlerinin listesi, FA'nın azalan değerlerine göre sıralanmış tespit edilen saldırıların listesi, ilk 100 FA'nın listesi, tespit edilen saldırıların PHAD'ın anormallik olarak gösterdiği hangi paketten bulunduğunu gösteren liste yer alır.

Tablo-2. PHAD($\sum t^3 n / r$)'ın EVAL'den Geçirilmiş Çıktısı

Detections/Total at 100 false alarms (weeks 4-5 only except row W2)

	All	Probe	DOS	R2L	U2R	Data	New	Stealthy
W45	50/201	13/37	27/65	8/56	2/37	0/16	12/62	11/36
IT	49/177	12/34	27/60	8/54	2/27	0/7	11/52	10/30
OT	30/151	9/32	16/44	4/46	1/26	0/11	4/38	6/23
BSM	9/38	1/1	6/12	2/10	0/11	0/6	1/8	0/6
NT	5/33	0/3	4/7	0/10	1/12	0/4	4/26	0/0
FS	49/189	13/37	26/62	8/56	2/31	0/11	12/54	11/34
pascal	17/55	4/8	10/20	3/12	0/11	0/6	2/11	3/9
hume	6/48	0/7	5/15	0/12	1/13	0/5	4/31	0/2
zeno	8/22	4/7	4/9	0/3	0/3	0/1	1/2	3/6
marx	11/44	1/6	5/17	5/18	0/2	0/2	3/11	2/10
Poor	17/72	10/21	4/17	2/15	1/18	0/7	9/38	10/29
W2	0/43	0/9	0/13	0/6	0/12	0/3	0/0	0/0

50 detections, 13748 alarms, 55 true, 101 false, 13592 not evaluated.

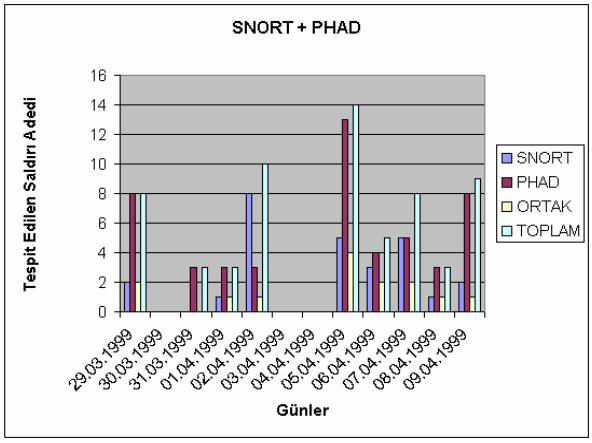
Tablo-2'de W2 - 2.Hafta (Week 2), 1999 değerlendirmesinden önce var olan veri (43 işaretli saldırı içerir). W45 – 4. ve 5. Haftalar (Weeks 4 and 5), değerlendirmede kullanılan veri (201 saldırı içerir). Pascal (Solaris), hume (NT), zeno (SunOS), marx (Linux) – 4 ana hedef makine. Probe, DOS, R2L, U2R, Data – Saldırının dahil olduğu sınıflar, bir saldırı birden fazlasına dahil olabilir. (örneğin R2L-Data veya U2R-Data).

Yalnızca 4. ve 5. hafta için geçerli olan değerler: New - 2.haftada yer bulunmayan saldırı türlerini, Stl – Gizlilik(Stealthy), IT - İç taraftan toplanan trafikte saldırı kanıtı varsa, OT - Dış taraftan toplanan trafikte saldırı kanıtı varsa, BSM - Solaris BSM sistem çağrı izlerinde saldırı kanıtı varsa, NT - NT izleme günlük dosyasında saldırı kanıtı varsa, FS - Dosya sistemi içinde saldırı kanıtı varsa ve Poor – 1999 değerlendirmesinde az sayıda sistemin tespit edebildiği saldırıları göstermektedir. "50 detections, 13748 alarms, 55 true, 101 false, 13592 not evaluated." ifadesi PHAD'ın anormallik olarak işaretlediği 13748 paket içinden 100 FA sınırlamasıyla 55 tanesini saldırı olarak tespit edildiği bunlardan 5 tanesi eşzamanlı tespit olduğundan tam olarak 50 adet saldırı tespit edildiğini göstermektedir. Ayrıca 13592 paket değerlendirmeye alınmamıştır çünkü 100 FA değerine ulaşıldığı anda EVAL, geriye kalan

anormalliklere bakmaz. 200 FA'da ise tespit edilen saldırı adedi 64'dür.

Veri Seti Üzerinde PHAD Eklenmiş Snort'un Başarımı :

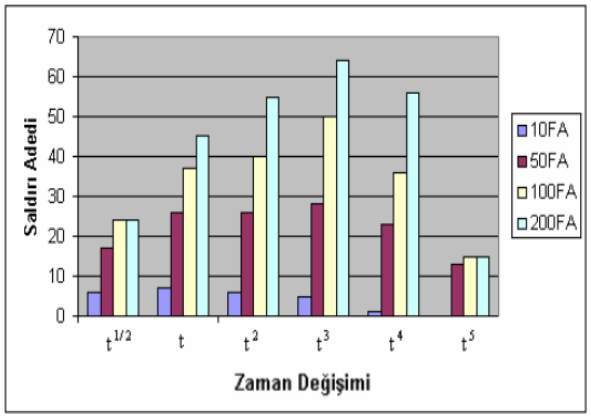
Şekil-4'de görüldüğü gibi önişlemci olarak Snort'a PHAD($\sum t^3 n/r$)'ın eklenmesiyle tek başına Snort tarafından tespit edilen saldırı adedi 27 iken yeni karma sistem(Snort+PHAD($\sum t^3 n/r$)) ile birlikte bu sayı 64'e çıkmıştır.



Şekil-4. Snort+PHAD ile Tespit Edilen Saldırıların Günlere Göre Dağılımı

7. SONUÇ VE YORUM

Bu çalışmada ilk etapta deneysel olarak anormallik tespit sistemlerinden istatistiksel-temelli bir STS PHAD üzerinde çalışılmış ve farklı "p" değerleri için "Anormallik Skoru" $\sum t^p n/r$ hesabı tekrar düzenlenmiştir. Şekil-5'de görüldüğü gibi *t*'nin farklı üsleri için en iyi sonucun Anormallik Skoru $\sum t^3 n/r$ verdiği belirlenmiştir. Şekil-5'e göre t^3'den sonra tespit edilen saldırı adedinde düşüş vardır. Bunun nedeni olarak skoru belirlemede kullanılan *n / r* çarpanının etkisinin diğeri yanında aşırı zayıf kalması gösterilebilir.

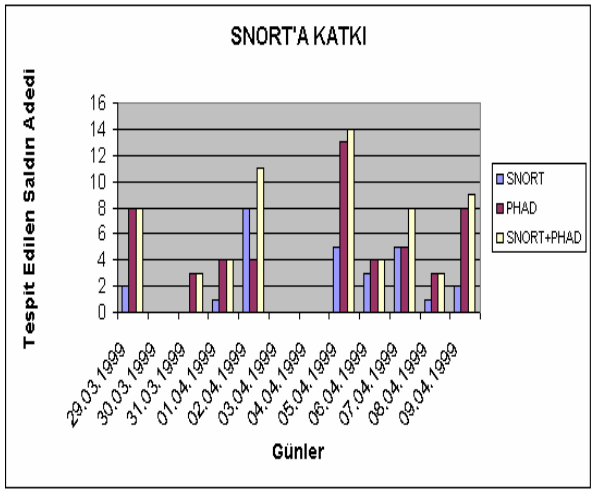


Şekil-5. t-FA Değişimine Göre Tespit Edilen Saldırı Adedi Dağılımı

Anormallik tespiti sistemlerinin, imza-temelli sistemler göz önüne alındığında sağladığı en büyük avantaj henüz tanımlanamamış yeni saldırıların da tespit edilebilmesidir. Ayrıca kural dosyaları içerisinde imzası olmayan saldırıların da yakalanması anormallik tespit sistemleri ile mümkündür. Anormallik tespit sistemlerinden istatistiksel-temelli bir STS olan PHAD ve deneysel olarak iyileştirilen PHAD bu çalışmanın ikinci etabında imza-temelli bir STS olan Snort'a önişlemci olarak eklenmiştir.

Çalışmada STS'lerin değerlendirilmesi için MIT Lincoln Laboratuarlarında hazırlanmış olan IDEVAL veri seti kullanılarak geliştirilen sistemlerin başarımları elde edilmiştir. İlk olarak Snort'un orijinal sürümü olan Snort 2.0.0 çalıştırılarak tespit edilen saldırılar ve saldırı adedi bulunmuştur.

İkinci olarak anormallik tespit sistemlerinden istatistiksel-temelli bir STS olan PHAD($\sum t^3 n/r$) önişlemci olarak Snort'a eklenmiş ve bu haliyle testler tekrarlanmıştır (Snort+PHAD). Burada tespit edilen saldırı adedinde artış gözlenmektedir. İyileştirilen modelin katkısı daha fazla olduğundan Karma Sistem Snort+PHAD($\sum t^3 n/r$) şeklindedir. Şekil-6'da görüldüğü üzere Snort'un tek başına yakaladığı saldırı adedi 27 iken PHAD($\sum t^3 n/r$)'ın önişlemci olarak eklenmesinden sonra bu sayı 64'e çıkmıştır.



Şekil-6. Geliştirilen Karma Sistemin Günlere Göre Tespit Ettiği Saldırılar

Sonuç olarak anormallik tespit sistemlerinden istatistiksel-temelli bir STS olan PHAD'ın iyileştirilerek Snort'a önişlemci olarak eklenmesiyle katkı sağladığı gözlenmiştir. Anormallik tespit sistemleri ile imza-temelli sistemlerin birleştirilmesiyle oluşturulan karma sistemin yalnızca imza tespiti yapan sisteme göre çok daha başarılı olduğu söylenebilir.

KAYNAKLAR

[1] Bace, R., 2000. Intrusion Detection, Macmillan Technical Publishing, Indianapolis USA.

[2] Bace, R. and Mell, P., 2001. Intrusion Detection Systems, NIST Special Publication on Intrusion Detection Systems, SP 800-31, Gaithersburg.

[3] McHugh J., Christie A., and Allen J., 2000. Defending Yourself: The Role of Intrusion Detection Systems, *IEEE Software*, 17(5), 42-51.

[4] Roesch, M., 1999. Snort – Lightweight Intrusion Detection for Networks, *In Proceedings of the 13th LISA Conference of USENIX Association*, Berkeley, CA, USA, 07 – 12 November, pp. 229-238.

[5] Russell, R., 2003. Snort Intrusion Detection 2.0, Syngress Publishing, Inc., Rockland, MA.

[6] Postel, J., 1982. *Simple Mail Transfer Protocol*, Internet Engineering Task Force Request for Comments, STD 10, RFC 821, Information Sciences Institute University of Southern California , California

[7] McCanne, S., Leres C., and Jacobson V., *Packet Capturing Library*, Network Research Group, Lawrence Berkeley National Laboratory, Berkeley CA. ftp://ftp.ee.lbl.gov /libpcap.tar.z

[8] Rehman, R. U., 2003. Intrusion Detection Systems with Snort, Publishing as Prentice Hall PTR, Upper Saddle River, New Jersey

[9] Dayıoğlu, B., 2001. Use Of Passive Network Mapping To Enhance Network Intrusion Detection, *Master Thesis*, The Graduate School Of Natural And Applied Sciences of The Middle East Technical University, Ankara

[10] Mahoney, M.V., Chan, P.K., 2001. *PHAD:* Packet Header Anomaly Detection for Identifying Hostile Network Traffic, *Florida Tech. Technical Report,* CS-2001-04, Melbourne, Florida

[11] Mahoney, M. V. and Chan, P. K., 2003. An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection, *Recent Advances in Intrusion Detection: 6th International Symposium, Raid 2003,* Pittsburgh, Pa, Usa, 8-10 September, pp. 220-239

[12] Kendall, K., 1999. A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems, *Master Thesis*, Massachusetts Institute Of Technology, Lexington, MA.

[13] Lippman, R., Haines, J. W., Fried, D. J., Korba, J., and Das, K., 2000. The 1999 DARPA Off-Line Intrusion Detection Evaluation, *Computer Networks*, 34(4), 579-595.

[14] Lippman, R., Haines, J. W., Fried, D. J., Korba, J., and Das, K., 2000. Analysis and Results of the 1999 DARPA Off-Line Intrusion Detection Evaluation, in *Proceedings of the Third International Workshop on Recent Advances in Intrusion Detection*, Toulouse, France, October 2-4, 162-182.

[15] Haines, J. W., Lippman, R., Fried, D. J., Zissman, M. A., Tran, E. and Boswell, S. B., 2001. 1999 DARPA Intrusion Detection Evaluation: Design and Procedures, *MIT Lincoln Laboratory Technical Report*, TR-1062, Massachusetts, USA.

[16] Data Set, 1999 DARPA Intrusion Detection Evaluation Data Set, http://www.ll.mit.edu/IST/ideval/data/1999/1999_data_index.html