ZEKİ SALDIRI TESPİT SİSTEMLERİNİN İNCELENMESİ, TASARIMI VE GERÇEKLEŞTİRİLMESİ

Esra Nergis GÜVEN

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

EKİM 2007
ANKARA

Esra Nergis GÜVEN tarafından hazırlanan ZEKİ SALDIRI TESPİT SİSTEMLERİNİN İNCELENMESİ, TASARIMI VE GERÇEKLEŞTİRİLMESİ adlı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Doç. Dr. Şeref SAĞIROĞLU …….
Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

Prof. Dr. İrfan KARAGÖZ …….
Elektrik Elektronik Mühendisliği, Gazi Üniversitesi

Doç. Dr. Şeref SAĞIROĞLU …….
Bilgisayar Mühendisliği, Gazi Üniversitesi

Doç. Dr. M. Ali AKCAYOL …….
Bilgisayar Mühendisliği, Gazi Üniversitesi

Tarih: 18/10/2007

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onamıştır.

Prof. Dr. Nermin ERTAN …….

Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Esra Nergis GÜVEN

ZEKİ SALDIRI TESPİT SİSTEMLERİ
İNCELENMESİ, TASARIMI VE GERÇEKLEŞTİRİLMESİ
(Yüksek Lisans Tezi)

Esra Nergis GÜVEN

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
Ekim 2007

ÖZET

Bu çalışmada, bilgi ve bilgisayar güvenliğini sağlamak için geliştirilen araçlardan birisi olan saldırı tespit sistemleri (STS) incelenmiş, STS geliştirmek için kullanılan yöntemler araştırılmış, yapılan STS çalışmaları gözden geçirilmiş ve geliştirilen STS'ler tanıtılmıştır. Bu çalışmanın ana konusu olan zeki STS'ler üzerine detaylı bir inceleme yapılmış ve karşılaştırmalı olarak farklılıklar sunulmuş, elde edilen sonuçlar neticesinde zeki bir STS geliştirilmiştir.

Yapay sinir ağı (YSA) ile geliştirilen STS uygulamaları çalışma kapsamında araştırılmış, STS'lerin uygulanması sırasında kullanılan veritabanları incelenmiş ve KDD'99 veritabanı kullanılarak YSA tabanlı zeki bir STS çalışması gerçekleştirilmiştir. Geliştirilen STS'lerin başarı kriterleri üzerinde durulmuş ve bu başarı kriterlerine göre önceki çalışmalar ile karşılaştırılmıştır.

Yapılan araştırma, inceleme ve değerlendirme çalışmalarından yola çıkarak bu tez kapsamında geliştirilen yazılımlar ve sunulan önerilerin, ülkemizde bilgi ve bilgisayar güvenliği konusunda yapılacak çalışmalara büyük katkılar sağlaması ve yeni ufuklar açması beklenmektedir.

Bilim Kodu : **902.1.014**
Anahtar Kelimeler : **Saldırılar, saldırı tespiti, saldırı tespit sistemleri (STS), zeki STS, saldırı tipleri, saldırı veri kümesi, yapay sinir ağı.**
Sayfa Adedi : **109**
Tez Yöneticisi : **Doç. Dr. Şeref SAĞIROĞLU**

ANALYSIS, DESIGN AND IMPLEMENTATION OF INTELLIGENT INTRUSION DETECTION SYSTEMS
(M.Sc. Thesis)

Esra Nergis GÜVEN

GAZI UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
October 2007

ABSTRACT

In this thesis, intrusion detection systems (IDSs) which are important tools for providing information and computer security were analyzed, the methods used in developing IDSs were reviewed, the studies on IDS's were revised, and an intelligent IDS was developed and introduced in this thesis. A detailed analysis about the intelligent IDSs, was presented and these IDSs were compared to the classical IDSs. In the light of those, an intelligent IDS was developed in this study.

In this study, an intelligent IDS based on artificial neural network (ANN) was introduced. The database used in developing and testing IDS applications was also reviewed. The intelligent IDSs using KDD'99 database were also focused on. The developed IDS was also tested and compared to the available IDSs.

According to the reviews, the evaluations and the experiences gained on IDSs during this study, it can be concluded that this study might contribute to improve the studies on information and computer security being done in near future and bring more awareness and perceptions to the security issues.

Science Code : **902.1.014**
Key Words : **Intrusions, intrusion detection, intrusion detection systems (IDS), intelligent IDS, intrusion types, intrusion data set, artificial neural network.**
Page Number : **109**
Adviser : **Assoc. Prof. Dr. Şeref SAĞIROĞLU**

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren Danışman Hocam Sayın Doç. Dr. Şeref SAĞIROĞLU'na, Yüksek Lisans eğitimim süresince yardımlarını esirgemeyen hocalarım Sayın Doç. Dr. M. Ali AKCAYOL ve Sayın Yrd. Doç. Dr. Hasan Şakir BİLGE'ye, çalışmam sırasında desteklerini esirgemeyen arkadaşlarım Uraz YAVANOĞLU, Rıza YOLAÇAN ve Başak ERÇALIK'a, ayrıca maddi ve manevi her türlü destekleriyle beni hiçbir zaman yalnız bırakmayan çok değerli aileme ve biricik kardeşim A. Emre GÜVEN'e teşekkür ederim.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
AFRL	Hava Kuvvetleri Araştırma Laboratuarı (Air Force Research Projects Agency)
ANN	Yapay Sinir Ağları (Artificial Neural Network)
ART	Adaptif Rezonans Teorisi (Adaptive Resonance Theory)
ASAX	Evrensel Denetim İzi Analizi için Yazılım Mimarisi ve Kural Tabanlı Dil (Software Architecture and Rule-based Language for Universal Audit Trail Analysis)
AT&T	Amerika Telefon ve Telgraf (American Telephony and Telegraph)
BP	Geri Yayılım (Back Propagation)
BSM	Temel Güvenlik Modülü (Basic Security Module)
CERIAS	Bilgi Güvence ve Güvenliği için Eğitim ve Araştırma Merkezi (Center for Education and Research in Information Assurance and Security)
CGI	Ortak Geçit Arayüzü (Common Gateway Interface)
CIDF	Ortak Saldırı Tespit Çatısı (Common Intrusion Detection Framework)
CSM	Birleşik Güvenlik Yöneticileri (Cooperating Security Managers)
CSNET	Bilgisayar Bilim Ağı (Computer Science Network)
DARPA	Savunma İleri Araştırma Projeleri Teşkilatı (Defence Advanced Research Projects Agency)
DBD	Delta Bar Delta
DIDS	Dağıtık Saldırı Tespit Sistemleri (Distributed Intrusion Detection System)
DoS	Hizmet Aksattırma-Engelleme (Denial of Service)

Kısaltmalar	Açıklamalar
DPEM	Dağıtık Program Yürütme Monitörü (Distributed Program Execution Monitor)
DVM	Destek Vektör Makinaları (Support Vector Machines)
EMERALD	Dağılımsal Gerçekleşen Anormallik Tepkilerinin İzlenmesi (Event Monitoring Enabling Responses to Anomalous Live Disturbances)
FIRE	Bulanık Saldırı Tanıma Motoru (Fuzzy Intrusion Recognition Engine)
FTP	Dosya Aktarım Protokolu (File Transfer Protocol)
GD	Güvenlik Duvarı (Firewall)
GrIDS	Grafik Tabanlı Saldırı Tespit Sistemi (Graph-Based Intrusion Detection System)
HTTP	Bağlantılı Metin Aktarım Protokolu (Hypertext Transfer Protocol)
ICMP	İnternet Kontrol Mesajı Protokolü (Internet Control Message Protocol)
IDES	Saldırı Tespiti Uzman Sistemi (Intrusion Detection Expert System)
IDEVAL	Saldırı Tespiti Değerledirmesi (Intrusion Detection Evaluation)
IDIOT	Günümüzde Saldırı Tespiti (Intrusion Detection In Our Time)
IMAP	Internet Mesaj Erişim Protokolü (Internet Message Access Protocol)
IP	İnternet Protokolü (Internet Protocol)
IST	Bilgi Sistemleri Teknolojisi (Information Systems Technology)
KDD	Bilgi Keşfi ve Teslimatı (Knowledge Discovery and Delivery)
LANL	Los Alamos Ulusal Laboratuarı (Los Alamos National Laboratory)
LM	Levenberg-Marquardt
LVQ	Vektör Kuantalama Öğrenme (Learning Vector Quantisation)
MIDAS	Multics Saldırı Tespit ve Alarm Sistemi (Multics Intrusion Detection and Alerting System)

Kısaltmalar	Açıklamalar
MIT	Massachusetts Teknoloji Enstitüsü (Massachusetts Institute of Technology)
MITRE	Massachusetts Teknoloji Araştırma ve Mühendislik Enstitüsü (Massachusetts Institute of Technology Research and Engineering)
MLP	Çok Katmanlı Perseptron (Multilayer Perceptron)
NADIR	Ağ Anormallik Tespiti ve Saldırı Raporlayıcı (Network Anomaly Detection and Intrusion Reporter)
NCSC	Ulusal Bilgisayar Güvenlik Merkezi (National Computer Security Central)
NIDES	Gelecek Nesil Saldırı Tespit Sistemleri (Next -Generation Intrusion Detection Systems)
NIDS	Ağ Saldırı Tespit Sistemi (Network Intrusion Detection System)
NIST	Ulusal Standartlar ve Teknoloji Enstitüsü (The National Institute of Standarts and Technology)
NNID	Sinir Ağı Saldırı Dedektörü (Neural Network Intrusion Detector)
NSM	Ağ Güvenlik İzleyicisi (Network Security Monitor)
NSTAT	Ağ Durum Geçiş Analiz Tekniği (Network State Transition Analysis Technique)
NT	Ağ (Network)
OS	İşletim Sistemi (Operating System)
QP	Hızlı Yayılım (Quick Propagation)
R2L	Uzaktan Yerele (Remote to Local)
RBFN	Radyal Tabanlı Fonksiyon Ağları (Radial Basis Function Networks)
RP	Esnek Yayılım (Resillient Propagation)
RST	İlk Duruma Getirmek (Reset)
SES	Saldırı Engelleme Sistemleri (Intrusion Prevention Systems)
SMB	Sunucu Mesaj Bloğu (Server Message Block)

Kısaltmalar	Açıklamalar
SMTP	Basit Posta Gönderme Protokolü (Simple Mail Transfer Protocol)
SOM	Kendi Kendini Düzenleyen Haritalar (Self-Organizing Maps)
SQL	Yapısal Sorgulama Dili (Structured Query Language)
SRI	Stanford Araştırma Enstitüsü (Stanford Research Institute)
STAT	Durum Geçiş Analizi Tekniği (State Transition Analysis Technique)
STS	Saldırı Tespit Sistemleri (Intrusion Detection Systems)
SYN	Senkronize (Synchronous)
TCP	İletim Kontrol Protokolü (Transmission Control Protocol)
U2L	Kullanıcıdan Yerele (User to Local)
U2R	Kullanıcıdan Yöneticiye (User to Root)
UDP	Kullanıcı Veri Birimi Protokolü (User Datagram Protocol)
USTAT	Unix Sistemleri İçin Durum Geçiş Analizi Tekniği (State Transition Analysis Technique for Unix Systems)
W&S	Bilge ve His (Wisdom and Sense)
YBS	Yapay Bağışıklık Sistemi (Artificial Immune System)
YSA	Yapay Sinir Ağları (Artificial Neural Network)

1. GİRİŞ

İnsanlığın gelişim süreçlerinde en önemli yeri alan "bilgi", bir konu hakkındaki belirsizliği azaltan, sürekli gelişen değişen ve yenilenen bir varlıktır. Ayrıca, saygınlık, ayrıcalık ve farkındalık oluşturduğundan yüksek değere sahiptir. Doğal olarak sahip olunan bu varlığın elektronik ortamlarda karşılaşabileceği tehdit ve tehlikelerden korunması gereklidir. Bilginin değeri, kullanılacağı yere göre değişim gösterir. İhtiyacımız olmayan bir bilgi bizim için önemli değil iken, başkaları için hayati önem taşıyabilir. Bu açıdan bilginin gizliği ve bütünlüğünün sağlanması, yararlı bilgilerin başkalarının veya saldırganların ellerine geçmemesi gereklidir.

Bilgi çağını yaşadığımız şu günlerde, e-devlet, e-imza, e-ticaret gibi kavramlardan oldukça sık bahsedilmektedir. Gerek hız ve verimlilik artışı, gerekse kolaylık sağlaması nedeniyle birçok bilgi elektronik ortamlara aktarılmıştır. Ancak, kişisel veya kurumsal açıdan önemli bir bilginin, başkalarının eline geçmesi ile maddi ve manevi zararlara yol açabileceği görülmüştür. Bu nedenle, elektronik ortamların yaygınlaşarak kullanılmaya başlaması ile birlikte, zaten önemli olan "bilgi ve bilgisayar güvenliği" kavramının önemi günümüzde daha da artmıştır. Birçok kurumsal ve ticari firma bilgi güvenliğine verdikleri önemi öncelikli hale getirmiştir. Geliştirilen e-devlet, e-kurum gibi projelerde güvenliğin en üst düzeyde tutulması ulusal bir amaç haline gelmiş, bu konuda hukuki ve teknolojik önlemler geliştirilmiştir [1]. Ülkemizde de bu konuda hızlı bir gelişim olduğu dikkat çeken konulardan birisidir.

Günümüzde siyasi gücün de bir temsili olan bilişim teknolojilerinin gelişmişliği, bilgi ve bilgisayar güvenliği konusundaki ulusal yapılanmaların önemini ortaya koymaktadır. Artık fiziksel saldırılardan farklı olarak, yakın zamanlarda örneklerini gördüğümüz siber saldırılar devri başlamıştır. Ulusal olarak elektronik ortama verilen önem sadece yeni sistemler geliştirmek ve prosedürel işlere hız kazandırmak olarak algılanırsa, çok büyük maddi kayıplara yol açılabileceği bilinmektedir. Elektronik ortama geçiş sürecinde, standartlarla belirlenmiş olan güvenlik prosedürleri dikkate alınarak, risk ve tehditler belirlenmeli, bilginin değeri ölçülmeli, saldırı senaryolarına

yönelik çalışmalar yapılmalıdır. Bilgiyi korumak için belirlenen kriterlere göre güvenlik politikaları oluşturulmalı ve uygulanmalıdır.

Teknolojik önlemlerin geliştirilmesi sırasında bir varlık olarak bilgiyi, tehdit ve saldırılara karşı korumak için güvenlik duvarları, antivirüs yazılımları, saldırı tespit sistemleri (STS), saldırı engelleme sistemleri (SES) gibi araçlar geliştirilmiştir. Bu araçların belirlenen kural ve politikalara göre yapılandırılması, bilgi güvenliğimizi büyük ölçüde sağlayacaktır.

Günümüzde bilgi ve bilgisayar güvenliğinin öneminin kavranmasıyla, geliştirilen araçlardan biri olan Saldırı Tespit Sistemleri (STS), saldırılara karşı sistemimizde "alarm" niteliği taşıyan yazılım ve donanımlardır. STS'lerin kullanılması ile sistemlere yapılan yetkisiz erişimler ve kötüye kullanımlar tespit edilerek, bunların yol açabileceği zararlar engellenmiş olur. Bilgisayar sistemlerinde STS'lerin kullanılması ile birlikte, sisteme ne tür saldırıların daha çok yapıldığı, sistemdeki mevcut açıklar ve saldırganlar hakkında daha detaylı bilgiler elde edilebilir.

İlk olarak 1980'de Anderson'un yaptığı çalışmalar sonucunda ortaya çıkan STS'ler, ardından yapılan birçok çalışma ile hızla gelişmesini devam ettirmiştir [2]. Denetim verilerinin STS'lerde kullanılmaya başlaması ile birlikte, denetim verilerinin elde edilmesi ve kullanılması ile ilgili çalışmalar artmıştır. 1988 yılında geliştirilen IDES (Saldırı Tespiti Uzman Sistemi), o yıla kadar yapılan birçok çalışmayı üzerinde barındırması açısından en önemli STS çalışmalarından biridir [2]. İstatistiksel yaklaşımların dışında yine o yıllarda, kural tabanlı, eşik değeri belirleme (threshold value), durum geçiş diyagramları (state transition diagrams), veri madenciliği gibi metotların da kullanıldığı bilinmektedir. Ancak teknolojinin hızlı gelişimi ve saldırganların bu gelişimden faydalanarak eskiye oranla daha az bilgi ve tecrübe sahibi olmalarına rağmen daha etkili ve hızlı saldırılar gerçekleştirmeleri, güvenlik boyutunu dinamikleştirmiş ve sürekliliği zorunlu kılmıştır. Bu nedenle, STS'lerin tarihsel gelişimi sürecinde zeki STS'lere ihtiyaç duyularak, yapay sinir ağları (YSA), yapay bağışıklık sistemi, bulanık mantık gibi zeki teknikler de kullanılmaya başlanmıştır. 1990'ların başında kullanılmaya başlanan ve ön plana çıkan zeki tekniklerin kullanımı ile STS'lerin başarı oranlarının arttığı gözlenmiştir. Özellikle,

önceden bilinmeyen yeni saldırıların tespit edilebilmesi için kullanılan anormallik tespiti yaklaşımında, zeki tekniklerin kullanılması başarı oranının arttırılmasında en büyük etkenlerdendir. Bu nedenle, bu tez kapsamında geliştirilen STS, klasik yaklaşımlar yerine zeki tekniklerinden birisi olan YSA'lar kullanılarak gerçekleştirilmiştir.

Günümüze kadar yapılan çalışmalar incelendiğinde, STS'lerde; veri toplama, etiketleme, depolama, veri azaltma (filtreleme, özellik seçme ve sınıflandırma), davranış modellerinin belirlenmesi ve sınıflandırılması, kural tabanlı sistemler için kuralların belirlenmesi, raporlama ve sonuç üretme aşamalarında güçlüklerle karşılaşılmaktadır. Çok sayıda işlemde karşılaşılan bu güçlüklerden de görülebileceği gibi aslında en büyük problem, STS'lerin tasarım ve uygulama aşamalarında kullanılabilecek veritabanlarının eksikliğinden kaynaklanır [3]. Uygulamalarda kullanılabilir bir veritabanının olması, gerçekleştirilmesi planlanan sistemin başarılı olup olmayacağı konusunda daha hızlı sonuç üretilebilmesi açısından önemlidir. Bununla birlikte, STS tasarımı sırasında araştırmacıların farklı veritabanları geliştirmesi sonucunda, başarı oranlarının objektif olarak değerlendirilememesinden dolayı, literatürde saldırı veritabanlarının oluşturulmasına yönelik çalışmalar yapılmıştır. STS'lerin geliştirilmesi ve test edilmesi için kullanılabilecek saldırı veritabanlarının:

- Geliştirilmesinin maliyetli ve zor olması,
- Tamamen ayrı bir çalışma zamanı gerektirmesi,
- Geliştirilse bile gerçeğe uygunluğunun kabul edilebilir olmasının sağlanması,
- Farklı yaklaşım ve tekniklerle kullanılmaya elverişli ve kolay işlenebilir olmasının sağlanması ve
- Güncelliğinin korunması,

gibi problemlerle karşılaşılmaktadır [3].

STS'lerde karşılaşılan problemlerden bir başkası ise yanlış alarm (false positive) oranlarının düşürülememesidir. İmza (kötüye kullanım) tespiti yapan STS'lerde bu durum ile pek karşılaşılmazken, anormallik tespiti yapan sistemlerde, gerek davranış

ya da kullanıcı profillerinin modellenmesi sırasında karşılaşılan güçlükler, gerekse anormallik tespitinin doğası nedeniyle bu durumla çokça karşılaşılır. Aslında bu problemin en aza indirilmesinin yolunun özenle hazırlanmış veritabanlarından geçtiğini de belirtmekte fayda vardır.

Zeki STS çalışmaları incelendiğinde, klasik STS'lerde karşılaşılan problemlerin birçoğuyla zeki STS'lerde de karşılaşıldığı görülmüştür. Bu nedenle, günümüze kadar yapılan zeki STS çalışmalarının pek çoğunda önceden hazırlanmış veri kümelerinin kullanılmış olduğu tespit edilmiştir. Ancak STS'ler için geliştirilen, literatürdeki veritabanı uygulamaları incelendiğinde, günümüzde araştırmacıların uygulamalarında kullanılabilecekleri güncel bir veritabanına rastlanmamıştır. DARPA'nın 1998 ve 1999'da STS'lerin başarılarının değerlendirilmesi için yaptığı çalışma ve DARPA'nın çalışmasının farklı bir versiyonu olan KDD'99 verilerinin oluşturulması, günümüzde hala kullanılan ve saldırı veritabanlarının nasıl hazırlanması gerektiğine örnek olan çalışmalardır. Bu çalışmaların en büyük dezavantajı ise son sekiz yıldır eklenen yeni saldırıları içermemesidir.

Yukarıda sıralanan problemlerin ortadan kaldırılabilmesi için zeki STS'lerin tasarımlarının artması, sistemlere yapılan saldırıların güncel veritabanlarında tutulması ve tasarlanan sistemlerin ise bu güncel verilerle test edilmesi önem arz etmektedir. Bu tez çalışmasında, belirtilen bu problemlerin kısmen ortadan kaldırılabilmesine yönelik olarak zeki bir STS tasarımı gerçekleştirilmiş, literatürdeki mevcut veritabanları kullanılarak test edilmiş ve STS'lerin geliştirilmesi ve başarıların test edilmesinde kullanılabilecek olan "Ulusal Saldırı Veri Kümesi" hazırlanması zorunluluğu ortaya konulmuş ve bunun için farklı önerilerde bulunulmuştur.

Bu tez çalışması 8 bölümden oluşmuştur. Tezin ikinci bölümünde, STS'lerin tarihsel gelişimi ve bilgi ve bilgisayar güvenliği açısından öneminden bahsedilmiş, bilinen saldırı tipleri ve STS'lerin sınıflandırılması üzerinde durulmuştur. STS'lerin geliştirilmesi sırasında kullanılan klasik teknikler açıklanarak, geliştirilen STS'lerin başarılarının sınanması için belirlenen kriterler sıralanmıştır.

Üçüncü bölümde, zeki STS'lerin geliştirilmesi için kullanılan tekniklere yer verilmiştir. Bu tekniklerin STS'lere uygulanması ile ilgili literatürde bulunan örnekler incelenmiştir. Bu tez kapsamında zeki STS tasarımı için kullanılan YSA ile ilgili temel bilgilerin yanında, uygulamalar hakkında daha geniş bir literatür taraması yapılmıştır.

Dördüncü bölümde, günümüze kadar geliştirilen STS yazılımları incelenmiş, bu uygulamaların karakteristik özelliklerine göre sınıflandırılarak karşılaştırılması yapılmıştır.

Beşinci bölümde, STS'lerin geliştirilmesi için en çok kullanılan veri kümeleri değerlendirilmiş, DARPA ve KDD'99 veri kümelerinin oluşturulma aşamaları, içerdikleri veriler, sistem yapıları gibi özellikleri ayrıntılı olarak açıklanmıştır.

Altıncı bölümde, bu tez çalışması kapsamında elde edilen bilgi birikimi ve deneyimlerimiz sonucunda, zeki bir STS ve veri kümesi tasarımı için gerçekleştirilmesi gereken işlem adımları belirlenerek, anormallik ve imza tanıma tabanlı hibrit bir zeki STS önerisi yapılmıştır. Ülkemizde eksikliği görülen saldırı veri kümesinin geliştirilmesi için, "Ulusal Saldırı Veri Kümesi" ile ilgili ön çalışmalar yine bu bölümde gerçekleştirilmiştir.

Yedinci bölümde, STS veri kümelerinden KDD'99 verileri kullanılarak, altıncı bölümde tasarımı yapılan zeki STS modelinin bir modülü olan anormallik tespiti yaklaşımı gerçekleştirilmiştir. Geliştirilen yazılımının işlem adımları, kullanılan eğitim ve test verilerinin içerikleri, kullanılan YSA yapısı gibi temel uygulama bilgileri adım adım açıklanmış ve uygulanmıştır.

Sekizinci ve son bölümde, STS'lerde karşılaşılan problemlerden bahsedilerek, bu tez kapsamında elde edilen sonuçlar ile kazanımlar sunulmuştur.

2. SALDIRI TESPİT SİSTEMLERİ (STS)

Bilgi ve bilgisayar teknolojilerinin hızlı gelişimi sonucu, elektronik ortamların kullanım oranının gün geçtikçe artması ve sağladığı kolaylıkların yanında, bu ortamlarda saklanan bilgilerin güvenliğinin sağlanması bir ihtiyaç haline gelmiştir. Korunacak bilginin değerine göre farklılık gösterebilecek olan koruma sistemlerinin aslında tek amacı, saldırganlara ve saldırılara karşı önlem alarak, bilginin mahremiyetinin korunmasıdır. Bilgisayar sistemlerine yönelik tehditler ve bu sistemlerdeki zayıflıklar olduğu sürece, saldırıları tespit etmek, bilgi güvenliğinde önemli rol oynayacaktır.

Bilgiyi korurken, var olan sistemlerin sürekliliğinin sağlanması da hayati önem taşımaktadır. Sürekliliği korumak için, yapılan saldırılara karşı alınan önlemlerin güncelliğini koruması gerekmektedir. Bu güncellik de ancak değişen saldırı ve yöntemlerin bilinmesi ve var olan sisteme adapte edilmesi ile sağlanabilir. Yeni bir saldırının veya yöntemin tespitine geçmeden önce "saldırı"nın ne anlama geldiğini bilmek gerekir. Bu nedenle güvenlik konusunda yapılan çalışmalarda saldırının birçok tanımı yapılmıştır. Anderson'a göre bir saldırı, izin almadan bilgiye ulaşım, değiştirme, sistemi kullanılmaz veya güvenilmez hale getirmektir [4].

Anderson'un 1980'de yapmış olduğu bu tanım hala geçerliliğini koruyan en temel tanımlardan biridir. Günümüzde ise saldırı, "bilginin mahremiyetini, bütünlüğünü ve erişilebilirliğini tehlikeye atabilecek girişimlerin kümesi" olarak tanımlanmaktadır [5].

Saldırı tespiti ise, bir bilgisayar sisteminde veya ağda meydana gelen olayları izleyerek, bilginin mahremiyetini, bütünlüğünü ve erişilebilirliğini bozmak ya da sistemin güvenlik mekanizmalarını aşmak için yapılan hareketler olarak tanımlanan saldırı işaretlerini analiz etme işlemidir [6].

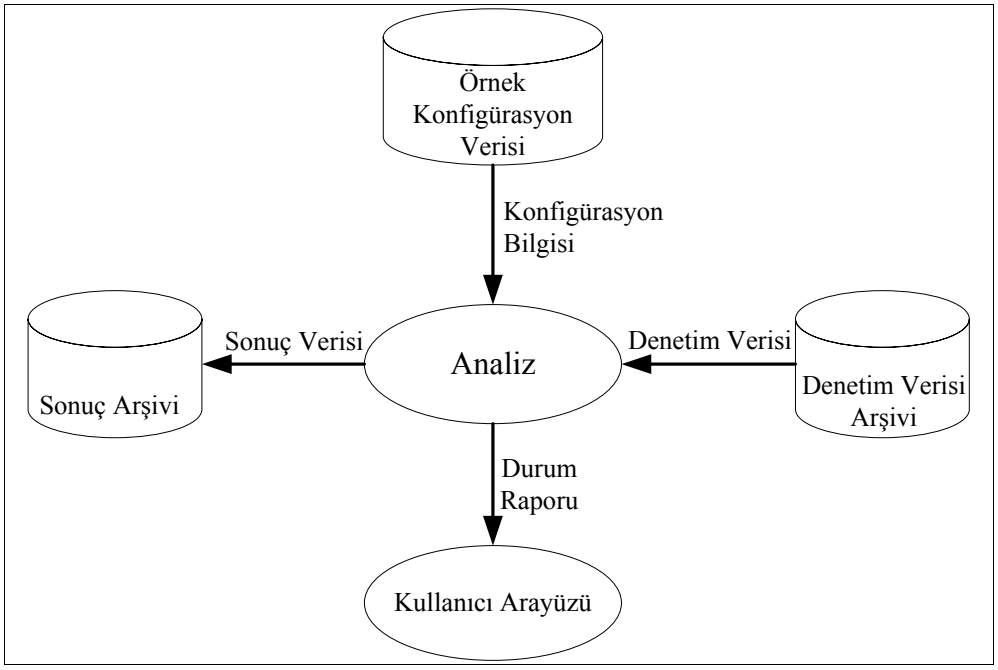
En genel anlamıyla, saldırı tespiti işini yapmak için geliştirilen sistemlere "saldırı tespit sistemleri" denir. Ancak günümüze kadar yapılan araştırmalar ve çalışmalar

incelendiğinde, saldırı tespit sistemlerinin tanımları olduğu görülmüştür. Yapılan bu tanımlara göre STS'ler;

- Bilgisayar sistemlerine yapılan atakları ve kötüye kullanımları belirlemek için tasarlanmış sistemlerdir [7].
- Tercihen gerçek zamanlı olarak, bilgisayar sistemlerinin yetkisiz ve kötüye kullanımı ve suistimalini tespit etmek için kullanılırlar [8].
- Saldırıyı durdurma girişiminde bulunmayan ve olası güvenlik ihlali durumlarında, sistem güvenlik çalışanlarına uyarı mesajı (alarm) veren sistemlerdir [9].
- Bilgisayar sistemlerinin kaynaklarına veya verilerine yetkisiz erişimleri belirler [10].
- Bilgisayar güvenliği alanındaki "hırsız alarm"larıdır [11].
- Bilgisayar veya ağ sistemine yapılan yetkisiz erişimleri tespit etmek için kullanılan yazılım araçlarıdır. STS'ler kötü niyetli ağ trafiği ve bilgisayar kullanımını tespit etme yeteneğine sahiptir. Bir STS, olası güvenlik açıklarını belirleyebilmek için bilgisayar veya ağ içerisinde değişik alanlardan bilgileri toplar ve analiz eder. Güvenlik duvarının statik izleme kabiliyetini tamamlayan dinamik izleme elemanıdır [12].

Yukarıda sunulan tanımlardan yola çıkarak, biz de STS'leri, bilginin elektronik ortamlarda taşınırken, işlenirken veya depolanırken başına gelebilecek tehdit ve tehlikelerin ortadan kaldırılması amacıyla, bilgiye yetkisiz erişim veya kötüye kullanım gibi girişimleri tespit edebilme ve bu tespiti sistem güvenliğinden sorumlu kişilere iletebilme özelliğine sahip yazılımsal ve/veya donanımsal güvenlik araçları olarak tanımlayabiliriz. Aynı zamanda STS'ler, ağ cihazlarını izleyerek anormal davranışları ve kötüye kullanımı tespit ederler.

Literatürdeki STS çalışmaları incelendiğinde, STS'lerin yapısı ile ilgili birçok gösterim yapıldığı görülmüştür. STS çalışmalarından biri olan NIDES'in geliştirilmesi sırasında çizilen örnek Şekil 2.1'de gösterilmiştir [13]. Kullanılan teknik ve yaklaşımlara göre şekillenen diğer gösterim şekillerinden ilerleyen bölümlerde bahsedilecektir.



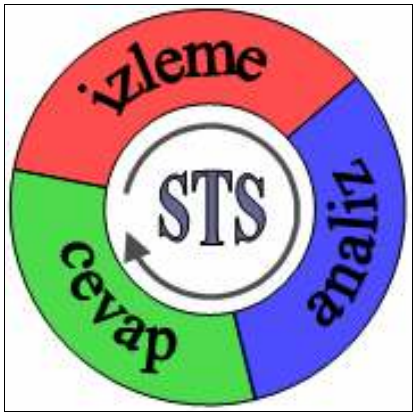
Şekil 2.1. STS'lerin temel yapısı [13]

Şekil 2.1'de sunucu tabanlı bir STS'nin temel yapısına bir örnek gösterilmiştir. Bu örnekte, bilgi kaynağı olarak denetim verileri kullanılmıştır. Denetim verisi arşivinden alınan denetim verileri, örnek konfigürasyon verileri ile karşılaştırılıp analiz edilerek, elde edilen sonuçlar, sonuç arşivine aktarılmaktadır. Aynı zamanda, olası saldırıların, kullanıcı tarafından görülebilmesi için kullanıcı arayüzüne de analiz sonuçlarını içeren durum raporu gönderilmektedir. İstenildiği takdirde, sonuç arşivi incelenebilir veya örnek konfigürasyon verileri güncellenebilir [13].

STS'lerin yaptıkları iş temelde izleme, analiz ve cevap olmak üzere üç adımda özetlenebilir. STS'lerin yapısal bileşenleri Şekil 2.2'de sembolik olarak gösterilmiştir. İzlemeyle başlayan saldırı tespit sürecini, analiz ve ardından sistem cevabı izlemektedir.

STS'ler, korunan sisteme ve kullanılan tekniklere bağlı olarak Şekil 2.2'de gösterilen üç işlevi gerçekleştirmelidir. Birbirini takip eden ve sürekli tekrarlanan bu işlemler ilk olarak izleme ile başlar. Korunacak sistem eğer bir bilgisayar ağı ise, bu bilgisayar ağında tanımlı kullanıcıların hareketleri ve ağdaki paketler, eğer bir sunucu ise bu sunucuya gelen giden veri paketleri izlenebilir. Sistemin izlenmesiyle

edinilen bilgiler gerekli önişlemlerden geçirildikten sonra, kullanılan yaklaşımlara, tekniklere ve kurallara göre analiz edilir. Analiz sonucunda, izlenen hareketin veya paketin saldırı olup olmadığı tespit edilir ve sistem yöneticisine veya sorumlusuna bir cevap dönülür. STS'lerde cevap verme işlemi, genellikle bir saldırı ile karşılaşıldığında sistem yöneticisine alarm veya bilgi verme niteliğinde olabilmektedir.



Şekil 2.2. Bir STS'nin fonksiyonel olarak gösterilmesi

2.1. STS'lerin Tarihçesi

1970'lerden itibaren güvenlikle ilgili araştırmalar yapılmasına rağmen, saldırı tespiti kavramı ilk olarak Anderson'un "Bilgisayar Güvenliği Tehdit Gözetleme ve İzleme (Computer Security Threat Monitoring and Survaillance)" makalesi ile 1980'de ortaya atılmıştır [2]. Bu çalışma, STS'lerin tanımlanması ve tanınması açısından büyük bir öneme sahiptir.

İlk nesil STS'ler, basit bilgisayar sistemleri üzerine düşünülmüştür. İkinci neslinde ise günümüzde STS'lerin vazgeçilmezi olan "denetleme izi (audit trail)" kavramı ve veritabanı mantığının bilgi güvenliği alanındaki önemi ortaya çıkmıştır. Bunu izleyen çalışmalarda, güvenlik konusundaki çalışmalara yardımcı olmak amacıyla günlük denetleme verilerinin otomatik araçlar ile elde edilmesi konusunda çalışmalar yapılmıştır [6]. 1985'ten itibaren bu akımla geliştirilen projelerde denetleme verileri üzerinde özenle durulmuş ve istatistiksel yaklaşımlar temel alınarak saldırı tespit modelleri geliştirilmiştir. Bu çalışmalardan biri olan IDES (intrusion detection expert

system), Denning tarafından geliştirilmeye başlanan ve 1988-1992 yılları arasında yapılan çalışmaların birçoğunu üzerinde barındıran bir sistemdir [13]. Daha sonra ismi NIDES (next-generation intrusion detection systems) olarak değişen bu sistem, ikinci nesil saldırı tespit sistemlerinin en eski ve en bilinen çalışmalarından biridir. IDES, saldırı senaryolarının kural kümelerinin çıkarılmasıyla dizayn edilmeye başlanan kural tabanlı bir STS'dir. Denning çalışmasında 3 farklı istatistiksel model tanımlamıştır. Bu modeller;

- kullanıcının belirli aralıklarla bir işlemi tekrar etmesine izin veren ve eşik değerine göre anormallik olduğunu tespit eden model,
- istatistiksel momentlerin bilindiği varsayılarak tespit edilen sapmalar ile anormallik olduğunu tespit eden model ve
- anormalliklerin tek olaya değil bir diziye bağlı olduğu Markov modeli

olarak verilmektedir [7].

STS'lerin geliştirilmesinde günümüze kadar istatistiksel yöntemlerin dışında, kural tabanlı (rule based), eşik değeri belirleme (threshold value), durum geçiş diyagramları (state transition diagrams), yapay sinir ağları (artificial neural networks), veri madenciliği (data mining), yapay bağışıklık sistemi (artificial immune system), bulanık mantık (fuzzy logic) gibi farklı birçok yaklaşım uygulanmıştır [14, 15]. Bu yaklaşımlara "STS'lerde kullanılan teknikler" başlığında ayrıntılı olarak Bölüm 2.4.'de ve "zeki STS'lerde kullanılan teknikler" başlığında Bölüm 3.2'de yer verilecektir.

STS'ler, bilgisayarlar ve veri ağları için yeni bir güvenlik yaklaşımı sunmaktadırlar. Daha önce de belirtildiği gibi STS'lerin amacı, saldırı, yetkisiz kullanım, suistimal, sistem içi ve dışından davetsiz misafirlerin sisteme zarar vermesi gibi durumların teşhis edilmesidir. Saldırı tespit problemi, bilgisayar ağlarının hızlı şekilde artması ve bu artışın sonucu olarak sisteme erişiminin çoğalması ile davetsiz misafirlerin kimliklendirmeyi kolaylıkla reddetmesinden sonra büyük önem kazanmıştır [8].

STS yazılımlarının test edilmesinde kullanılan "saldırı veritabanları" örneklerine literatürde rastlanmaktadır [16-18]. STS'nin temelini oluşturan bu veri kümelerini oluşturmak oldukça zor bir iştir. Verileri oluşturmanın zorluğu dışında, oluşturulan verilerin gerçek verilere uygunluk göstermesi gerekir. Normal davranış simule edilse bile bu verileri saklamak masraflıdır. Günümüzde de bazı uygulamalarda hala kullanılan MIT'nin Lincoln Laboratuarlarında 1998 yılında oluşturulan DARPA Saldırı Tespit Değerlendirme veri kümesi [16] bilinen en kapsamlı olanıdır. Bu veri kümesinin amacı, saldırı tespit sistemlerinin değerlendirilmesi ile ilgili standart bir ortam oluşturulmasına katkı sağlamaktır. DARPA verilerinin tcpdump ile alınan paket başlıklarını içermesi ve bu konuda yapılacak çalışmalar için oldukça fazla ön işlemden geçirilmesi gerekliliğinden dolayı geliştirilen KDD'99 veri kümesi, DARPA verilerinin ön işlemden geçirilmesi ile elde edilmiştir [17]. Kullanımı kolay olmasıyla birlikte yapılan ön işlemlerin çok fazla olması nedeniyle gerçek ortamda bunun gerçekleştirilmesi oldukça zordur. Ayrıca DARPA veri kümelerinin güncellenmemiş olması nedeniyle her iki veritabanı da 2000 yılından sonra yapılan farklı saldırı türlerini içermemektedir.

Saldırı veritabanlarının güncellenmemesi, ticari veritabanlarının ise başkalarıyla paylaşılmaması, ülkemizde bu konuda çalışmak ve STS geliştirmek isteyenlerin önünde bir engel olarak görülmektedir. Bu konuyla ilgili olarak ülkemizde herhangi bir çalışma yapılmaması ve ulusal çalışmalarda kullanılabilecek bir saldırı veritabanının eksikliği görülmüştür. Bu tez kapsamında, bu eksikliğin giderilebilmesi için ulusal veri kümesi konusunda girişimlerde bulunulmuş ve böyle bir veri kümesinin oluşturulması için gerekli alt yapı çalışmalarına başlanmıştır. Büyük destek alan bu çalışmanın kısa sürede tamamlanması hedeflenmektedir.

2.2. Bilgi Güvenliğinde STS'lerin Önemi

İnternetin ve iletişim olanaklarının artmasıyla birlikte saldırganlar tarafından saldırılabilecek daha çok sistem ortaya çıkmıştır. Bu saldırıların büyük bir bölümü kullanılan sistemin kusurları veya eksiklerinden faydalanılarak yapılır. Bu tür saldırıları engellemenin iki yolu vardır; ilki tamamen güvenli bir sistem ve ortam

oluşturmak, ikincisi ise saldırıları tespit edip gerekli önlemleri almaktır. Bunlardan ilki pratik açıdan mümkün olmamaktadır. Bunların gerekçeleri ise [19],

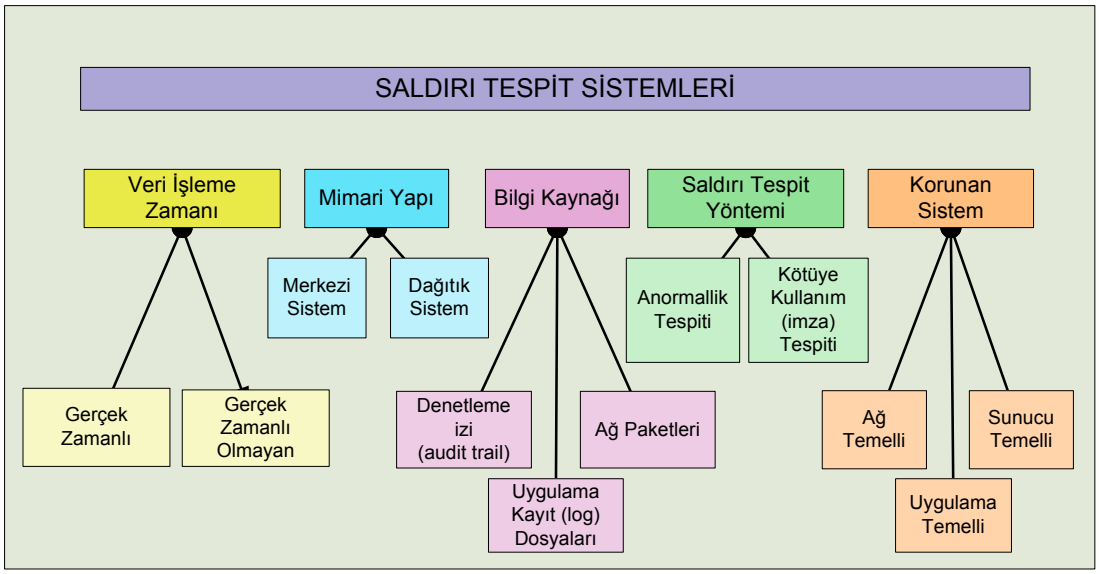
- Kullanılan işletim sisteminde var olan açıkların genellikle ilk olarak saldırganlar tarafından fark edilmesi ve önlem alınana kadar bu açıkların kullanılabilmesi,
- Veri iletiminde kullanılan protokollerin yapısında var olan bazı kuralların saldırı amaçlı kullanılabilmesi,
- Kriptografik metotların ve anahtarlarının kırılabilmesi, kullanıcıların şifrelerini unutabilmesi veya kripto-sistemin kırılabilmesi gibi nedenlerle yüksek seviyede bir güvenlik sağlanamaması,
- Dış ortama karşı güvenliği sağlanan sistemin, iç ortamlardan suistimal edilerek güvenliğin ortadan kaldırılması,
- Güvenlik amacıyla kullanıcı yetkilerinin minimuma indirilmesi sonucu kullanıcı verimliliğinin düşmesi gibi nedenleri vardır.

Sistemlerini korumak isteyenler, genelde saldırı gelene kadar bekleme pozisyonunda kalmak, saldırı geldiğinde ise olabildiğince hızlı tespit etmek isterler. Bu ise STS'nin yaptığı iştir [19]. Bir saldırının hangi adresten veya hangi porttan geldiğini bilmeden engel olmak mümkün değildir. STS'ler saldırıları tespit ederken bu bilgileri de elde ederler. STS'ler, detaylı olarak topladığı ve depoladığı bilgilerden yararlanarak, saldırıları olabildiğince erken tespit etme özelliğine sahiptir. Yine aynı bilgilerin incelenmesi ile daha önce hiç karşılaşılmamış bir saldırıyı da tespit edebilir. STS'leri de cazip hale getiren bu özelliktir.

Bu nedenlerden dolayı, tamamen güvenli bir sistem oluşturmanın mümkün olmadığı görülse de, üst düzey güvenliğe sahip bir sistem oluşturmak mümkündür. Bunun için maliyet de göz önünde bulundurularak gerekli olan tüm güvenlik araçlarından faydalanılmalıdır. Ancak saldırganların, kapatılan açıkların ardından, saldıracak yeni açıklar bulabilmesinin ve önceden tahmin edilemeyen yollara başvurmasının, tamamen güvenli bir ortamın oluşturulamamasına neden olduğu unutulmamalıdır. STS'lerin bu konudaki önemi, kullanılan yeni teknikler sayesinde önceden bilinmeyen saldırıların da tespit edilebilmesini sağlamasıdır.

2.3. STS'lerin Sınıflandırılması

STS'ler günümüze kadar farklı birçok kritere göre sınıflandırılmıştır. Bunlardan en çok bilinen sınıflandırma türü saldırı tespit yöntemine göre olup, anormallik tespiti ve kötüye kullanım tespiti olarak ikiye ayrılır [13]. Ancak STS'lerin mimari yapısı, korunan sistemin türü, verinin işlenme zamanı gibi farklı sınıflandırmalar da yapılabilir [11]. STS'lerin en çok kullanılan kriterlere göre sınıflandırılması Şekil 2.3'de verilmiştir.



Şekil 2.3. Saldırı tespit sistemlerinin sınıflandırılması

Bir STS, Şekil 2.3'de gösterilen sınıflandırma kriterlerinden her biri ile farklı sınıflarda yer alır [11]. Bu, sınıflandırmanın hangi kritere göre yapıldığına bağlıdır.

Örneğin; 1988'de geliştirilen IDES, veri işleme zamanına göre gerçek zamanlı, mimari yapısı bakımından merkezi sistemli, kullanılan bilgi kaynağı olarak sunucu tabanlı, saldırı tespit yöntemi olarak anormallik tespiti yaklaşımı, koruduğu sisteme göre sunucu temelli sınıfına dahildir [20]. Belirlenen sınıflandırma kriteri sadece saldırı tespit yaklaşımı ise IDES anormallik tespiti yaklaşımını kullanan sınıfının bir üyesidir. Bu durumda diğer kriterlerden bahsedilmez. Ancak tüm bu kriterler, aynı zamanda bir STS'nin karakteristiğini ortaya koyması açısından önemlidir.

Literatürde STS'ler birçok farklı şekilde sınıflandırılsa da bu tez kapsamında, STS'lerin sınıflandırılması ve tanımlanması için, veri işleme zamanı, mimari yapı, bilgi kaynağı, saldırı tespit yöntemi ve korunan sistem temel alınan özelliklerdir. Şekil 2.3'de gösterilen bu özellikler, takip eden alt başlıklarda açıklanmıştır.

2.3.1. Veri işleme zamanı

STS'ler için veri işleme zamanı, izlenen olaylar ve olayların analizi arasında geçen zamanı ifade eder. STS'ler "gerçek zamanlı" ve "gerçek zamanlı olmayan" şeklinde ikiye ayrılırlar [11].

Gerçek zamanlı sistemlerde; veri, iletişim anında analiz edilir ve eğer bir saldırı tespit edilirse saldırıya cevap olarak gereken çevre değişimleri gerçekleştirilir. Bu tip STS'ler, yoğun bilgi akışı olan ağlarda uygulanması zor ve maliyetli bir yöntem olmakla birlikte aktif olarak cevap üretilmesi gereken durumlarda da tek çözüm olan sistemlerdir. Ticari uygulamaların gerçek zamanlı sistemler olduğunu da hatırlatmakta fayda vardır.

Gerçek zamanlı olmayan sistemlerde; alınan veriler depolanarak, analiz edilmesi için ilgili STS'ye gönderilir. "Depola ve yolla" mantığıyla çalışan bu sistemlerde belirlenen tehditler ve olası saldırılar, sistem kullanıcısının dikkatini çekecek şekilde açılır pencerelerde alarm olarak gösterilir. Saldırının profilinin çıkarılması ve bu tür saldırılara bir daha maruz kalınmaması için, geçmişe yönelik depolanan veriler üzerinde yapılan analizlerle sonuçlar veya saldırılar elde edilir. STS'nin gerçek zamanlı olması şartı aranmadığı durumlarda kullanılan bu yöntem, sistemin belirlenen güvenlik açıklarını kapatmak için kullanılır.

2.3.2. Mimari yapı

STS'lerin mimari yapısı, fonksiyonel bileşenlerin birbirlerine göre nasıl yerleştirildiklerini anlatır [21]. Temel fonksiyonel bileşenler; izlenen sistem, analizin yapıldığı sunucu ile çevresi ve problemler için izlenen hedef olarak sıralanabilir.

STS'lerin mimari yapısı, merkezi sistem ve dağıtık sistem olmak üzere ikiye ayrılır [20]. Merkezi sistem kontrol stratejilerinde; tüm izleme, tespit ve raporlama işlemleri merkezde kontrol edilir. Bilinen STS'lerin çoğu bu kategoriye girer. Merkezi kontrol için fiziksel yakınlık önemli bir etken değildir. Dağıtık sistemlerde, izleme, tespit ve cevap işlemleri, analiz noktasında yapılan ajan (agent) tabanlı yaklaşım ile yapılır. Genellikle çok geniş olan ağlarda kullanılan bir yapıdır. Örneğin, bir kurumun farklı şehirlerde yer alan bilgisayar sistemlerinin tümünü kapsamına alan STS'ler bu şekildedir.

2.3.3. Bilgi kaynağı

STS'ler, bilgi kaynaklarını analiz ve karşılaştırma yapmak için kullanırlar. Bilgi kaynakları, bilgisayar veya ağ paketlerinin dinlenmesinden elde edilebildiği gibi, kullanıcı profillerinin davranış modellerinden de elde edilebilir. Bilginin nasıl ve nereden toplanacağı, geliştirilecek olan STS'nin amaçlarına göre değişir. Bunlar; denetleme izi, ağ paketleri, uygulama kayıt dosyalarıdır. Bunlar aşağıda kısaca açıklanmıştır.

Denetim (hesap, günlük) izi; bilgi ve iletişim güvenliği için, sistem aktivitelerinin kronolojik olarak sıralanmış şeklidir. Denetleme izleri, sistemde gözlenen olayların sıralaması bozulduğunda veya olaylarda değişiklikler meydana geldiğinde, yeniden yapılandırma ve test etmeyi mümkün kılmak için kullanılır. Kullanıcı tanımlama sistemleri ve veritabanı yönetim sistemlerinin çoğu denetleme izi bileşeni içerir. STS'ler bilgi kaynağı olarak denetleme izini, daha çok sistemde tanımlanmış olan kullanıcıların veya grupların hareket profillerini çıkarmak için kullanır. Kullanıcıların günlük yaptıkları işler ve bu işlere yönelik sistemdeki yetkileri göz önünde bulundurularak, bir kullanıcının gün içindeki hareketleri, o kullanıcının profilini oluşturur. Uzun süren bir gözlem ve inceleme ile bu profilin doğruluğunun kanıtlanması gerekir. Eğer kullanıcının profili belirlenmişse, STS'ler belirlenen profil dışına çıkılan hareketleri saldırı olarak algılarlar. Bu nedenle, yanlış alarm oranlarını azaltmak için kullanıcı profillerinin güncellenebilir olması gerekir.

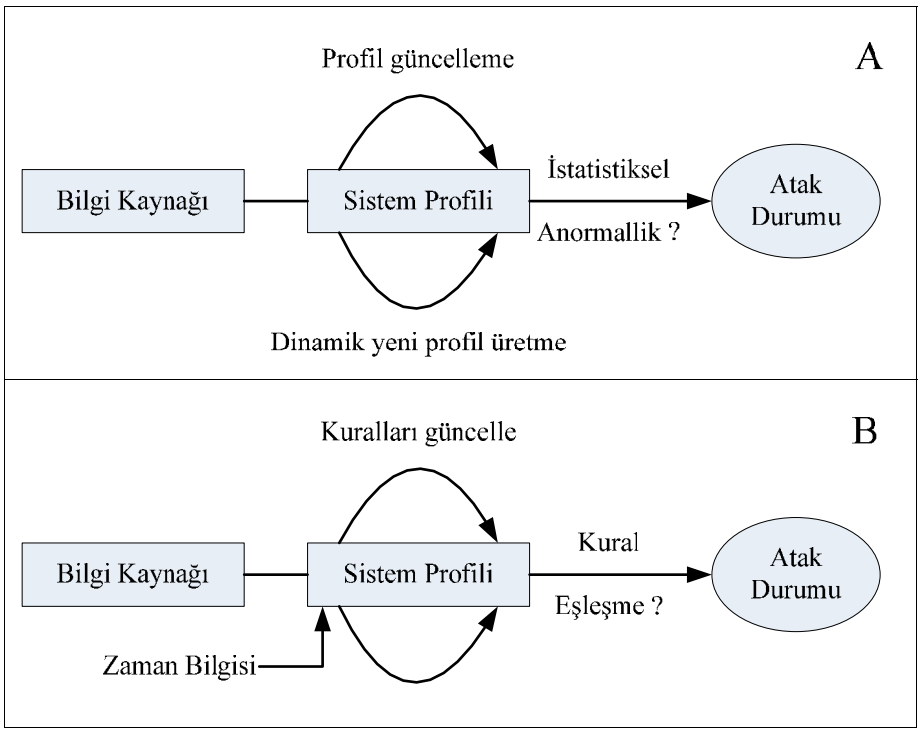
Ağ paketleri; koklayıcılar (sniffers) tarafından trafiğin dinlenmesiyle elde edilir. Bu günlükler daha çok hizmet aksattırma (engelleme) saldırılarını tespit etmekte kullanılır. Ağ paketlerinden elde edilen bilgiler sayesinde, sunucu tabanlı STS'lerden farklı olarak, ağ katmanında gerçekleşen atak olaylarını tespit etmek de mümkündür. Ancak bunun için atak tespit sisteminin ağ yapılandırması sırasında en uygun yere konumlandırılmış olması gerekir.

Uygulama kayıt dosyaları; uygulama katmanında gerçekleşen atakları tespit etmekte kullanılırlar. Bu veri kaynağı, diğer iki kaynaktan daha kolay elde edilmesiyle birlikte sağladığı atak tespit oranı sınırlıdır. Bilgi kaynağı olarak uygulama kayıt dosyalarını kullanan bir STS, web tabanlı saldırıları tespit etmek için geliştirilen bir yazılım olan WebWatcher'ın uygulama log dosyalarını kullanabilir.

2.3.4. Saldırı tespit yöntemi

STS'lerde, saldırı tespit yöntemi olarak anormallik tespiti ve kötüye kullanım tespiti olmak üzere iki farklı yaklaşım kullanılır [13]. Anormallik tespitine dayanan yaklaşım, sistemdeki kullanıcı davranışlarını modellerken, kötüye kullanım (imza) tespitine dayanan yaklaşım, saldırganların davranışlarını modeller. Şekil 2.4'de bu modeller gösterilmiştir.

Anormallik tespiti (anomaly detection) yaklaşımı; temel olarak, sistemde meydana gelen anormal olayları, normal olaylardan ayırt etme yaklaşımıdır. STS için anormallik, normal davranıştan sapma anlamına gelir. Normal davranış, sistemin uzun bir süre analiz edilmesi ile elde edilir. Sistemdeki kullanıcı veya kullanıcı gruplarının davranış profillerinin belirlenmesi anormallik tespiti için temel iştir. Normal davranış profili belirlendikten sonra, farklılık gösteren davranışlar saldırı olarak tespit edilir. Anormallik tespitinde saldırıların doğru tespit edilmesi, normal davranış profilinin ne kadar doğru belirlendiğiyle orantılıdır.



Şekil 2.4. Saldırı tespit yöntemleri (A: Anormallik tespiti yaklaşımının temel yapısı, B: Kötüye kullanım tespiti yaklaşımının temel yapısı) [19]

Anormallik tespiti yaklaşımının kötüye kullanım tespiti yaklaşımına göre avantajı, önceden bilinmeyen saldırıların da tespit edilebilmesidir. Dezavantajı ise yanlış alarm (false positive) yani gerçekte saldırı olmayan davranışların da saldırı olarak belirlenmesi oranının yüksek olmasıdır.

Anormallik tespitinde istatistiksel yöntemler, yapay zeka teknikleri, yapay sinir ağları, veri madenciliği, yapay bağışıklık sistemi gibi birçok farklı teknik kullanılabilir. Bu teknikler ve yaklaşımlardan, klasik olanları Bölüm 2.4'te, yapay zeka teknikleri ise Bölüm 3.2'de açıklanacaktır.

Kötüye kullanım tespiti (misuse detection) yaklaşımında; saldırganların davranış modellerinin çıkarılması esastır. Saldırının imzası olarak da nitelendirilen bu modeller, daha önce karşılaşılmış saldırıların analiz edilerek kendine has karakteristik özelliklerinin çıkarılması ile elde edilir. Saldırılara yönelik imza veritabanları oluşturulduktan sonra, bu imzalarla eşleşen hareketler saldırı olarak tespit edilir.

Kötüye kullanım tespiti yaklaşımının anormallik tespitine göre avantajı, imzası bilinen her saldırının tespit edilebilmesi ve yanlış pozitif üretmemesidir. Dezavantajı ise, imzası bilinmeyen saldırıların tespit edilememesi ve bu nedenle yanlış negatif (saldırı olan bir davranışın saldırı olarak tespit edilmemesi) oranının yüksek olabilmesidir. Yanlış negatif oranının azaltılması için imza veritabanının yeni saldırı imzaları ile güncellenebiliyor olması gerekir.

2.3.5. Korunan sistem

STS'ler korudukları sisteme göre üç gruba ayrılırlar. Korumak istenen sisteme göre; ağ, sunucu ya da uygulama temelli STS'ler olarak adlandırılırlar [20].

Ağ temelli STS'ler; ağdaki trafiği dinleyerek ağ sistemine yapılan saldırıları tespit etmeye yönelik çalışır. Ağ paketlerini yakalayıp bunları analiz ederek saldırı tespiti yaparlar. Ticari saldırı tespit sistemleri genellikle ağ tabanlıdır.

Sunucu temelli STS'ler; bir bilgisayar sistemi içerisinde işletim sistemi hesap izlerini ve sistem kayıtlarını depolarlar ve toplanan veriler üzerinde çalışırlar. Bu şekilde işletim sistemine yönelen saldırılar için hangi sistem çağrılarının ve hangi kullanıcıların sorumlu olduğu tespit edilebilir. Sunucu-temelli STS'ler, denetim izi ve sistem günlük dosyaları olmak üzere iki tür bilgi kaynağı kullanır. Bunların ağ temelli sistemlere göre bir avantajı, olayların olduğu yerel sunucuyu izleme yetenekleri sayesinde ağ-temelli STS'lerin yakalayamayacağı saldırıları tespit edebilir olmalarıdır. Bu saldırılar çoğu zaman fazla trafik yaratmayan R2L veya U2R saldırılarını içerir. Bazı sunucu tabanlı STS'ler merkezi kontrollü STS'lere destek vermek için tasarlanmışlardır.

Uygulama temelli STS'ler; olayları bir yazılım uygulaması ile analiz eder ve sunucu tabanlı STS'lerin özel bir alt kümesidir. Uygulama temelli STS'lerde bilgi kaynağı olarak genellikle uygulamaya ait günlük dosyalar kullanılır. Analiz motoruna, belirli uygulamalar için o uygulamalara has özellikler bildirildiği takdirde, yetkisini aşan kullanıcıların gerçekleştirdiği saldırılar uygulama temelli STS'lerce tespit edilebilir [11].

2.4. STS'lerde Kullanılan Teknikler

STS'lerde, anormallik ve kötüye kullanım (imza) tabanlı yaklaşımları modellemek için günümüze kadar birçok teknik kullanılmıştır [12]. Bu teknikler, elde edilen verilerin modellenmesi, sınıflandırılması veya kural tablolarının oluşturulması için geliştirilmiştir. Kullanılan tekniklerden elde edilen veriler sayesinde, saldırı tespit yaklaşımlarının uygulanması için gerekli olan platform oluşturulmuştur. STS'lerde kullanılan tekniklerden bazıları aşağıda açıklanmış ve Çizelge 2.1'de sunulmuştur.

Veri Madenciliği: Veritabanındaki saklı olayları ortaya çıkarmak için yapılan bilgi açılımıdır. Paternleri ve veriler arasındaki ilişkileri bularak kural çıkarmak için kullanılır. Bu şekilde, hesap izlerini kullanarak normal kullanıcı aktiviteleri tanımlanır [22].

Kural Tabanlı (Rule Based) Sistemler: Sistem trafiğini inceleyip kurallar oluşturur ve saldırı tespiti sırasında belirlenen kurallara göre davranışlar sınıflandırılır [23].

Açıklayıcı İstatistikler (Descriptive Statistics): Kullanıcı veya sistem davranışları farklı değişkenlere göre ölçülerek istatistiksel bir model oluşturulur. Bu değişkenlerden bazıları; kullanıcı oturum girişi, oturum kapatma, belli bir zaman periyodunda erişilen dosya sayısı, kullanılan disk alanı ve hafıza olarak sıralanabilir. Kullanıcı profilleri ve hesap izleri kullanılarak normal davranışların modeli oluşturulur ve anormallik tespit edilir [17]. Kullanıcı profilinin basit istatistiklerle oluşturulup, buradan uzaklık vektörlerini (distance vector) kullanarak karar alan sistemlerdir. Davranış profili oluşturulurken, kullanılan işlemci zamanı, bir zaman periyodundaki ağ bağlantı sayısı gibi farklı ölçütler de kullanılabilir. İstatistiksel yaklaşımların dezavantajlarından biri, saldırganın bu istatistikleri öğrenerek ona göre davranış sergileyebilmesidir [19].

Eşik Değeri Tespiti: Bu model oluşturulurken spesifik olayların tekrarlama sayısı ve spesifik zaman periyodu dikkate alınır. Karşılaşılan en büyük sorun; eşik değerinin belirlenmesi ve spesifik olaylar için pencere boyutunun belirlenmesidir. Örnek

olarak; yanlış girişler, giriş/çıkış hata sayısı veya silme sayıları verilebilir. Tek başına pek güçlü değilse de büyük STS'lerde alt bileşen olarak kullanılır.

Durum Geçiş Analizi: Durum değişimi serileri oluşturularak gerçekleştirilir. Bir işin yapılması için birbirini takip eden durum sırası olduğu varsayılır ve buna göre bir seri oluşturulur. Sızmaların senaryosu çıkarıldıktan sonra, anahtar hareketler, imza hareketler olarak tanımlanır. İmza hareketler, saldırının tamamlanması için gereken en küçük hareket kümesidir. Durumlar, geçişler ve imzalar, durum geçiş diyagramı olarak grafiksel biçimde sunulur [24]. Burada tüm davranışlar durumlara karşı düşer. Eğer bir davranış daha önceden tanımlı durumlara ve durum geçişlerine denk düşen hareketler yapıyorsa saldırı olarak tanınır.

Uzman Sistemler: Belirli bir alanda sadece o alan ile ilgili bilgilerle donatılmış ve problemlere o alanda uzman bir kişinin getirdiği şekilde çözümler getirebilen bilgisayar programları olarak tarif edilebilir. Sızma belirleme sistemlerinin ilkleri kural-tabanlı (rule based) uzman sistemlerdir [11].

Örüntü Eşleme (Pattern Matching): Sistemde daha önceden tanımlanmış ve karşılaşılmaması gereken bazı sözcüklerin tanınması için kullanılır. Esnek değildir fakat basittir. Örneğin "parola dosyasını kopyala" komutu görüldüğünde bunun bir saldırı olduğunu en basit şekilde bu yöntem tespit eder [11].

STS'lerde kullanılan teknikler, literatür gözden geçirilerek Çizelge 2.1'de özetlenmiştir.

2.5. Saldırı Tipleri

Bilgisayar sistemlerine yapılan saldırılar birçok araştırmacı tarafından çeşitli şekillerde gruplandırılmıştır [25, 26]. Saldırganların sürekli kendilerini yenilemeleri ve bilgisayar sistemlerinde var olan açıkları tespit etmeleri nedeniyle saldırı tiplerinin çeşitliliği çok artmıştır. Bu çalışmada STS'lerin gelişimi sırasında önemli bir yeri olan DARPA veri kümelerinin oluşturulması sırasında belirlenen ve hala geçerliliğini koruyan saldırı tipleri esas alınmıştır. MIT Lincoln Laboratuarlarında yapılan bu çalışmada, saldırılar bilgisayar sistemine yapılan atak türlerinin kullandıkları

yöntemlere göre dört gruba ayrılmış ve DoS, U2R, R2L ve Probing olarak adlandırılmışlardır [16]. Bu saldırılar, aşağıda verilen alt başlıklarda açıklanmıştır.

Çizelge 2.1. STS'lerde kullanılan tekniklerin karşılaştırılması [21]

Teknik	**Tespit Yaklaşımı**	**Kullanılan Bilgi Kaynakları**	**Bilinen Ataklar**	**Bilinmeyen Ataklar**	**Performans**
İstatistiksel	Anormallik	Denetim verisi, kullanıcı profili	Evet	Evet	Orta
Veri Madenciliği	Anormallik	Denetim verisi, bilgi tabanı	Evet	Evet	Orta
Durum Geçiş Analizi	Kötüye Kullanım	Denetim kayıtları, bilinen atak örneklerinin durum geçiş diyagramları	Evet	Hayır	Yüksek
Dosya Kontrol	Anormallik	Sistem dosyaları	Evet	Evet	Yüksek
Örüntü Eşleme	Kötüye Kullanım	Denetim kayıtları, saldırı imzaları	Evet	Hayır	Yüksek
Protokol Analizi	Anormallik	Denetim kayıtları, bir protokolün normal kullanım modeli	Evet	Evet	Düşük
Tuş Vuruşu İzleme	Kötüye Kullanım	Bilinen saldırıların bilgi tabanı, Tuş vuruşları	Evet	Hayır	Yüksek

2.5.1. Hizmet aksattırma saldırıları

Hizmet aksattırma (DoS) saldırıları, saldırılan sistemlerin hizmetlerini engellemek amacıyla yapılırlar. Genellikle hizmetin verilmesi engellenmek istenildiğinde sisteme cevap verebileceğinden çok istek gönderilmesi ile gerçekleştirilir. DARPA veritabanında isimlendirildiği haliyle, en çok bilinen DoS saldırı tipleri, SYN flood, Smurf, UDPstorm, Pingflood, Neptune, Mailbomb gibi saldırılardır [27].

2.5.2. U2R saldırıları

U2R saldırıları, kullanıcıların normal yetkilere sahip olan kendi hesaplarından oturum açtıktan sonra yönetici yetkisine ulaşmaya çalışmasıdır. Bu şekilde sistem üzerinde istedikleri bilgilere erişebilirler. DARPA veritabanında isimlendirildiği

haliyle, en çok bilinen U2R saldırı tipleri Eject, Ffbconfig, Fdformat, Loadmodule, Perl gibi saldırılardır [27].

2.5.3. R2L saldırıları

Bu saldırı tipinde saldırgan saldırdığı makineye ağ üzerinden paketler yollayarak açıklardan yararlanmaya çalışır. Bu konuda birçok araç olması ve bu araçlara kolay erişilebilir olması nedeniyle, sistemde var olan açıklar saldırgandan önce belirlenip kapatılmamışsa oldukça etkili ve kolay bir saldırı yöntemidir. DARPA veritabanında isimlendirildiği haliyle, en çok bilinen R2L saldırı tipleri Dictionary, Guest, Imap, Named, Sendmail gibi saldırılardır [27].

2.5.4. Probing saldırıları

Probing saldırısı, ağı veya bilgisayarı tarayarak zayıflıkları tespit etmek ve sistem yapısı ile ilgili genel bir bilgiye ulaşmak için yapılmaktadır. Sistem hakkında detaylı bilgi edinildikten sonra nasıl bir saldırı yapılması gerektiği belirlenir. Probing saldırısı için kullanılan araçlar aynı zamanda güvenlik uzmanları tarafından sistemin güvenliğinin test edilmesi için de kullanılan araçlardandır. DARPA veritabanında isimlendirildiği haliyle, en çok bilinen Probe saldırı tipleri, Ipsweep, Mscan, Nmap, Saint, Satan gibi saldırılardır [27].

2.6. STS'lerin Başarı Kriterleri

Saldırı tespit sistemleri, hızlı gelişimi ve sağladığı güvenlik desteği nedeniyle büyük ağlarda standart bir araç haline gelmiştir. Her geçen gün yenileri eklenen STS'lerin, diğerlerine oranla ne kadar başarılı olduğu ya da gereksinimlere ne kadar cevap verdiği sorusunu cevaplayacak net bir cevap yoktur. STS'lerin test edilmesi için günümüze kadar birçok çalışma yapılmıştır, ancak bu çalışmalar genellikle bazı STS'lerin karşılaştırılmasından ibarettir. Bu nedenle saldırı tespit sistemlerinin başarılarını ölçmek için genel bazı kurallar ve hesaplanabilir değişkenler belirlenmiştir. Mell ve arkadaşlarının yaptığı bir çalışmada STS'lerin başarı kriterleri belirlenmiştir [28]. Bu kriterler, ilgili kaynak temel alınarak sırasıyla aşağıda alt başlıklarda açıklanmıştır.

2.6.1. Kapsam

Bu ölçüt, bir STS'nin ideal koşullarda hangi atakları tespit edebildiği olarak tanımlanır. Kötüye kullanım tespitine dayalı STS'lerde kapsam, imza sayısı ve bu imzaların standart isim düzeni ile haritalanmasını içerir. Anormallik tespiti yapan STS'lerde, hangi atakların özel metodolojilerle tespit edilebildiği bilinen ataklar kümesinin dışında kaldığına karar vermek gerekir. Ataklarda değişiklikler yapılarak türetilen ataklar bu ölçütü zorlaştırır.

2.6.2. Yanlış alarm olasılığı (probability of false alarms)

Bu ölçüt, verilen çevrede belli bir zaman aralığında STS tarafından üretilen yanlış pozitif oranını tanımlar. Yanlış alarm, tespit edilmesi planlanan büyüklük için yapılan yanlış değerlendirmeleri içerir. Yanlış değerlendirmeler iki çeşit olabilir. Biri var olan bir değeri kaçırmak diğeri var olmayan bir değeri varmış gibi tespit etmektir. Yanlış alarm ikincisidir. Genelde STS sistemlerinin başarımını ölçmekte önemli bir parametredir çünkü bir sistemde izin verilen yanlış alarm sayısı ve doğru tespit miktarı birbiriyle ilintilidir. Yanlış alarmlara izin verildikçe doğru tespit oranı artmaktadır. Sistem parametreleri ikisinin de optimum olduğu noktaya ayarlanmalıdır.

2.6.3. Tespit etme olasılığı (probability of detection)

Bu ölçüt, verilen çevrede belli bir zaman aralığında STS tarafından doğru tespit edilebilen atak oranını tanımlar. Bir STS'nin var olan saldırıların kaçını yakaladığı önemli bir parametredir. Fakat STS'ler tespit etme oranını artırırken, yanlış alarm oranını da çok yükseltmemelidir.

2.6.4. STS'ye yönelik ataklara karşı direnç

Bu ölçüt, bir STS'nin doğru çalışmasını bozmak için yapılan girişimlere karşı ne kadar dayanıklı olduğunu kanıtlar. STS'nin çalışmasını bozacak ataklarla karşılaşmak mümkün olduğundan, eğer STS bu ataktan etkilenir ve karşı koyamazsa güvenliği sağlayacak tüm işlevleri durabileceğinden sistemi koruyamaz.

2.6.5. Yüksek bant genişlikli trafiği yönetebilmek

Bu ölçüt, bir STS'nin geniş hacimli trafikle karşılaştığında ne kadar iyi olduğunu gösterir. Ağ tabanlı STS'lerin birçoğu trafik hacmi arttığında paketleri düşürmeye (drop) başladığından, atakların olduğu paketleri de kaybederler. Bu kriter ağ tabanlı sistemler için geçerlidir. Bazı STS'ler, trafik yoğunluğu çok arttığı zaman saldırı tespitinde tamamen başarısız olabilirler. Özellikle DoS saldırılarında sistemin güvenli olması için bu önemli bir parametredir.

2.6.6. Olayları ilişkilendirebilme

Bu ölçüt, bir STS'nin atak olaylarını ilişkilendirebilmede ne kadar iyi olduğunu gösterir. Farklı kaynaklardan gelen saldırı bilgilerini birleştirebilme yeteneğidir ve hâlihazırdaki STS'lerin bu konuda sınırlı yeteneği vardır.

2.6.7. Daha önce görülmemiş atakları tespit edebilme

Bu ölçüt, bir STS'nin daha önce hiç karşılaşmadığı bir atağı tespit etmede ne kadar iyi olduğunu gösterir. Bu sadece anormallik tespitine dayalı sistemler için geçerlidir. İmza temelli sistemler yeni gelen hiçbir saldırıyı tanıyamazlar ve birçok ticari STS imza tanıma yöntemine göre geliştirilmiştir.

2.6.8. Bir atağı tanımlayabilme

Saldırının varlığını tespit etmek ve saldırının tipini söyleyebilmek iki ayrı kavramdır.

STS'lerin ilk odaklandığı konu doğal olarak saldırıların varlığını tespit etmektir. Genelde saldırı tipi, saldırı olduktan sonra ağ yöneticisi tarafından çeşitli kayıtlar incelenerek ortaya çıkartılır. Fakat gene de çok genel sınıflandırmalar yapabilen STS'ler mevcuttur.

2.6.9. Atak başarısını belirleyebilme

Bilgisayar sistemlerine karşı gerçekleştirilen saldırıların hepsi başarı ile sonuçlanmaz yani sisteme zarar vermez. Genelde STS'ler sisteme zarar veren veya vermeyen

saldırıları sınıflandırmak konusunda bir çalışmayı barındırmazlar. Fakat bu özellik bir STS'nin sahip olması istenen özelliklerdendir.

2.6.10. Diğer kriterler

Kullanım kolaylığı, bakım kolaylığı, kaynak ihtiyacı, gerçekleştirilebilirlik ve kalite gibi hemen hemen tüm yazılımlarda istenen genel kriterler, diğer ölçütlerdir.

3. ZEKİ STS'LER

Bilgisayar veya ağ sistemlerine yapılan saldırıları tespit ederek güvenliğin sağlanması için geliştirilen STS'ler, her ne kadar yapılan saldırıların büyük bir çoğunluğunu tespit edebilseler de daha önce hiç karşılaşılmamış olan saldırıların büyük çoğunluğunun tespit edilememesi ve bu saldırıların sistemlerde büyük zararlara yol açması, yeni saldırı çeşitlerinin tespit edilebilme başarısının artırılması ihtiyacını getirmiştir. Bu ihtiyacın karşılanması ve hızla değişen saldırı tiplerinin karşısında, bilgi ve bilgisayar güvenliğinin sağlanması amacıyla, STS'lerin geliştirilmesinde yapay zeka yöntemleri kullanılarak STS performanslarının iyileştirilmesi hedeflenmiştir. Yapay zeka tekniklerinin öğrenilebilmesi hızlı hesaplama, genelleme matematiksel olarak modellenmesi zor olan problemlere çözüm sunabilmesi gibi özellikler, bu yaklaşımların STS'lerde kullanılmasının önemli gerekçelerindendir. Zeki yaklaşımların kullanılmaya başlaması ile birlikte anormallik tespiti yaklaşımı biraz daha ön plana çıkmış ve bu sayede anormallik tespitinin yeni saldırıları tespit edebilme yeteneği arttırılmıştır.

Bu bölümde zeki STS'ler konusuna bir giriş yapılmış, zeki STS'leri geliştirmek için kullanılan yapay zeka yöntemleri kısaca açıklanmış ve zeki STS'lerde kullanılan teknikler ve uygulamalar ile ilgili önemli bilgiler verilmiştir. Zeki yaklaşımlardan biri olan ve bu tez kapsamında zeki STS tasarımı için kullanılan yapay sinir ağları yöntemi ayrı bir alt bölümde daha detaylı açıklanarak, YSA ile geliştirilmiş zeki STS'ler ve YSA'nın STS'lerde kullanımı incelenerek daha ayrıntılı bilgiler verilmiştir.

3.1. Yapay Zeka Teknikleri ve STS'ler

Yapay zeka araştırmacılarının baştan beri ulaşmak istediği ideal, insan gibi düşünen ve davranan sistemler yaratmaktır. Ancak buna ulaşmanın güçlüğü anlaşılınca çalışmanın yönü rasyonel düşünen ve davranan sistemlerin tasarlanmasına çevrilmiştir. Geleneksel yöntemlerle çözümü zor veya imkansız olan problemlerin çözümünde kullanılan, yapay zeka teknikleri, yapay sinir ağları, bulanık mantık,

sezgisel (genetik, tabu arama, karınca koloni, ısıl işlem (tavlama) benzetimi, bağışıklık sistemi, arı) algoritmalar olarak sıralanabilir.

STS'lerde, yapay zeka teknikleri tek başlarına kullanılabildiği gibi bazı uygulamalarda birlikte de kullanılmıştır. Bu teknikler "Mühendislikte Yapay Zeka Uygulamaları-1:Yapay Sinir Ağları" [29] kitabından faydalanılarak aşağıdaki kısaca açıklanmış ve STS'lerde kullanımları hakkında bilgi verilmiştir.

3.1.1. Bulanık mantık

Bulanık mantık kavramı ilk kez 1965 yılında Prof. Lotfi Asker Zadeh tarafından ortaya atılmıştır. Bu mantık insanların günlük hayatta kullandıkları ifadelerin, bilgisayar ortamında matematiksel olarak modellenmesi prensibine dayalı bir yaklaşımdır. Klasik yöntemlerde 1 ya da 0'larla gösterilen kararların, keskin ve ani geçişleri, bulanık mantık ile yumuşatılmış ve ara değerlerle ifade edilebilir hale gelmiştir. İnsan hayatına bakıldığında ılık, yarı açık, yarı dolu gibi çok sık kullanılan kavramların bulanık mantık ile bilgisayar ortamında da ifade edilebilmesiyle problemlerin çözümü gerçekleştirilmiştir.

Bulanık mantık metodunu uygulama girişimleri, STS'nin farklı bileşenlerinin geliştirilmesi için 2000'li yıllarda başlamıştır. Bu çalışmalarla ortaya çıkan FIRE (Fuzzy Intrusion Recognition Engine) ile ağ verileri işlendikten sonra atakları tespit etmek için bulanık mantık kullanılmıştır [15]. Ağ paketleri üzerinde çalışılan FIRE'de, TCP, UDP ve ICMP için otonom ajanlar kullanılır [30]. Aynı zamanda kural tabanlı bir sistem olan bu çalışmada, güvenlik yöneticisi ve edinilen tecrübe sayesinde belirlenen bulanık kurallar oluşturulmuştur [31]. 2002 yılında bulanık mantığın, anormallik tespiti yapan bir STS'nin karar verme aşamasında kullanılması önerilmiştir [32]. Bu çalışmada bulanık mantık, uzman tarafından tavsiye edilen, bulanık "if-then" kurallarını temel alarak çözüm sunmaktadır.

3.1.2. Genetik algoritmalar

Genetik algoritmalar, canlılardaki doğal gelişim prensibine dayanmaktadır [33]. Genetik algoritmalarda; çözümlerin başlangıç popülasyonunu oluşturması için bir

yöntem, çözümleri uygunluk açısından değerlendirecek bir değerlendirme fonksiyonu, genetik yapıyı değiştirecek genetik operatörler ve popülasyonun büyüklüğü gibi kontrol parametrelerine ihtiyaç vardır.

Genetik algoritmalar STS'lerde, trafik verilerine basit kurallar uygulamak için kullanılabilir. Bu kurallar, anormal trafik verilerinden normal verileri ayırmak içindir. Veri kümesi, tcpdump ya da snort gibi trafik dinleyiciler (sniffers) kullanılarak toplanır [21]. Genetik algoritmalar öncelikle küçük boyutlu rasgele üretilmiş kurallar kümesi ile başlar, daha sonra bu kural kümesi genişletilir.

3.1.3. Yapay bağışıklık sistemi

Yapay bağışıklık sistemi (YBS), 1986 yılında yayınlanan "bağışıklık sistemi, adaptasyonu ve makine öğrenmesi (The immune system, adaptation and machine learning)" adlı makale ile başlamıştır [34]. İnsan vücudunun mikropları tanıyabilme ve ani olarak onları tahrip edebilmeleri doğal bağışıklık sistemi olarak bilinir. Bu yaklaşımda ise amaç, lenfosit aktiviteleri, doğal antikor üretimi, ön bağışıklık, dağarcık seleksiyonu, tolerans, hafıza ve bağışıklık sisteminin gelişimine benzer yaklaşımları getirmektedir.

Literatürde insan bağışıklık sistemine dayalı birçok STS çalışması sunulmuştur. Bu çalışmalardan bazıları doğal bağışıklık sistemi karakteristiklerinden esinlenerek, bilgisayar sistemlerinde anormallik tespiti için biçimsel (formal) çatı (framework) önermişlerdir. Doğal bağışıklık sisteminin, bağışıklığı tanımladığını düşündükleri 4 önemli özelliğini kullanmışlardır. Bunlar; farklılık (diversity), doğal dağıtık yapısı (distributed nature), hata toleransı (error tolerance) ve doğal dinamik yapısıdır (dynamic nature) [35].

3.1.4. Destek vektör makineleri

Destek Vektör makineleri (DVM) eğitim verilerini yüksek boyutlu bir özellik uzayında çizen ve her vektörü kendi sınıfı ile etiketleyen öğrenme makineleridir.

DVM'ler STS'lerde, özellik vektörünün seçilmesinde sıkça kullanılmıştır. Hızlı olmaları nedeniyle STS'ler için oldukça kullanışlı bir yöntemdir. DVM'ler geniş bir örnek setini öğrenebilir ve iyi ölçeklendirirler [36].

3.2. YSA ve STS'ler

Zeki yaklaşımların STS'lerde kullanılmaya başlaması, farklı birçok zeki yöntemin de kullanılabilir olduğunu göstermiştir. Bu tekniklerden hemen hemen hepsi STS'lerin geliştirilmesi için kullanılmış olsa da, elde edilen başarılı sonuçlardan dolayı en çok kullanılanlardan biri YSA'dır [37, 38].

Bu bölümde YSA'lar hakkında gerekli bilgilere ve YSA ile geliştirilmiş zeki STS çalışmalarına yer verilmiştir.

3.2.1. Yapay sinir ağları

YSA'lar yapılarına göre ileri beslemeli (feed forward) ve geri beslemeli (feed back) olarak ikiye ayrılır. İleri beslemeli ağlarda işaretler, giriş katmanından çıkış katmanına doğru tek yönde iletilirler. Geri beslemeli ağlarda, çıkış ve ara katman çıkışları kendinden önceki katmanlara ya da girişe geri beslenir. İleri beslemeli ağlara MLP, RBFN ve LVQ örnek verilebilirken, ART, SOM ve Elman ve Jordan ağları geri beslemeli ağlara örnek olarak verilebilir [29].

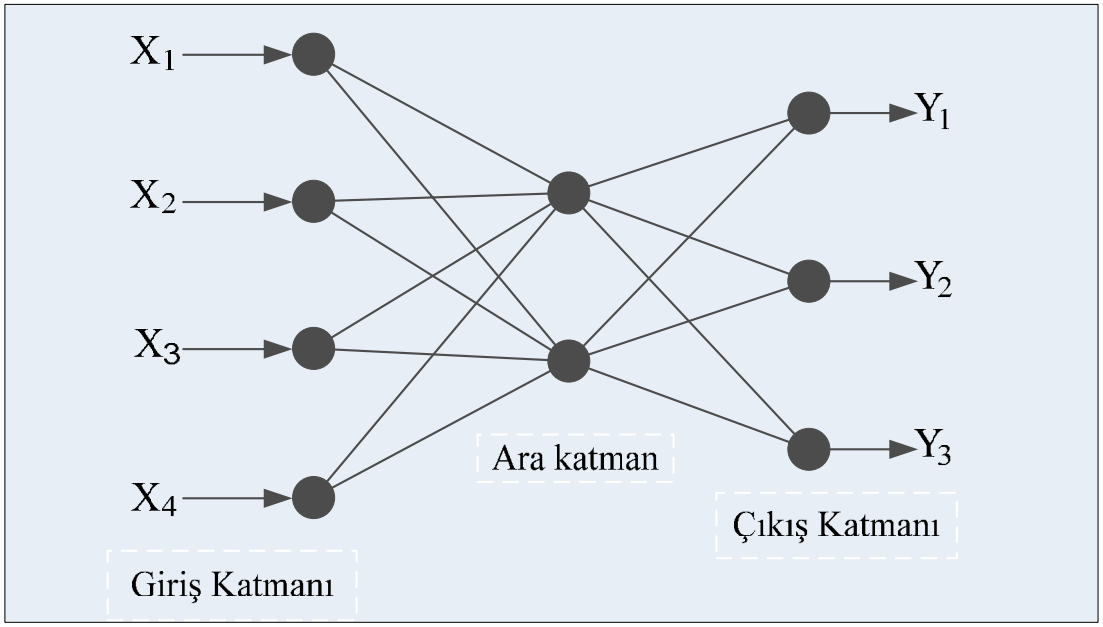
Bu çalışmada, ileri beslemeli ağ yapılarından olan MLP kullanılmıştır. MLP tercih edilmesinin nedeni, bilinen en eski YSA modellerinden olması ve sınıflandırma problemlerinde başarılı sonuçlar üretmesidir. Bunların yanında, farklı öğrenme algoritmaları ile kullanıma uygun olması MLP'nin sağladığı diğer bir avantajdır.

Şekil 3.1'de gösterilen MLP modeli, bir giriş, bir veya daha fazla ara katman ve bir de çıkış katmanı içerir. Bir katmandaki bütün nöronlar bir sonraki katmandaki bütün nöronlara bağlıdır. Giriş katındaki nöronlar tampon gibi davranırlar ve giriş sinyalini ara katmandaki nöronlara dağıtırlar. Ara katmandaki her bir nöronun çıkışı, kendine gelen bütün giriş sinyallerini takip eden bağlantı ağırlıkları ile çarpımlarının

toplanması ile elde edilir. Elde edilen bu toplam, çıkışın toplam bir fonksiyonu olarak hesaplanabilir. MLP'de giriş katmanı hariç bir nöronun çıkışı,

$$y_k = f(\sum_k w_k x) \tag{3.1}$$

ile hesaplanır. Burada, w nöronlar arasındaki ağırlık değerini, f ise kullanılan transfer fonksiyonunu gösterir.



Şekil 3.1. Bir MLP yapısı (4 giriş, 1 ara katman, 3 çıkış)

MLP'de giriş ve çıkış sayıları, uygulanacak olan probleme göre belirlenirken, ara katman sayısı ile ara katman nöron sayıları "deneme yanılma" yolu ile bulunur [29].

YSA'larda diğer bir önemli etken ise öğrenme algoritmalarıdır. Öğrenme algoritmaları danışmanlı, danışmansız ve takviyeli olmak üzere üç farklı gruba ayrılabilir. Geri Yayılım (BP), Levenberg-Marquardt (LM), Esnek Yayılım (RP), Delta-Bar-Delta (DBD) ve Hızlı Yayılım (QP) gibi algoritmalar danışmanlı öğrenme algoritmalarına, Adaptif Rezonans Teorisi (ART) ve Kendi Kendini Düzenleyen Haritalar (SOM) gibi algoritmalar ise danışmansız öğrenme algoritmalarına, Genetik Algoritmalar (GA) ise takviyeli öğrenme algoritmalarına örnek olarak verilebilir.

Bu tez çalışmasında YSA modelini eğitmek için LM öğrenme algoritması kullanılmıştır. LM algoritması, maksimum komşuluk fikri üzerine kurulmuş bir en az kareler hesaplama metodudur [29]. Bu algoritma, Gauss-Newton ve En Dik İniş (Steepest Descent) algoritmalarının en iyi özelliklerinden oluşur ve bu iki metodun kısıtlamalarını ortadan kaldırır. Genel olarak bu metot yavaş yakınsama probleminden etkilenmez.

LM algoritmasında; $E(\underset{\sim}{w})$'nin bir amaç hata fonksiyonu olduğu düşünülürse, *m* adet çıkış nöronu için hata terimi $e_i^2(\underset{\sim}{w})$ aşağıda verilmiştir [29].

$$E(\underset{\sim}{w}) = \sum_{i=1}^{m} e_i^2(\underset{\sim}{w}) = \left\| f(\underset{\sim}{w}) \right\|^2 \tag{3.2}$$

Eş. 3.2'de, $\underset{\sim}{w}$ ağırlıkları ifade ederken,

$$e_i^2(\underset{\sim}{w}) \equiv (\underset{\sim}{y}_i - \underset{\sim}{yd}_i)^2 \text{'dir.} \tag{3.3}$$

Burada, amaç fonksiyonu *f(.)* ve onun Jakobiyeni *J*'nin bir noktada $\underset{\sim}{w}$ bilindiği farz edilir.

LM öğrenme algoritmalarında hedef, parametre vektörü $\underset{\sim}{w}$'nin, $E(\underset{\sim}{w})$'yi minimum yapacak şekilde bulunmasıdır. LM algoritmasının kullanılmasıyla yeni vektör $\underset{\sim}{w}_{k+1}$, farz edilen vektör $\underset{\sim}{w}_k$'dan Eş. 3.4 yardımıyla hesaplanabilir.

$$\underset{\sim}{w}_{k+1} = \underset{\sim}{w}_k + \delta \underset{\sim}{w}_k \tag{3.4}$$

burada $\delta \underset{\sim}{w}_k$ ifadesi

$$(J_k{}^T J_k + \lambda I)\delta \underset{\sim}{w}_k = -J_k{}^T f(\underset{\sim}{w}_k) \tag{3.5}$$

eşitliğinden faydalanılarak hesaplanır. Eş. 3.5'de;

$J_k : f$'in $\underset{\sim}{w}_k$ değerlendirilmiş Jakobyeni,

λ:Marquardt parametresi, ve

I: birim veya tanımlama matrisidir.

Levenberg-Marquardt algoritmasında hesaplama akışı aşağıdaki şekilde özetlenebilir [29].

(i) $E(\underset{\sim}{w}_k)$'yı hesapla,

(ii) küçük bir λ değeri ile başla (mesela $\lambda = 0.01$),

(iii) $\delta \underset{\sim}{w}_k$ için Eş. 3.5'i çöz ve $E(\underset{\sim}{w}_k + \delta \underset{\sim}{w}_k)$ değerini hesapla,

(iv) şayet $E(\underset{\sim}{w}_k + \delta \underset{\sim}{w}_k) \geq E(\underset{\sim}{w}_k)$ λ'yı 10 kat artır ve (iii)'e git,

(v) şayet $E(\underset{\sim}{w}_k + \delta \underset{\sim}{w}_k) < E(\underset{\sim}{w}_k)$ λ'yı 10 kat azalt, $\underset{\sim}{w}_k : \underset{\sim}{w}_k \leftarrow \underset{\sim}{w}_k + \delta \underset{\sim}{w}_k$'yı güncelleştir ve (iii)'e git.

Hedef çıkışı hesaplamak için bir YSA'nın ağırlıklarının LM öğrenme algoritması kullanılarak eğitilmesi ağırlık dizisi $\underset{\sim}{w}_0$'a bir başlangıç değerinin atanması ile başlar ve hataların karelerinin toplamı e_i^2'nin hesaplanmasıyla devam eder. Her e_i^2 terimi, hedef çıkış (y) ile gerçek çıkış (y_d) arasındaki farkın karesini ifade eder. Bütün veri seti için e_i^2 hata terimlerinin tamamının elde edilmesiyle, ağırlık dizileri **(i)'**den **(v)**'ye kadar olan hesaplama akışı içerisinde, daha önce de açıklandığı gibi LM öğrenme algoritması adımlarının uygulanmasıyla adapte edilir.

3.2.2. YSA'nın STS'lerde kullanılması

Yapay sinir ağları (YSA), saldırı tespit sistemlerinde 1990'ların başında kullanılmaya başlanan zeki yaklaşım yöntemlerinden biridir. YSA'lar, anormallik tespiti yapan STS'ler için istatistiksel yöntemlere alternatif olarak önerilmiştir [38].

YSA'lar, giriş ve çıkış vektörleri arasında ilişki kurarak kendi algoritmalarını uygularlar ve genelleştirerek yeni giriş/çıkış ilişkilerini ortaya çıkarırlar. Bu yaklaşım, sistemdeki kullanıcıların davranışlarının öğrenilmesiyle gerçekleşir. Spesifik bir kullanıcı için önceki komutlardan yola çıkarak yeni komutu tahmin eder.

YSA'ların STS'lerde kullanımı, YSA'nın normal sistem davranış izleriyle eğitilmesi ile başlar. Normal veya anormal olarak sınıflandırılan olay akışları yapay sinir ağına verilir. Toplanan veriler ile sistemin davranışına bağlı olarak öğrenme değişimi yapılabilir. Yani eğitim, izin veriliyorsa sürekli hale getirilebilir. Buradaki yaklaşım, kullanıcının n adet hareket veya komutundan, sonraki hareket veya komutunun tahminen eğitilmesidir [19]. Bu tahminin yapılması için öncelikle eğitim veri seti oluşturulmalıdır, daha sonra da eğitim tamamlanmalıdır. Eğitim verileri, belirli zaman periyodunda (birkaç gün) her kullanıcı için, sistem hesaplarından toplanır. Her gün ve her kullanıcı için veriler vektörel forma sokulur ki bu vektör kullanıcının her komutu ne kadar sıklıkta yürüttüğünü gösterir. Eğitim aşamasında ise daha önce elde edilen vektörler, kullanıcıları tanımlamak için eğitilir.

YSA'lar, anormallik tespitinde kullanıldığı gibi kötüye kullanım tespitinde de kullanılan bir tekniktir. Ağ ataklarının sürekli değişen yapısı, ağ trafiğini geniş çapta analiz edebilen ve kural tabanlı sistemlerden daha esnek bir savunma sistemi ihtiyacını doğurmuştur. YSA tabanlı kötüye kullanım tespiti yapan sistemlerle, kural tabanlı sistemlerde var olan bu tür problemlere çözüm sağlanabilmektedir [38].

YSA'lar, kötüye kullanım tespitinde iki farklı şekilde kullanılmaktadır [38]. İlk yaklaşımda yapay sinir ağları, zaten var olan bir uzman sistemin bir parçası olarak kullanılmaktadır. Bu yaklaşımda, saldırı tespitinin daha etkin yapılması amacıyla gelen verilerin filtrelenmesi ve uzman sisteme gönderilmesinde YSA'lardan faydalanılır. İkinci yaklaşımda ise, YSA'lar tek başına bir sistem olarak kötüye kullanımı tespit etmekte kullanılır. Bu yaklaşımda YSA, ağ akış bilgilerinden, kötüye kullanım tespitinin analizinde kullanılır.

YSA'lar, STS'lerde karşılaşılan pek çok problemlere esnek çözümler sunabildikleri STS tasarımlarında önemli çözümlerin başında gelmektedir. YSA'nın çözüm

sunduğu temel iki problem, veri redaksiyonu ve sınıflandırmadır [39]. Veri redaksiyonu (azaltma), işleme zamanını, iletişim yükünü ve depolama gereksinimlerini azaltmak amacıyla veri koleksiyonunu analiz ederek en önemli girişleri tanımlama işidir. Sınıflandırma ise saldırganları ve atak yapanları tanımlama işlemidir. YSA'lar pek çok STS'lerde bu iki önemli problemi çözmek için kullanılmışlardır [39].

STS'lerde karşılaşılan diğer problemler, istatistiksel yayılımı doğrulama ihtiyacı, tespit ölçütlerinin değerlendirilmesinin zorluğu, algoritma geliştirmenin yüksek maliyeti ve ölçeklemede zorluk olarak sıralandırılabilir [40]. Bu problemler aşağıda açıklanmıştır;

İstatistiksel yayılımı doğrulama ihtiyacı: İstatistiksel metotlar kullanıcı davranışlarının yayılımı hakkında varsayımlara dayanır. Bu varsayımlar doğru olmayabilir ve yanlış alarm oranının artmasına neden olabilir. YSA'lar veri yayılımında bu tür varsayımları ortadan kaldıran çözümler sunarlar.

Tespit ölçütlerinin değerlendirilmesinin zorluğu: İstatistiksel metotlarda belirlenen ölçütler deneyim ve gözlemler ile elde edilir. Ancak bu belirleme sırasında bir ölçütün ne kadar önemli olduğu kesin olarak tespit edilemez. Genelde etkili olmadığı için kullanılmayan bir ölçüt, özel bir çözüm için önemli olabilir. Yapay sinir ağları, çeşitli ölçüt kümelerinin ne kadar etkili olduğunu değerlendirmede de çözümler sunabilir.

Algoritma geliştirmenin yüksek maliyeti: Yeni bir istatistiksel algoritma önermenin ve yeni bir yazılım geliştirmenin süresi önemlidir. Algoritmanın tekrar yapılandırılması ve yazılım uygulamasının tekrar oluşturulması maliyetli bir iştir. YSA simülatörleri ise modifiye işlemi için daha kolaydır, daha az çaba ve zamanda kullanılan yöntem değiştirilebilir.

Ölçeklemede zorluk: STS'lerin geniş topluluklarda kullanılması ile binlerce kullanıcıya göre çalışmak gibi yeni problemlerle karşı karşıya gelinmesine sebep olmuştur. Bu problemi çözmek için, güvenlik yöneticisi tarafından kullanıcıların iş

tanımları veya sorumluluklara göre gruplandırılması gerekebilir. YSA kullanılarak, var olan davranışlarına göre sınıflandırılma yapılması, grupları izlemeyi daha etkili hale getirecektir.

STS'lerin tasarımında farklı çözümler sunabilen farklı YSA çalışmaları aşağıda özetlenmiştir.

90'lı yıllarda geliştirilen sistemde, kullanıcı tarafından daha önce girilen komutların sırasına bağlı olarak, sonraki komutları tahmin etmek için kullanılmıştır [40]. Gerçekleştirilen sistemde YSA'ya giriş olarak o an ve geçmişteki belirli sayıda komut verilir. Aynı zamanda pencere boyutu olan bu sayının, sistemde önemli bir rolü vardır. Eğer bu sayı çok küçük olursa "yanlış alarm (false positive)" artar, eğer bu sayı çok büyük olursa bazı ataklar tespit edilemez [40].

Ryan ve arkadaşlarının geliştirdiği NNID (neural network intrusion detector), kullanıcılar tarafından kullanılan komutların dağılımına göre saldırıları tespit etmeye yöneliktir [41]. Sistemin üç safhası vardır. İlk safhada, kullanıcıların belli periyotlarda komut yürütme dağılımlarının gösterilmesi için her kullanıcı için günlük kayıtlarından eğitim verileri toplanır. İkinci safhada, komut dağılım vektörlerine bağlı olarak kullanıcıları tanımlamak için YSA eğitilir. Üçüncü safhada ise YSA, test için verilen her yeni komut dağılım vektörü için kullanıcıları tanımlar. YSA'nın bulduğu sonuç tanımlanmış kullanıcılardan farklı olursa, sistemde anormallik olduğu tespit edilmiş olunur [40].

YSA'nın STS'lerde kullanılmasının avantajları;

(1) Kötüye kullanım ataklarının karakteristiğini öğrenmesi ve daha önce ağda kaydedilen örneklere benzemeyenleri ayırt edebilmesi,
(2) Hızlı sonuç üretmesi sayesinde gerçek zamanlı uygulamalar için kullanışlı olması,
(3) Gürültülü verilerle de çalışabildiklerinden, gürültülü verilerin sonuçları diğer yöntemlere göre daha az bozması,
(4) Farklı sistemlerle beraber çalışabilmeleri,

(5) Genel olarak çözüm sağlayabilmeleri yanında kısmi olarak da STS'lerin tasarımında kullanılabilmeleri,
(6) Az örneklerde sistemin veya atağın genel davranışını öğrenebilme yetenekleri olarak sıralanabilir.

Tüm bu avantajların yanı sıra, en büyük problem yapay sinir ağlarının eğitilmesidir. Yapay sinir ağı, etkin şekilde çalışması için iyi eğitilmelidir ve geniş bir veri seti kullanılmalıdır.

Bu çalışma kapsamında, zeki STS'lerde kullanılan teknikler gözden geçirilerek Çizelge 3.1'de sunulmuştur.

Çizelge 3.1. Zeki STS'lerde kullanılan tekniklerin karşılaştırılması [21]

Teknik	**Tespit Yaklaşımı**	**Kullanılan Bilgi Kaynakları**	**Bilinen Ataklar**	**Bilinmeyen Ataklar**	**Performans**
Yapay Sinir Ağları	Anormallik	Denetim verisi, Komut Sırası	Evet	Evet	Orta
Yapay Bağışıklık Sistemi	Anormallik	Denetim Kayıtları, Sistem Çağrı Sırası	Evet	Evet	Orta
Genetik Algoritma	Kötüye Kullanım	Denetim verisi, Bilinen Atak Örnekleri	Evet	Hayır	Düşük
Bulanık Mantık	Kötüye Kullanım	Denetim Kayıtları, Komut Sırası	Evet	Hayır	Yüksek

Bu tez kapsamında, literatürdeki bazı zeki STS çalışmaları incelenerek Çizelge 3.2'de sunulmuştur.

Çizelge 3.2. Literatürdeki örneklerin karşılaştırılması

Yazar	Teknik	YSA Yapısı	Nöron Sayısı	Veri Kümesi	Özellik Vek.(giriş)	Atak Tipleri	Örnek Say. (eğt-test)	Başarı (%). test
[38]	YSA	MLP	9-x-x-2	3000 örnek (simule edilmiş atak).	Protokol_id, Src_port, Dest_port, Src_ip, Dest_ip, Icmp_type, Icmp_code, Raw_data_lenght, Raw_data	SYNflood, SATAN, ISS Scan	8462-1000	91,00
[10]	YSA	MLP	12-25-x	6 haftalık BSM verileri	---	---	sinyal sayıları 22444-25457	---
[41]	YSA	MLP	100-30-10	2 günlük veriler	100 önemli komut	---	65-24	96,00
[42]	YSA	MLP	---	---	---	SYNflood, SYNport Scan, Stealthy	4000-6000	---
[43]	YSA	SOM	8-8-1	DARPA 99	her kullanıcı için, dinlenen portlardan geçen paket sayısı	DoS, Probe, U2R, R2L, Normal	---	---
[27]	YSA ve DVM	MLP	41-40-40-1	KDD'99	41	DoS, Probe, U2R, R2L, Normal	7312-6980	99,25

Çizelge 3.2. (Devam) Literatürdeki örneklerin karşılaştırılması

[44]	YBS	---	---	---	service, source port, source host, destination host, duration, bytes sent by source host, bytes sent by destination, flag	---	---	---
[45]	YSA	MLP	35-35-35-3	KDD'99	35	Normal, Satan ve Neptune	900	
[46]	YSA ve DVM	---	---	KDD'99	41	DoS, Probe, U2R, R2L, Normal	5092 - 6890	97,04 doğruluk, 2,76 false positive, 0.20 false negatif
[47]	YSA ve DVM	MLP	---	KDD'99	41	DoS, Probe, U2R, R2L, Normal	5092 - 6890	99,60 Dos, 85,70 Probe, 34,30 U2R, 99,30 R2L, 96,40 Normal

4. MEVCUT STS YAZILIMLARI

Saldırı tespit sistemlerinin, bilgi ve bilgisayar güvenliğindeki yerini almaya başlaması ile birlikte, bu konuda yapılan çalışmalarda artmıştır. Araştırmacılar, farklı ortamlarda, farklı sistemler üzerinde ve farklı tekniklerle birçok STS yazılımı geliştirmişlerdir [11]. Bu çalışmalar, Çizelge 4.1'de listelenerek, karakteristik özelliklerine göre karşılaştırılmıştır [11, 48].

Haystack, MIDAS, IDES, W&S, Computer Watch, NSM, NADIR, Hyperview, DIDS, USTAT, IDIOT, Janus, EMERALD, Bro, Ripper, Snort ve Snortsam bunlardan önemli olanlarıdır. Çizelge 4.1'de listelenen ve STS'lerin gelişiminde önemli yer tutan bazı çalışmalar aşağıda alt başlıklarda kısaca açıklanmıştır.

4.1. Haystack

Bu STS, 1988 yılında Amerikan Hava Kuvvetlerinde kullanılan, OS/1100 işletim sistemini çalıştıran çok kullanıcılı Unisys 1100/60 ana bilgisayarları için tasarlanmış bir saldırı tespit sistemidir [49].

Haystack aslında 6 farklı tür saldırının tespiti için dizayn edilmiştir.

- Kırma girişimleri (attempted break-ins)
- Kılık değiştirilen saldırılar (disguised intrusion)
- Güvenlik kontrol sistemine sızma (penetration of the security control system)
- Sızma (leakage)
- Hizmet aksattırma (denial of service)
- Kötü niyetli kullanım (malicious use)

Sistem iki platforma ayrılmıştır. Unisys (Sperry) çalışma sistemi veri toplamayı denetlemekten sorumludur. Haystack'in Unisys parçası daha sonra bu denetleme takibini, birleşmiş denetleme takibine dönüştürür.

Çizelge 4.1. STS yazılımlarının karşılaştırılması

Yıl	STS	Saldırı Tespiti Yöntemi	Gerçek Zamanlı	Veri Kaynağı	Mimari yapı (veri işleme)
1988	Haystack	Hibrit	Hayır	Sunucu	Merkezi
1988	MIDAS	Hibrit	Evet	Sunucu	Merkezi
1988	IDES	Anormallik	Evet	Sunucu	Merkezi
1989	W&S	Anormallik	Evet	Sunucu	Merkezi
1990	Comp-Watch	Anormallik	Hayır	Sunucu	Merkezi
1990	NSM	Hibrit	Evet	Ağ	Merkezi
1991	NADIR	İmza	Hayır	Sunucu	Merkezi
1992	Hyperview	Hibrit	Evet	Sunucu	Merkezi
1992	DIDS	Hibrit	Evet	Hibrit	Dağıtık
1992	ASAX	İmza	Evet	Sunucu	Merkezi
1993	USTAT	İmza	Evet	Sunucu	Merkezi
1994	DPEM	İmza	Evet	Sunucu	Dağıtık
1994	IDIOT	İmza	Evet	Sunucu	Merkezi
1995	NIDES	Hibrit	Evet	Sunucu	Merkezi
1996	GrIDS	Hibrit	Hayır	Hibrit	Dağıtık
1996	CSM	İmza	Evet	Sunucu	Dağıtık
1996	JANUS	İmza	Evet	Sunucu	Merkezi
1997	JiNao	Hibrit	Evet	Sunucu	Dağıtık
1997	EMERALD	Hibrit	Evet	Hibrit	Dağıtık
1998	Bro	İmza	Evet	Ağ	Merkezi
1997	Anzen flight jacket	Hibrit	Evet	Ağ	Hibrit
1998	Centrax security suite	Hibrit	Evet	Hibrit	Dağıtık
1991	Computer misuse detection system	Hibrit	Evet	Sunucu	Dağıtık
1998	Cross-site for security	İmza	Evet	Ağ	Dağıtık
1999	Cybercop monitor	İmza	Evet	Sunucu	Dağıtık
1992	Intruder alert	İmza	Evet	Hibrit	Dağıtık
1996	Kane security monitor	İmza	Evet	Sunucu	Dağıtık
1997	Netprowler (née id-trak)	İmza	Evet	Sunucu	Dağıtık
1996	Netranger	İmza	Evet	Ağ	Dağıtık
1996	Realsecure	İmza	Evet	Ağ	Dağıtık
1997	Securenet pro	İmza	Evet	Ağ	Dağıtık
1997	Sessionwall-3	İmza	Evet	Ağ	Dağıtık
1998	Smartwatch	İmza	Evet	Sunucu	Dağıtık
1998	Stake out	İmza	Evet	Ağ	Hibrit
1998	Tripwire	İmza	Hayır	Sunucu	Merkezi
2000	FIRE	Anormallik	Hayır	Ağ	Merkezi

4.2. MIDAS (Multics Intrusion Detection and Alerting System)

MIDAS, NCSC (Ulusal Bilgisayar Güvenlik Merkezi) tarafından, Bilgisayar Bilimleri laboratuarı (Computer Science Laboratory) ve SRI (Stanford Araştırma Enstitüsü) işbirliği ile, NCSC'nin ağa bağlanmış ana bilgisayarı Honeywell DPS-8/70 için saldırı tespiti sağlamak üzere geliştirilmiştir [11].

MIDAS, sezgisel saldırı tespiti konsepti çevresinde geliştirilmiş uzman sistem tabanlı bir STS'dir.

4.3. IDES (Intrusion Detection Expert System)

IDES, Denning ve Neuman'ın 1985'li yıllarda yaptıkları çalışmalar ile gündeme gelmiştir [50]. IDES için gereksinimleri ve tasarım modelini ortaya koyan bu çalışmanın ardından, IDES üzerinde yapılan çalışmalar birçok araştırmacının da katkıları ile 1987-1992 yılları arasında yoğun olarak devam etmiştir [6, 7, 40]. Bu nedenle, bu yıllarda gerçekleştirilen birçok çalışmayı içermektedir.

IDES prototipi, denetleme verilerinde raporlanan kullanıcı davranışını, kullanıcının geçmiş hareket profiline göre normal olarak belirler [40]. Zamanla değişebilen kullanıcı davranışlarına karşılık hareket profilleri her gün yenilenir.

IDES, bir uzman sistem bileşeni içerdiğinden, önceki saldırılardan edinilen bilgilere dayanan özel davranışları, bilinen sistem açıklarını veya kuruluma özel güvenlik politikalarını tanımlayan kurallar içerir [40].

IDES'in ilk orjinal versiyonu tek bir sunucu için tasarlanmıştır. Daha sonra yapılan çalışmalarda tekrar dizayn edilerek, farklı birçok hedef sistem için kullanılabilir hale getirilmiştir. Pek çok versiyonu olan bu yazılım, son olarak NIDES (Next-Generation Intrusion Detection Expert System) adını almıştır [51].

4.4. W&S (Wisdom and Sense)

W&S (Wisdom and Sense – Bilge ve His), Los Alamos Ulusal Laboratuarlarında geliştirilen anormallik tabanlı bir çalışmadır. Geliştirilmesine 1984 yılında

başlanmasına rağmen ilk çalışma 1989'da sunulmuştur. Bilge kısmı (W), geçmişteki denetleme verilerini inceleyerek normal davranışı oluşturması anlamına gelir. His kısmı (S) ise bunların kural haline getirilip uzman sisteme verildikten sonra anormal davranışların yakalanması anlamına gelmektedir [52].

4.5. NSM (Network Security Monitor)

NSM'de IDES gibi revizyonlardan geçmiştir ve geliştirilmesi 1990-94 arasına denk düşer. NSM, ağı dinleyerek, ağın kullanımıyla ilgili bir profil geliştirir ve geçerli kullanımı onunla karşılaştırır. Elde edilen veri beklenen bağlantı verisiyle karşılaştırılır ve beklenen aralıkta çıkmayan her veri anormal olarak işaretlenir [11, 53].

4.6. NADIR (The Network Anomaly Detection and Intrusion Reporter)

NADIR (The Network Anomaly Detection and Intrusion Reporter) 1991-1993 yılları arasında Los Alamos Ulusal Laboratuarlarında (LANL-Los Alamos National Laboratory) bilgisayar ağları mühendisliği grubu tarafından geliştirilmiştir [53]. NADIR kullanıcılar hakkında haftalık istatistikler tutar, her hafta profil analizinin sonucunda elde edilen detaylı raporlar oluşturulur. Daha sonra bu istatistikleri uzman sistem kurallarıyla karşılaştırır [54].

4.7. Hyperview

1992 yılında geliştirilmiş olan Hyperview, diğer sistemlerden oldukça farklı bir sistemdir. İki ayrı bölümden oluşur. İlki davranışları izleyen ve sınıflayan bir uzman sistem, ikincisi buradan öğrendikleriyle eğitilen yapay sinir ağlarını içeren bölümdür [11].

4.8. USTAT (State Transition Analysis Technique for Unix Systems)

1993-1995 yılları arasında geliştirilmiştir. Durum geçiş analizi yöntemi kullanılmıştır [55]. Bu STS'de, eğer bir davranış saldırı için tanımlı durum geçişlerini yapıyorsa saldırı olarak sınıflandırılır.

4.9. DIDS (Distributed Intrusion Detection System)

1992 yıllarında geliştirilmiş dağıtık mimarili sistemleri kapsar [56]. Bu tip sistemlerde ağın değişik noktalarından veriler toplanır ve bir merkezde incelenir.

4.10. IDIOT (Intrusion Detection In Our Time)

1994-1996 yılları arasında CERIAS (Center for Education and Research in Information Assurance and Security)'da Kumar tarafından geliştirilmiştir [11, 57]. Kumar'ın tasarımı olan IDIOT, saldırı yöntemlerinin eşleştirme ve geçici karakteristiklerin karmaşıklığına dayalı olan bir sınıflandırma yöntemidir.

4.11. Janus

1994-1996'da Berkeley'de Wagner tarafından tez çalışması sırasında geliştirilmiş bir sistemdir [11].

4.12. EMERALD (Event Monitoring Enabling Responses to Anomalous Live Disturbances)

EMERALD (Event Monitoring Enabling Responses to Anomalous Live Disturbances) 1997-1998 yıllarında geliştirilmiş ölçeklenebilir, dağıtık bir STS'dir [58]. Bu alanda önemli çalışmalardan birisi olup, pek çok çalışmada kaynak olarak alınan bir sistemdir.

4.13. Bro

1998'de Vern Paxson tarafından geliştirilmiş bir STS'dir. Gerçek zamanda ağ trafiğini pasif olarak gözlemleyerek saldırı tespiti yapmaktadır [59]. Birçok değişik özelliği bünyesinde barındırması açısından sıkça kaynak verilen bir STS'dir.

4.14. Ripper

1999 yılında geliştirilen bu sistem veri madenciliği yaklaşımını kullanır. Ripper DARPA değerlendirmesine katılmış olan sistemlerden biridir [11].

4.15. Snort

1990'ların sonunda Martin Roesch tarafından geliştirilen Snort, gerçek zamanlı trafik analizi yapabilen ve paket kaydedebilen açık kaynak kodlu kural tabanlı bir saldırı engelleme ve tespit sistemidir. İmza ve anormallik tespiti yaklaşımlarının avantajlarını üzerinde barındırır [60]. Protokol ve içerik analizleri yaparak birçok saldırı ve sızma girişimini tespit eder ve çeşitli alarm mekanizmaları ile kullanıcıyı uyarır. Saldırı imzalarının düzenli olarak güncellenmesi sayesinde oldukça farklı sayıda ve türde saldırıyı tespit edebilmek mümkündür [60]. Açık kaynak kodlu ve sürekli geliştirilmeye açık olması popülerliğini artırmış bununla birlikte ticari alternatifleri ile kıyaslanabilir duruma gelmesini sağlamıştır.

Ticari yazılımlarda kaynak kodun açık olmaması, ordu, kamu kuruluşları ve özel sektör gibi üst düzey güvenlik isteyen kuruluşlar tarafından Snort'un daha çok tercih edilir olmasını sağlamıştır. Farklı kullanım modları sayesinde kullanıcıya kolaylık sağlaması ve yönetilebilir olması ise diğer önemli özelliklerindendir.

4.16. SnortSam

Snortsam, Snort STS sistemine SES özelliği katan bir eklentidir. Snortsam iki ana kısımdan oluşmuştur. Bu kısımlardan biri Snort için çıkış sistemidir, diğeri de güvenlik duvarı (GD) üzerinde ajan vazifesi görecek kısımdır. Snortsam kurulduktan sonra Snort kurallarına "fwsam" anahtar kelimesi eklenir. Snortsam'ın kullanımı bu anahtar kelimeyle yapılır. Snort üzerine yazılan kurallar snortsam ajanı aracılığı ile GD'ye aktarılır ve engellenmesi gereken trafik GD tarafından durdurulur.

5. STS VERİ KÜMELERİ

Bir STS tasarlanırken ele alınması gereken en önemli hususlardan birisi kullanılacak olan veri kümesidir. STS veri kümesi, geliştirilecek olan STS'nin eğitim ve test aşamalarında saldırıyı tanımlamak için gereken ve içerisinde saldırı verileri içeren ağ paketleri veya günlük kayıtlardan elde edilen veriler bütünüdür. STS çalışmalarının bazılarında, uygulama geliştiriciler, veri kümelerini kendileri oluşturmuşlardır. Ancak bu oldukça zahmetli, bir o kadar da maliyetli bir çalışmadır. STS geliştiricilerinin, sınırlı olan işgücü ve zamanı, yeni ve farklı teknikler kullanmak yerine, veri kümesi oluşturmaya harcaması yüklerini oldukça artırmıştır. Bunun yanında, geliştirilen sistem kendilerinin tanımladığı veri kümesi üzerinde yüksek başarı oranları gösterse de, bu sonuçlar gerçeklikten uzaktır. Hem STS'lerin tasarım ve uygulama çalışmalarının hızlandırılması hem de standart ve objektif bir test ortamı oluşturulabilmesi için güncel ve standart hale gelmiş STS veri kümelerine ihtiyaç duyulacağı aşikardır.

Literatüre bakıldığında, bu ihtiyaca yönelik bazı çalışmalar yapılmıştır [16, 17, 28]. Ancak 1998-2000 yılları ile sınırlı kalan bu çalışmaların, o yıllarda çok değerli sonuçlara ve değerlendirmelere imza atmış olsalar da güncelliklerini kaybettikleri görülmektedir. Bu nedenle STS'ler için; yeni, gerçeğe uygun ve güncel saldırıları da kapsayan veri kümelerinin oluşturulması gerekmektedir.

Bu tez çalışması esnasında karşılaşılan güçlükler arasında olan güncel veri kümeleri oluşturma konusunda çalışmalar yapılmış ve tez kapsamında bazı önerilerde bulunulmuştur. Bu önerilere geçmeden önce, bu konuda daha önce yapılmış ve halen kullanılmaya devam eden IDEVAL ve KDD'99 veri kümeleri hakkında bilgiler verilmiştir.

5.1. IDEVAL (Intrusion Detection Evaluation)

MIT Lincoln Laboratuarları, Bilgi Sistemleri Teknolojisi (Information Systems Technology - IST) grubunun, DARPA (Defence Advanced Research Projects Agency) ve AFRL (Hava Kuvvetleri Araştırma Laboratuarı - Air Force Research

Projects Agency) desteğiyle yürüttüğü çalışmaların sonucunda, saldırı tespit sistemlerinin değerlendirilmesi ve karşılaştırılması için IDEVAL (Intrusion Detection Evaluation) adı verilen ilk standart veri kümesi gövdesi oluşturulmuştur. Bu çalışmada, Hava Kuvvetleri Araştırma Laboratuarı ile birlikte saldırı tespit sistemlerinin, ilk düzenli, tekrarlanabilir ve önemli-istatistiksel ölçütleri belirlenmiştir. Bu ölçütler, test altındaki her sistem için tespit olasılığı ve yanlış-alarm olasılığını kapsamaktadır. Bu ölçütler, mevcut saldırı tespit sistemleri tasarımına, yeni yapılacak araştırma çalışmalarına yön verme ve objektif bir kalibrasyon sağlama gibi önemli hususları desteklemektedir.

5.1.1. IDEVAL veri kümeleri

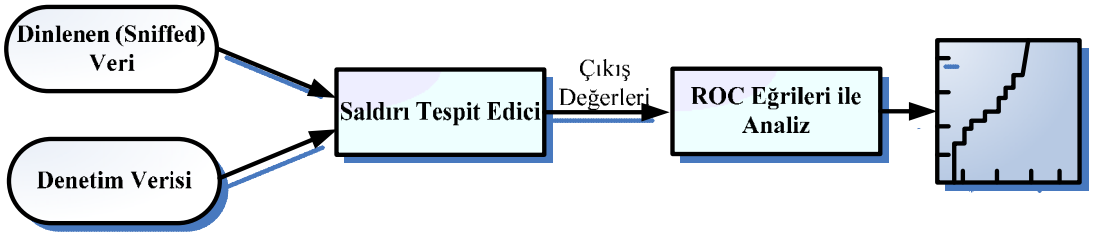
Lincoln Laboratuarlarında gerçekleştirilen, saldırı tespit sistemlerini değerlendirme çalışmaları 1998-2000 yıllarında yürütülmüştür. Oluşturulan gerçek zamanlı olmayan (off-line) veri kümeleri, geniş atak ve ağ trafiği örnekleri ile araştırmacılar tarafından kullanılmaya uygun hale getirilmiştir.

DARPA saldırı tespit değerlendirme çalışmaları 1998 ve 1999 DARPA IDEVAL veri kümeleri olmak üzere iki veri kümesi elde edilmiştir [61, 62].

Bu veri kümelerinin haricinde 2000 yılında özel senaryolar için 3 farklı veri kümesi daha oluşturulmuş ve bu veri kümeleri araştırmacıların kullanımına sunulmuştur [63]. Bu tez çalışmasında, günümüzde halen kullanılmaya devam eden ve ağ trafiğinin dinlenmesi ile oluşturulan 1998 ve 1999 veri kümeleri kullanılmıştır. DARPA 2000 veri kümeleri sadece özel senaryoları içerdiğinden ve bu çalışmanın amacına uygun olmadığından kullanılmamıştır.

IDEVAL 1998 ve 1999 veri kümeleri, gerçek zamanlı ve gerçek zamanlı olmayan değerlendirme olmak üzere iki kısımdan oluşmaktadır. Lincoln Laboratuarlarında oluşturulan veri kümeleri, STS'lerin gerçek zamanlı olmayan (off-line) değerlendirilmesinde kullanılır. IDEVAL 1998 ve 1999 değerlendirmesinin gerçek zamanlı kısmı hakkında ayrıntılı bilgi AFRL'den elde edilebilir.

Lincoln Laboratuarlarının sunduğu gerçek zamanlı olmayan değerlendirme aşamaları Şekil 5.1'de gösterilmiştir.



Şekil 5.1. Lincoln laboratuarlarında gerçekleştirilen, gerçek zamanlı olmayan değerlendirme aşamaları

Şekil 5.1'den de görülebileceği gibi ağın dinlenmesi ve denetlenmesi ile elde edilen veriler saldırı tespit sistemine gönderilmiştir. Saldırı tespit sisteminden elde edilen sonuçlar, yanlış alarm (false positive), yanlış negatif ve doğru tespit oranlarına göre değerlendirilerek ROC eğrileri ile analiz edilmiştir. STS sonuçlarına göre yanlış alarm ve tespit oranının hesaplanması şu şekildedir [64].

Yanlış alarm oranı = yanlış alarm sayısı / toplam normal bağlantı sayısı (5.1)

Yanlış negatif oranı = yanlış negatif sayısı / toplam atak sayısı (5.2)

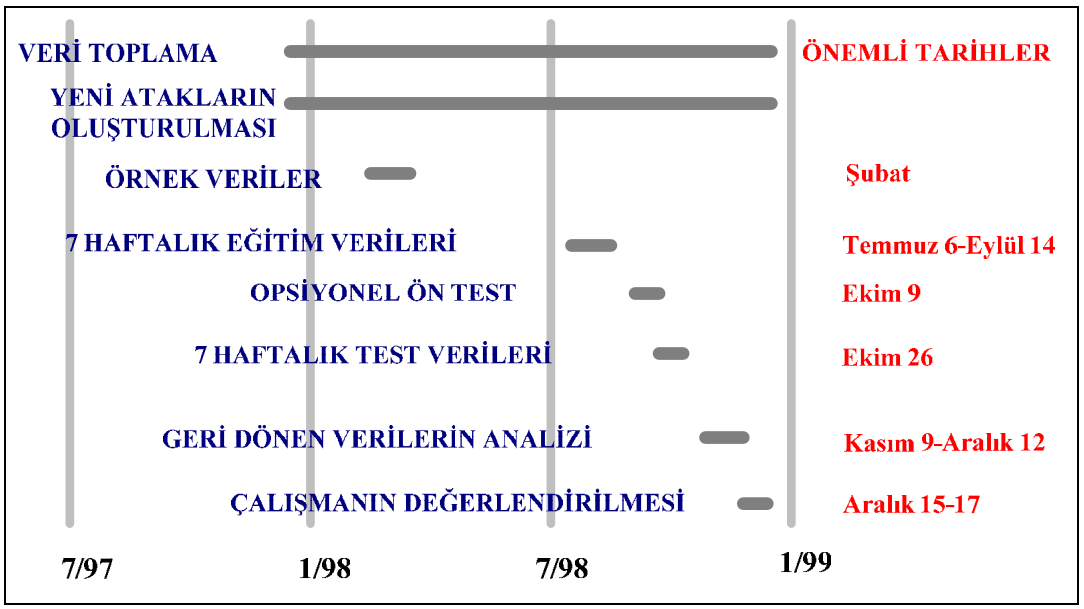
Doğru tespit oranı = 1 - yanlış negatif sayısı / toplam atak sayısı (5.3)

Lincoln Laboratuarlarında gerçekleştirilen değerlendirme süreçleri, hem DARPA 1998 ve hem de DARPA 1999 için geçerlidir. Bu iki veri kümesini birbirinden farklı kılan durum, simülasyon ortamlarında ve saldırı veri kümelerinde gerçekleştirilen yapısal değişikliklerdir. Bu nedenle 1998 veri kümesinin oluşturulma aşamalarına ayrıntılı olarak yer verilirken, 1999 veri kümesinde ise gerçekleştirilen değişikliklerden bahsedilecektir.

5.1.2. 1998 DARPA

1998 DARPA veri kümesi, 1998 IDEVAL veri kümesinin gerçek zamanlı olmayan değerlendirme kısmı için geliştirilmiştir [61].

1998 veri kümesinin oluşturulma aşamalarını gösteren zaman çizelgesi Şekil 5.2'de gösterilmiştir [61].



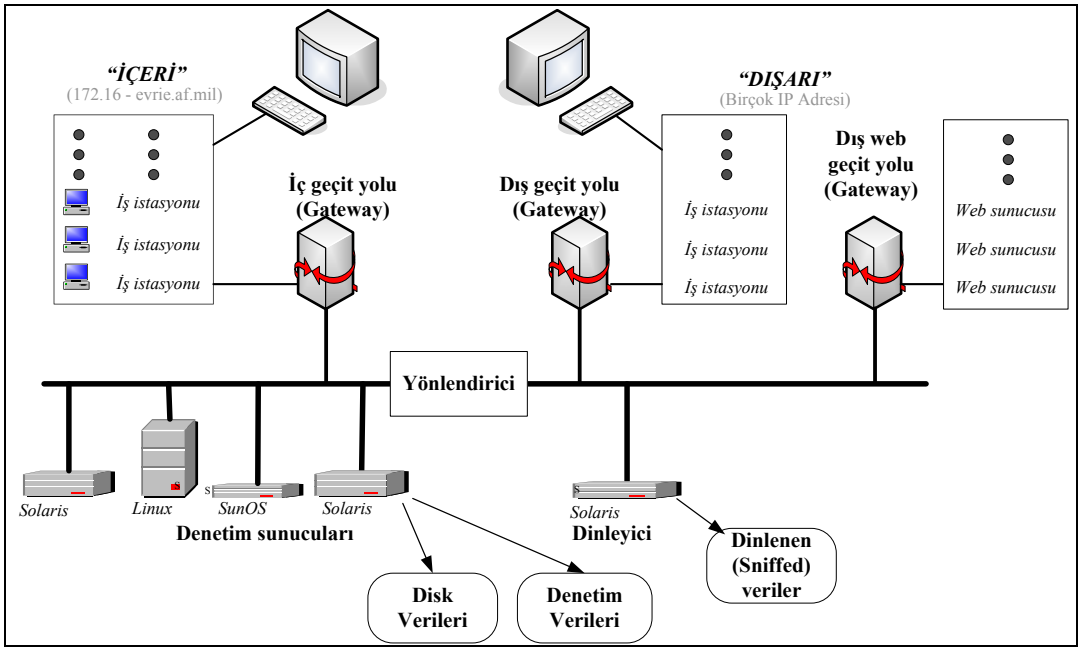
Şekil 5.2. 1998 DARPA veri kümesinin oluşturulma süreçleri

Şubat 1998'de 10 dakikalık örnek bir veri kümesi oluşturulmuştur. Bu veri kümesi STS'lerin nasıl skorlandığını göstermek amacıyla, sadece bazı örnek saldırı tiplerini içerir. Bunlar; guess, ping-sweep, port-scan, phf, rlogin, rsh ve rcp'dir. Oluşturulan örnek veri kümesi EK-1'de sunulmuştur.

Örnek veri kümesinden sonra, Mayıs 1998'de eğitim verilerinin alt kümesi olan 4 saatlik veri kümesi oluşturulmuştur. Bu veri kümesinin amacı, bazı ilk verilerin oluşturularak, verinin doğru okunabildiğini ve değerlendirme için yeterli bilginin sağlanabildiğini gösterebilmektir. Oluşturulan veri kümesi, gerçek eğitim veri kümesinin ilk 4 saati değildir ve ilk hafta elde edilen verilere benzer olmasına rağmen özdeşi de değildir. 4 saatlik veri kümesi, örnek 10 dakikalık veri kümesindeki saldırıların haricinde 2 farklı saldırı içerir, bunlar, EK-2'de yer alan dict ve eject'dir.

1998 IDEVAL eğitim veri kümeleri 7 haftalık verilerden oluşur. Bu verilerin oluşturulması sırasında kullanılan simulasyon ortamının ayrıntıları Şekil 5.3'de

verilmiştir [61]. Bu ağda iç ve dış olmak üzere iki ortam oluşturulmuştur. İçerideki sunucularda denetim verileri ve ağ paketlerinin toplanması sağlanmıştır.

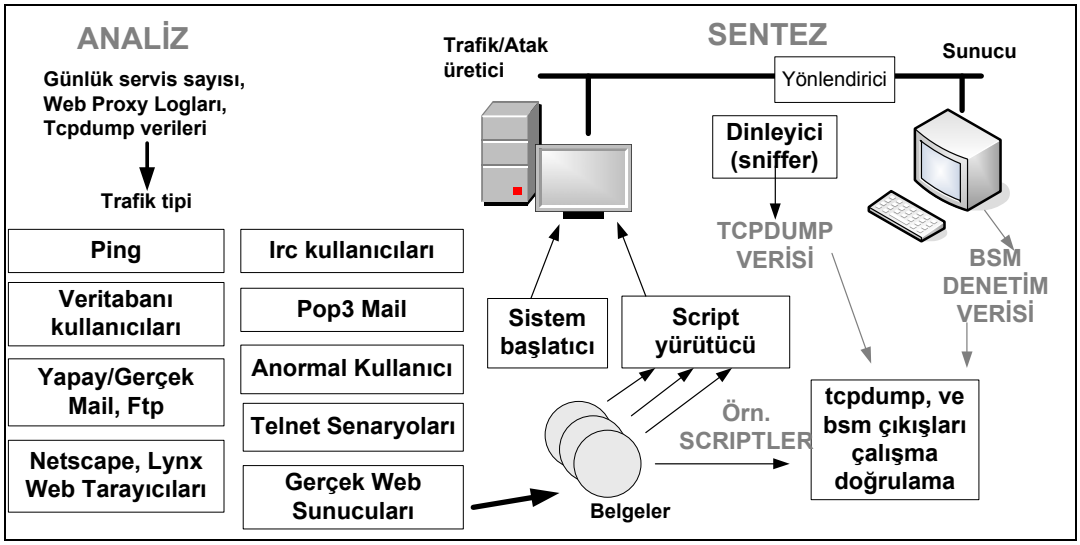


Şekil 5.3. Simülasyon ağının detaylı gösterimi

Simülasyon ortamındaki bilgisayar ve sunucuların isimleri ve IP adresleri EK-3'de verilmiştir.

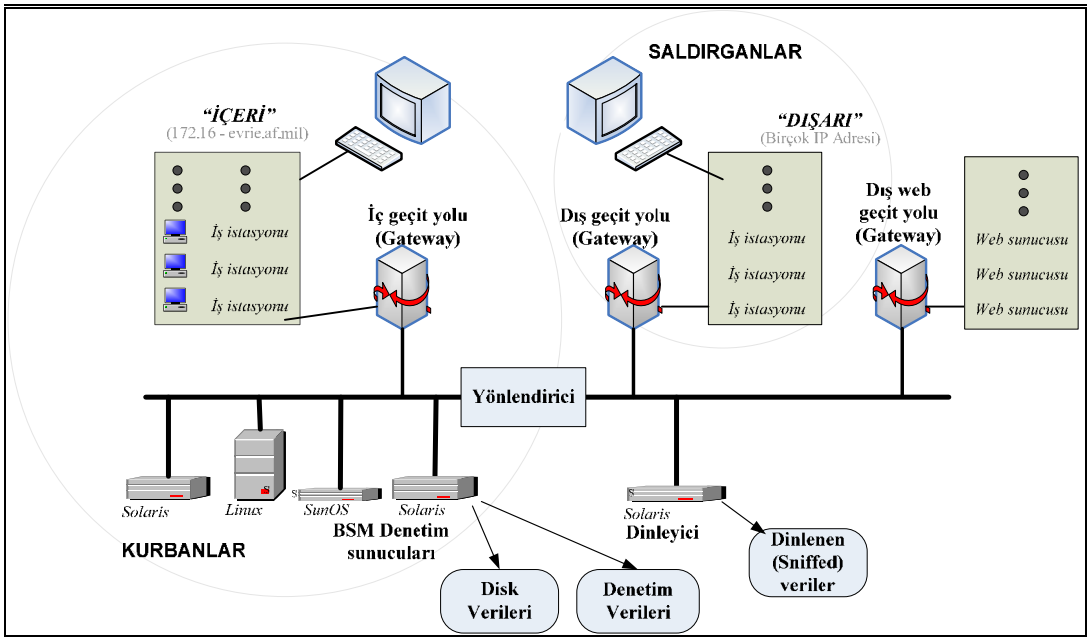
Simülasyon ortamı gösterilen bu veri kümesi için trafik üretme ve analiz etme işlemleri Şekil 5.4'de gösterilmiştir.

Şekil 5.4'de trafik üretmek için bir sistem başlatıcı tarafından tetiklenen trafik/atak üretici sistem, paketleri ve komutları üretmeye başlar. Sentez bölümünde, üretilen bu trafik, bir sniffer ile dinlenerek trafik verilerinin toplanması sağlanırken, bir Solaris sunucu ile de denetim verileri toplanır. Sistemin analiz kısmında, trafik üretici tarafından yürütülecek olan trafiğin tipine göre belge gönderilir.



Şekil 5.4. Trafik oluşturma ve analiz etme

Simülasyon ortamındaki saldırganlar ve bu saldırılara maruz kalan kurbanlar Şekil 5.5'de gösterilmiştir. Bu yapıda, saldırganların sadece dışarıdan birileri olabileceği varsayılarak tasarım yapılmıştır. İç ağdaki bilgisayarların tamamı kurban olarak kabul edilir.



Şekil 5.5. Simülasyonda saldırganlar ve kurbanlar

Eğitim veri kümesi elde edildikten sonra, 2 hafta boyunca üretilen verilerden test veri kümesi oluşturulmuştur. 1998 test veri kümesinde üretilen atak sayısı 38 tanedir. Bu ataklar Bölüm 2.5'de belirtilen saldırı tiplerine göre 4 ana gruba ayrılmıştır. Atakların hangi atak tipine ait olduğunun gösterildiği Çizelge 5.1'de, 38 ataktan 14 tanesi sadece test veri kümesinde bulunmaktadır. Çizelge 5.1'de sadece test veri kümesinde bulunan bu ataklar altı çizili olarak gösterilmiştir.

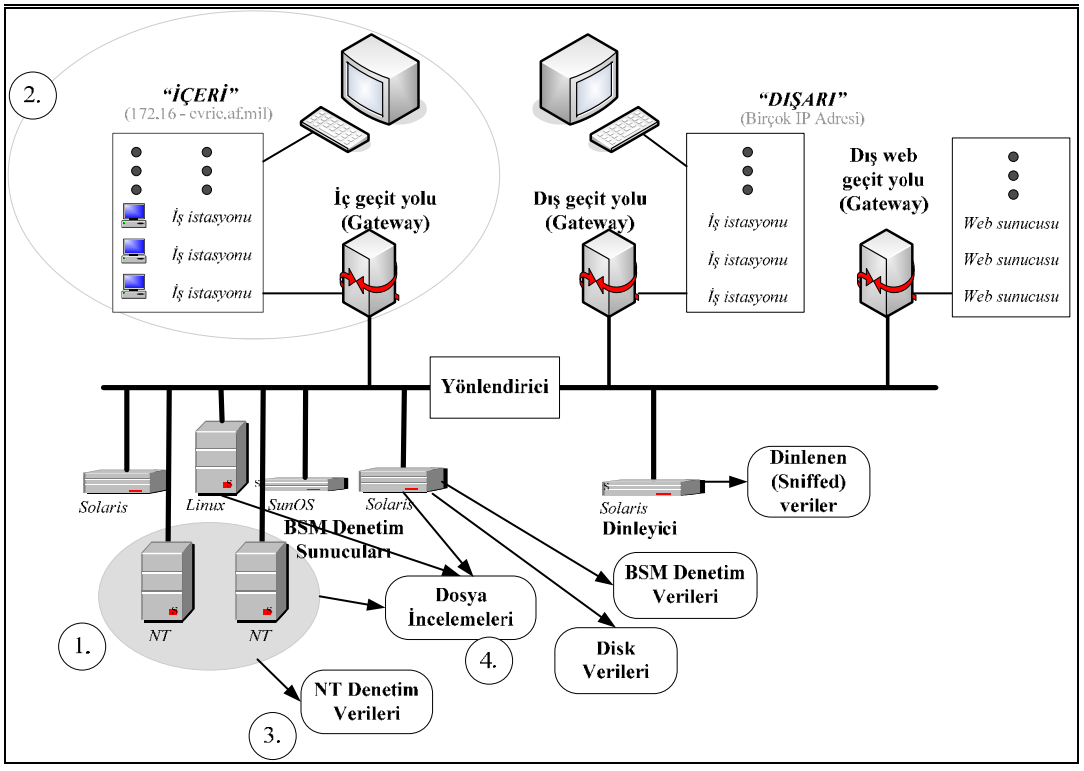
Çizelge 5.1. Test veri kümesinde yer alan ataklar

	Solaris Sunucu	SunOS	Linux	Cisco Yönlendirici
DoS	back neptune ping of death smurf syslog land apache2 mailbomb process table UDP storm	back neptune ping of death smurf land apache2 mailbomb process table UDP storm	back neptune ping of death smurf teardrop land apache2 mailbomb process table UDP storm	snmp getattack
R2U	Dictionary ftp-write guest phf ftp-write httptunnel xlock xsnoop	dictionary ftp-write guest phf httptunnel xlock xsnoop	dictionary ftp-write guest imap phf httptunnel named sendmail xlock xsnoop	
U2R	eject ffbconfig fdformat ps	loadmodule ps	perl xterm	
PROBE	ip sweep nmap port sweep satan mscan saint	ips weep nmap port sweep satan mscan saint	ip sweep nmap port sweep satan mscan saint	ip sweep nmap port sweep satan mscan saint

Çizelge 5.1'de gruplara ayrılmış olan atakların eğitim veri kümesindeki yerini gösteren liste EK-4'de verilmiştir. Anormalliklerin ve kullanıcı tanımlamalarının olduğu liste EK-5'de verilmiştir.

5.1.3. 1999 DARPA

DARPA 1999 veri kümesi, 1998'de geliştirilen veri kümesinin değiştirilmesi ile oluşturulmuştur. Yeni ve farklı bazı atakların da veri kümesine dahil olması için böyle bir değişikliğe ihtiyaç duyulmuştur. DARPA 1998 veri kümesinin geliştirilmesinde kullanılan ağ yapısına eklenen bu yeni özellikler Şekil 5.6'da gösterilmiştir [62].



Şekil 5.6. 1999 veri kümesinin ağ yapısı

Şekil 5.6'da gösterilen ağ yapısındaki önemli 4 yeni özellik şunlardır;

(1) NT iş istasyonları- NT trafiği ve atakları
(2) İç ataklar
(3) NT denetleme verileri incelemesi

(4) Dosya sistemi incelemesi

Bu değişikliklerin yanında bazı küçük değişiklikler de yapılmıştır. Bunlar;

(1) Eğitim verilerinde atağın hiç olmadığı günler sağlamak
(2) Eğitim ve test verilerinde daha az örtüşen ataklar
(3) Liste dosyası skorlama prosedürlerini tekrar gözden geçirme, olarak sıralanabilir

Geliştirilen DARPA veri kümelerinin asıl amacı olan, STS'lerin değerlendirilmesi işlemleri, simülasyon ağından toplanan ağ trafiği ve günlük kayıtları (audit data) kullanılarak, gerçek zamanlı olmayan değerlendirme aşamasında gerçekleştirilmiştir. Gerçek zamanlı test için STS'ler AFRL'ye gönderilmiştir. Bu sistemler AFRL ağ test yatağına eklenmiş ve gerçek zamanlı olarak, normal aktiviteler arasında atak oturumları belirlenmeye çalışılmıştır.

5.2. KDD'99

KDD'99 veri kümesi 1999 yılında DARPA veri kümesinin bazı önişlemlerden geçirilmesi ile elde edilmiş 41 özellikten oluşur. Bu veri kümesinin tasarlanmasının amacı, son yıllarda farklı tekniklerle gerçekleştirilmek istenen STS'ler için eğitim ve test işlemlerinde kolaylık sağlamaktır. Her ne kadar STS'ler için veri kümesi problemi DARPA ile çözülse de uygulanan bu yeni tekniklerde kullanabilmek için çok fazla önişlem gerektirir. KDD'99 veri kümesi ile eğitim ve test sonuçlarının daha hızlı alınabilmesi ise araştırmacılar için büyük bir avantajdır.

KDD'99 veri kümelerinde, 9 temel ve 32 adet türetilmiş olmak üzere toplamda 41 tane özellikten oluşan bir özellik haritası çıkarılmıştır. Bu 41 özellik 3 temel kategoriye ayrılarak ifade edilmiştir.

- İçerik özellikleri (content features)
- Sunucu tabanlı trafik özellikleri (host-based traffic features)
- Zamana bağlı trafik özellikleri (time-based traffic features)

Çizelge 5.2, Çizelge 5.3 ve Çizelge 5.4'de, sırasıyla bu 3 kategori ve kategoriler içerisindeki veri özellikleri gösterilmiştir.

Çizelge 5.2. İçerik özellikleri

Özellik adı	**Tanım**	**Tip**
duration	Bağlantı uzunluğu	sürekli
protocol_type	Protokol tipi	ayrık
service	Servis tipi	ayrık
src_bytes	Kaynaktan hedefe veri	sürekli
dst_bytes	Veri byte sayısı	sürekli
flag	Bayrak	ayrık
land	Kaynak ve hedef IP aynı ise 1 değilse 0	ayrık
Wrong_fragment	Yanlış parçalama	sürekli
urgent	Acil paket sayısı	sürekli

İçerik özellikleri, sadece TCP bağlantılarından alınan temel özelliklerdir. Bu özelikleri elde etmek, ağ trafiği verileri üzerinde önişlem yapılması gerekmediğinden, diğer kategorilere göre daha kolaydır.

Çizelge 5.3. Sunucu tabanlı trafik özellikleri

Özellik adı	**Tanım**	**Tip**
hot	"hot" göstergesi	sürekli
num_failed_logins	Hatalı giriş sayısı	sürekli
Logged_in	Giriş başarılı ise 1 değilse 0	ayrık
num_compromised	Gizliliğin ihlal edilme sayısı	sürekli
root_shell	"Root Shell" elde edildiyse 1 değilse 0	ayrık
su_attempted	"Su Root" komutu girildiyse 1 değilse 0	ayrık
num_root	"Root" erişim sayısı	sürekli
num_file_creations	Dosya oluşturma işlemleri sayısı	sürekli
num_shells	Shell promptlarının sayısı	sürekli
num_access_files	Kontrol dosyalarına erişim işlemleri sayısı	sürekli
num_outbound_cmds	ftp oturumunda giden komut sayısı	sürekli
is_hot_login	Giriş "hot" listesindeyse 1 değilse 0	ayrık
is_guest_login	Giriş "guest" ise 1 değilse 0	ayrık

Sunucu tabanlı trafik özellikleri, etki alanı (domain) bilgisi ile ortaya çıkan bağlantı içerik özellikleridir.

Zamana bağlı trafik özellikleri, "aynı sunucu" ve "aynı servis" özellikleri kullanılarak çıkarılan özelliklere verilen isimdir. "Aynı sunucu" özellikleri, son iki saniye içerisinde aynı sunucuya yapılan bağlantıların gözden geçirilmesi ile elde edilir. Benzer olarak "aynı servis" özellikleri son iki saniye içerisinde aynı servise yapılan bağlantıların gözden geçirilmesi ile elde edilir.

Çizelge 5.4. Zamana bağlı trafik özellikleri

Özellik adı	**Tanım**	**Tip**
count	Aynı sunucuya önceki iki bağlantıyla aynı bağlantıların sayısı	sürekli
serror_rate	"SYN" hata bağlantılarının yüzdesi	sürekli
rerror_rate	"REJ" hata bağlantılarının yüzdesi	sürekli
same_srv_rate	Aynı servise bağlantıların yüzdesi	sürekli
diff_srv_rate	Farklı servislere bağlantıların yüzdesi	sürekli
srv_count	Aynı servise önceki iki bağlantıyla aynı bağlantıların sayısı	sürekli
srv_serror_rate	"SYN" hata bağlantılarının yüzdesi	sürekli
srv_rerror_rate	"REJ" hata bağlantılarının yüzdesi	sürekli
srv_diff_host_rate	Farklı servislere bağlantıların yüzdesi	sürekli

DARPA veri kümelerinin, belirli önişlemlerden geçirilmesi ile oluşturulan KDD'99 veri kümesi toplamda 38 farklı atak içerir. Bunlardan 24 tanesi eğitim veri kümesinde iken, test veri kümesi, eğitim veri kümesinde bulunmayan 14 farklı atak tipini daha içerir.

KDD'99 eğitim veri kümesinde bulunan 24 atak ve bu atakların ait oldukları saldırı tipleri ve saldırıları veri kümesinde bulunan örnek sayıları Çizelge 5.5'de gösterilmiştir.

Sadece test veri kümesinde yer alan ve etiketlenmiş KDD'99 dosyasından alınan 14 farklı atağa ait saldırı tipi ve örnek sayıları Çizelge 5.6'da gösterilmiştir [65].

Çizelge 5.5. KDD'99 veri kümesinin %10'luk kısmından alınan saldırı örneklerinin sayıları

Atak	Örnek sayısı	Kategori
smurf.	280790	dos
neptune.	107201	dos
back.	2203	dos
teardrop.	979	dos
pod.	264	dos
land.	21	dos
normal.	97277	normal
satan.	1589	probe
ipsweep.	1247	probe
portsweep.	1040	probe
nmap.	231	probe
warezclient.	1020	r2l
guess_passwd.	53	r2l
warezmaster.	20	r2l
imap.	12	r2l
ftp_write.	8	r2l
multihop.	7	r2l
phf.	4	r2l
spy	2	r2l
buffer_overflow.	30	u2r
rootkit.	10	u2r
loadmodule.	9	u2r
perl.	3	u2r

Çizelge 5.6. Eğitim kümesinde yer almayan ataklar

Atak	Örnek sayısı	Kategori
apache	794	dos
mailbomb	5000	dos
processtable	759	dos
udpstorm	2	dos
mscan	1053	probe
saint	736	probe
httptunnel.	138	r2l
named	17	r2l
sendmail	17	r2l
snmpgetattack	1040	r2l
xlock	9	r2l
xsnoop	4	r2l
ps	16	u2r
xterm	13	u2r

6. ZEKİ STS VE VERİ KÜMESİ TASARIMI

Bu tez çalışmasında, uygulaması gerçekleştirilen YSA tabanlı zeki STS'nin geliştirilmesindeki temel adımlar ve uygulanan aşamalar önceki bölümlerde sunulmuştur. Bu bölümde ise daha başarılı sonuçlara ulaşabilmek için farklı zeki bir STS önerisi ile birlikte büyük eksikliğini gördüğümüz "ulusal saldırı veritabanı" geliştirmek için önerilerde bulunulmuş ve yapılması gerekenler ve atılması gereken adımlar bir plana göre aşağıda alt başlıklarda açıklanmıştır.

6.1. Planlama

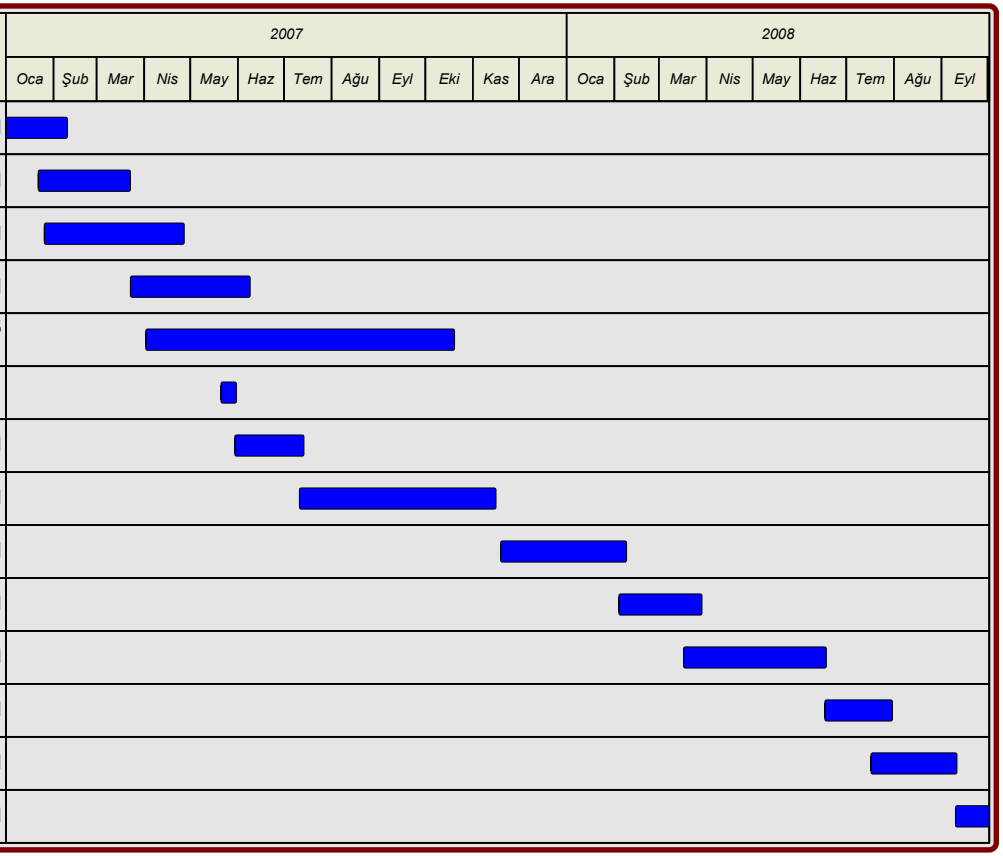
Zeki STS'lerin eğitilmesinde ve test edilmesinde kullanılabilecek, güncel ve yeterli örneğe sahip bir STS tasarımı yapabilmek için literatürde yapılan benzer çalışmalar incelenmiştir. Bunlardan, DARPA ve KDD'99 veri kümeleri Bölüm 5'te açıklandığını bir kez daha hatırlatmakta fayda görüyoruz. Yapılan araştırmalar ve DARPA veri kümelerinin incelenmesinden edinilen bilgiler ışığında, "ulusal saldırı veri kümelerinin" oluşturulması için gereken öneriler yapılmıştır. Veri kümesi oluşturulmasına yönelik örnek bir planlama tablosu Çizelge 6.1'de sunulmuştur. Bu çalışma kapsamında, bu plana göre tezin aşamaları oluşturulmak istenilse de bu tez kapsamına alınmayan birçok aşama bulunmaktadır.

6.2. Eğitilecek Saldırı Tiplerinin Belirlenmesi

Geliştirilecek olan saldırı veri kümesi içerisinde, öncelikle hangi saldırıların yer alacağına karar verilmelidir. Tercih edilen durum, en güncel saldırıları içinde barındıran bir veri kümesidir. Ancak sistemin yapısına göre, daha önceden elde edilen istatistiksel sonuç verileri sayesinde, en çok karşılaşılan saldırıların tespit edilmesi yararlı olacaktır. Veri kümesi içerisinde yer alacak olan saldırı örneklerinin miktarı, yine aynı istatistiksel verilerle orantılı olarak tasarlanabilir. Örneğin bir sistemde, en çok DoS saldırıları ile karşılaşıldığı tespit edilmişse, veri kümesinin içeriğinde DoS saldırılarının yoğun olması beklenir.

Çizelge 6.1. Veri kümesi oluşturma, görev çizelgesi

No.	Görev	Başlangıç	Bitiş	Gün
1	Saldırıların istatistiksel araştırması	01.01.2007	09.02.2007	30d
2	Saldırı araçlarının araştırılması	22.01.2007	22.03.2007	44d
3	Ağ paketlerini dinlemek için geliştirilen yazılımların araştırılması	26.01.2007	26.04.2007	65d
4	Ağ dinleme aracı geliştirme	23.03.2007	08.06.2007	56d
5	Veritabanının oluşturulacağı ağı kurma saldırgan ve kurban olan makineleri belirleme	02.04.2007	19.10.2007	145 d
6	Örnek verilerin toplanması	21.05.2007	30.05.2007	8d
7	Verilerin incelenmesi	30.05.2007	13.07.2007	33d
8	Gerekli iyileştirmelerin yapılması	11.07.2007	15.11.2007	92d
9	Eğitim verilerinin toplanması	19.11.2007	08.02.2008	60d
10	Test verilerinin toplanması	04.02.2008	28.03.2008	40d
11	Özellik vektörünün çıkarılması	17.03.2008	17.06.2008	67d
12	Eğitim ve test için kullanılır hale getirilmesi	17.06.2008	30.07.2008	32d
13	Sonuçların değerlendirilmesi	17.07.2008	10.09.2008	40d
14	Veri kümelerinin oluşturulma aşamaları ve içeriğiyle araştırmacılara sunulması	10.09.2008	03.10.2008	18d



Güncel verilerin elde edilmesi için bu konuda çalışan laboratuar ve enstitülerin araştırma sonuçlarından faydalanılabilir. MITRE'nin 2000-2007 yıllarını kapsayan raporunda bilinen ve en çok karşılaşılan açıklar sunulmuştur [18]. Bu rapordan elde edilen genel sonuç Çizelge 6.2.'de gösterilmiştir. Çizelgede verilen rakamlar, zayıflıkların ait olduğu yılda görülme oranına göre sıralanmıştır.

Çizelge 6.2. 2000-2007 yıllarındaki ilk 10 zayıflık

	2001	**2002**	**2003**	**2004**	**2005**	**2006**
xxs	-	2	2	2	1	1
buf	1	1	1	1	3	4
sql-inject	-	-	4	3	2	2
php-include	-	-	-	10	6	3
dot	2	4	5	4	4	5
infoleak	9	5	6	5	5	6
dos-malform	3	3	8	6	8	7
link	4	9	3	7	7	-
format-string	7	10	7	8	9	-
crypt	5	6	9	-	10	-
priv	10	8	-	-	-	-
perm	8	-	-	-	-	9
metachar	6	7	-	-	-	-
int-overflow	-	-	10	9	-	8
webroot	-	-	-	-	-	10

6.3. Veri Kümesi Oluşturmak İçin Kullanılan Saldırı Programları

Saldırı veri kümesi oluşturmak için kullanılan programlar, saldırı araçları olarak da adlandırılabilir. Bu araçların kullanıldıkları amaca göre, bilgi ve bilgisayar güvenliği için yararlı ya da zararlı oldukları değişir. Örneğin Nmap, sistem yöneticileri tarafından, yönettikleri sistemin açıklarını taramak ve belirlenen açıkları kapatmak için kullanılabildiği gibi, kötü amaçlı kişiler tarafından, sızmak istedikleri sistemin açıklarını öğrenmek için kullanılabilir.

Örnek veri kümesi saldırılarını oluşturmak için, kullanılabilecek araçlardan biri olan Hping'in asıl görevi, TCP/IP paketleri üretmek ve analiz etmektir [66]. Ancak

paketler üretilirken paket başlığında istenilen alanlar değiştirilebildiği için saldırı amaçlı kullanılmaya uygundur. Bunu bir örnek vererek aşağıdaki gibi gösterebiliriz,

#hping -a 10.11.27.156 10.11.27.156 -S -p 22

Komutu ile hedef ve kaynak ip adresleri aynı yazılmıştır. Komut satırındaki "–S" komutu SYN paketi yollanacağını, "-p" komutu ise hangi porta gönderildiğini göstermektedir. Yapılan bu atak ile sistem kendi kendine paket gönderiyormuş gibi davrandığından hizmet veremez hale gelebilir.

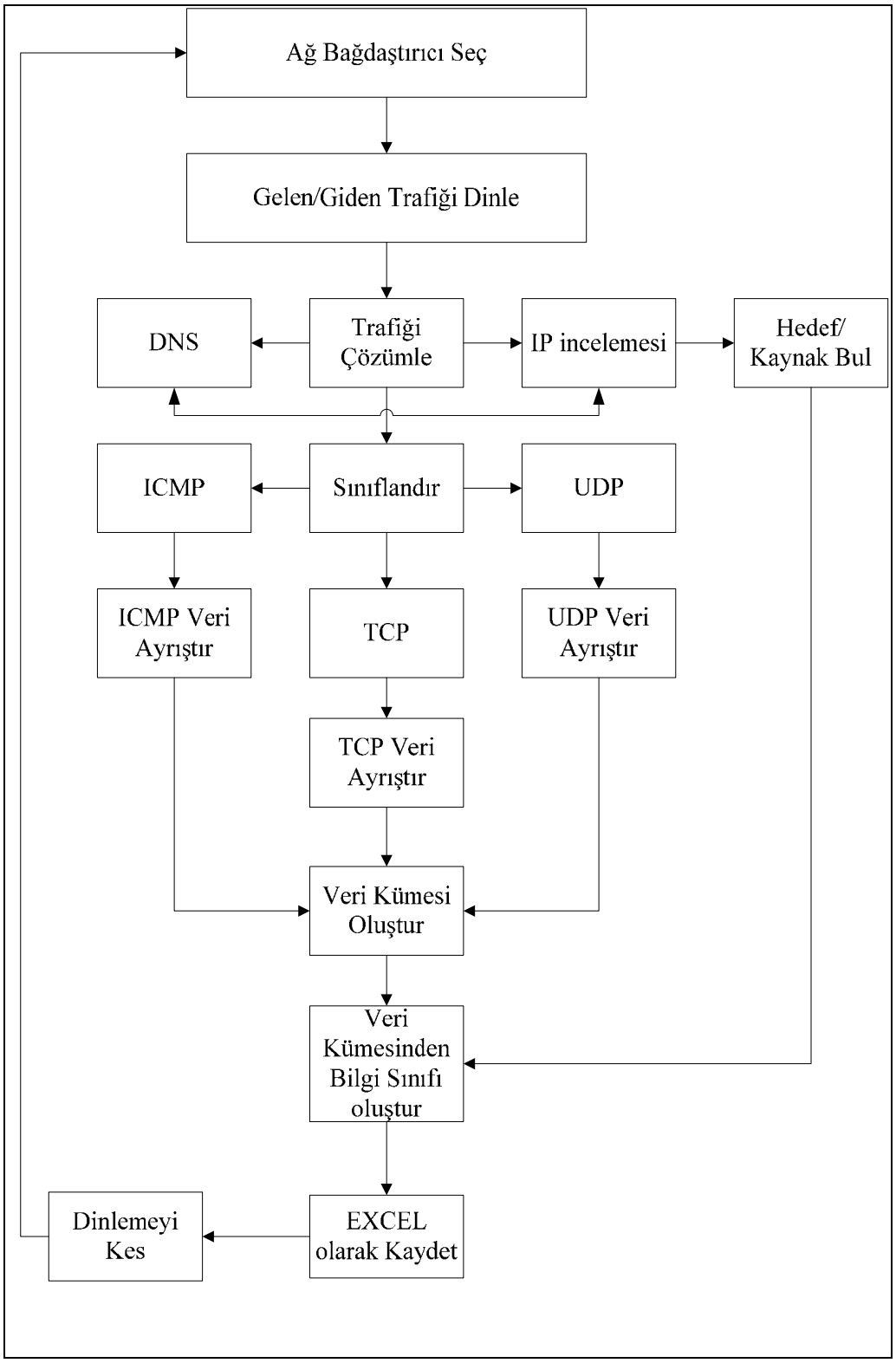
Hping'in günümüzde Windows işletim sistemlerinde de çalışması, kullanıcı sayısının artmasını sağlamıştır.

6.4. Paket Trafiğini Dinleme ve Kaydetme

Paket trafiğini dinlemek için geliştirilmiş birçok yazılım vardır. Bunlardan en çok kullanılanlarından biri, DARPA'nın da kullandığı Tcpdump'dır. Linux için tasarlanmış ve Libpcap kütüphanesini kullanan Tcpdump, günümüzde Windump olarak Windows sistemlerde de kullanılabilir duruma getirilmiştir. Tcpdump, belirlenen filtre veya kurallara göre, ağdaki paketlerin başlık bilgilerini alır [67]. Daha görsel bir uygulama olan Ethereal'da yine sistem güvenlik uzmanlarının sıkça kullandığı başka bir dinleyici programdır. Ayrıca açık kaynak kodlu bir STS yazılımı olan Snort da ağ dinleyicisi modunda çalışabilen bir uygulamadır [68].

Bu tez çalışmasında öneri olarak sunulan ulusal saldırı veritabanının oluşturulması için bir sniffer uygulaması gerçekleştirilmiştir. Bunun nedeni, bu veri kümesi oluşturulduktan sonra, tasarımı yapılan zeki STS'ye modül olarak eklenmesi ve trafiği dinleme özelliğine sahip olarak daha sonra yapılacak olan önişlemler için kolaylık sağlanmasının istenmesidir. Diğer bir sebep ise, tcpdump'dan alınan çok sayıda verinin, filtrelenmiş dahi olsa standart formatta okumanın ve önişlem yapmanın zorluğu. TCP, UDP ve ICMP protokollerini dinleyen bu sniffer programında, toplanan veriler filtrelenerek, her biri ayrı ayrı tutulabildiği gibi, hepsi birlikte de tutulabilir.

Geliştirilen dinleyici (sniffer) programının akış diyagramı Şekil 6.1'de gösterilmiştir.



Şekil 6.1. Geliştirilen dinleyici (sniffer) programının akış diyagramı

TCP protokolündeki paketlerin, bit bit nasıl ayrıştırıldığı Çizelge 6.3'de gösterilmiştir.

Çizelge 6.3. Ağ paketlerinin, paket başlıklarına ayrıştırılması

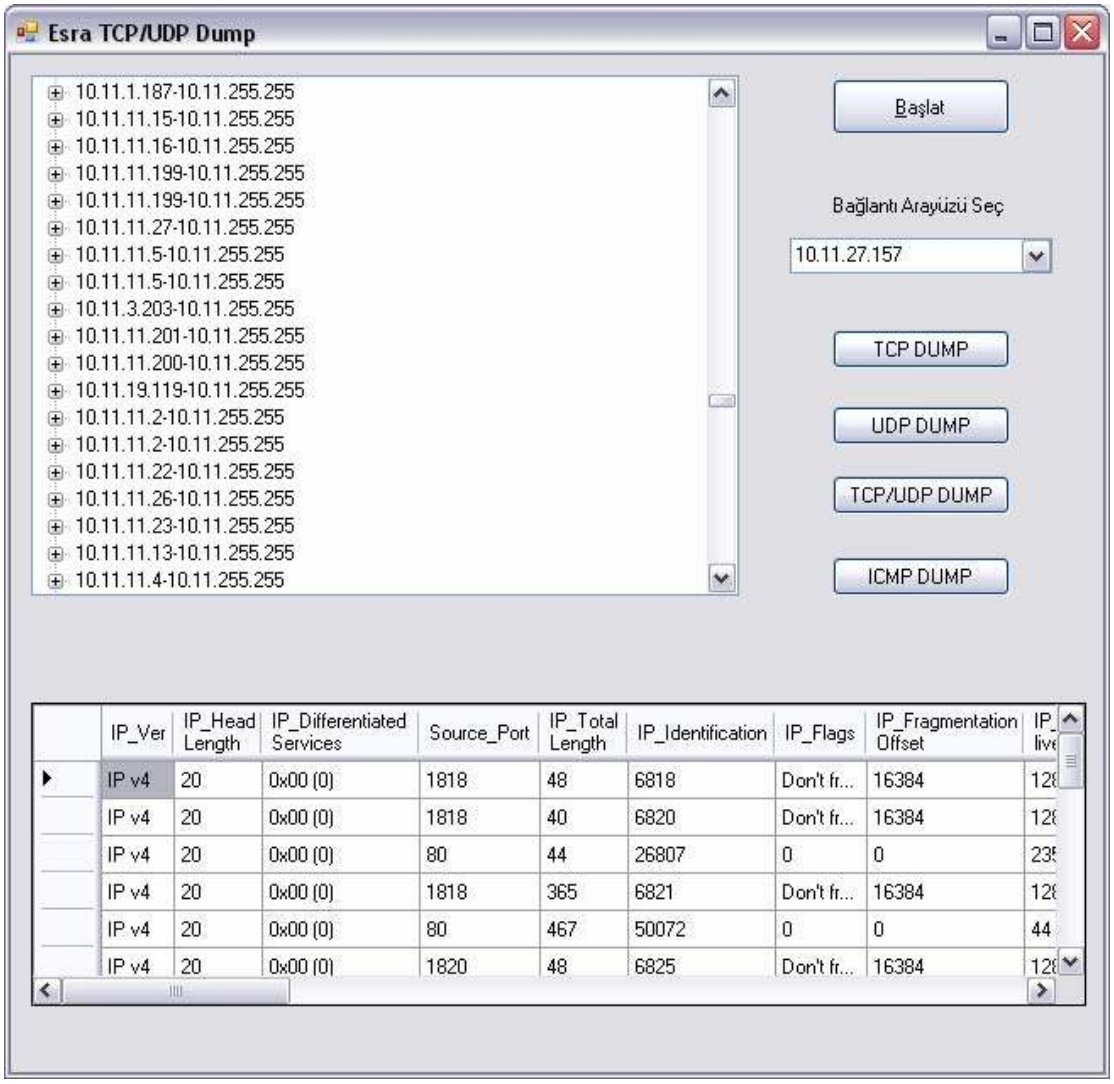
```
//ilk 16 bit kaynak port
usSourcePort=(ushort)IPAddress.NetworkToHostOrder(binaryReader.ReadInt16
());
//ikinci 16 bit hedef port
usDestinationPort=(ushort)IPAddress.NetworkToHostOrder(binaryReader.ReadInt
16 ());
//32 bit sequence number
uiSequenceNumber=(uint)IPAddress.NetworkToHostOrder(binaryReader.ReadInt
32());
//32 bit acknowledgement number
uiAcknowledgementNumber=(uint)IPAddress.NetworkToHostOrder(binaryReade
r.ReadInt32());
//16 bit hold the flags and the data offset
usDataOffsetAndFlags=(ushort)IPAddress.NetworkToHostOrder(binaryReader.Re
adInt16());
//16 bit window size
usWindow=(ushort)IPAddress.NetworkToHostOrder(binaryReader.ReadInt16());
//16 bit checksum
sChecksum=(short)IPAddress.NetworkToHostOrder(binaryReader.ReadInt16());
//16 bit urgent pointer
usUrgentPointer=(ushort)IPAddress.NetworkToHostOrder(binaryReader.ReadInt1
6());
//data offset verinin nerde başladığını gösterdiğinden, başlık uzunluğunu
hesaplamak için kullanıldı
byHeaderLength = (byte)(usDataOffsetAndFlags >> 12);
byHeaderLength *= 4;

//Mesaj uzunluğu = TCP paketinin toplam uzunluğu – Başlık uzunluğu
usMessageLength = (ushort)(nReceived - byHeaderLength);
```

TCP protokolünde olduğu gibi, diğer protokoller için de aynı şekilde veriyi bit bit parçalama ve kaydetme işi yapılır.

Tcpdump kullanılmamasının sebepleri;

- Tcpdump'dan alınan veriler filtrelenmiş dahi olsa, standart formatta okumanın ve önişlem yapmanın zorluğu ve

- Kendi verilerimizi yönetmek ve ayrı ayrı kaydetme imkanına sahip olmak olarak sıralanabilir.



Şekil 6.2. Geliştirilen dinleyici (sniffer) programı arayüzü

Şekil 6.2'de ekran çıktısı gösterilen dinleyici (sniffer) programı, kullanıldığı makineye gelen ve giden paketleri dinler. Programı kullanmak için öncelikle makinenin IP numarası yazıldıktan sonra, dinlemeye başlamak için "Başlat" butonuna tıklanır. Bu sırada sol taraftaki ekranda, protokol ayırt etmeksizin gelen giden tüm paketler listelenmeye başlar. "Durdur" butonuna tıklayana kadar bu listeleme işlemi ve aynı zamanda dinleme işlemi devam eder. Program durdurulduktan sonra, listelenmek istenen protokole göre sağ taraftaki butonlar

kullanılır. Listeleme işlemi ile aynı zamanda protokol ismi ile gruplandırılmış Excel dosyalarına veriler aktarılır.

Bu program, veri paketlerinin belli zaman aralıklarında toplanması için geliştirilmiştir. Örneğin her gün saat 10:00 ile 14:00 arasında veriler toplanarak depolanabilir.

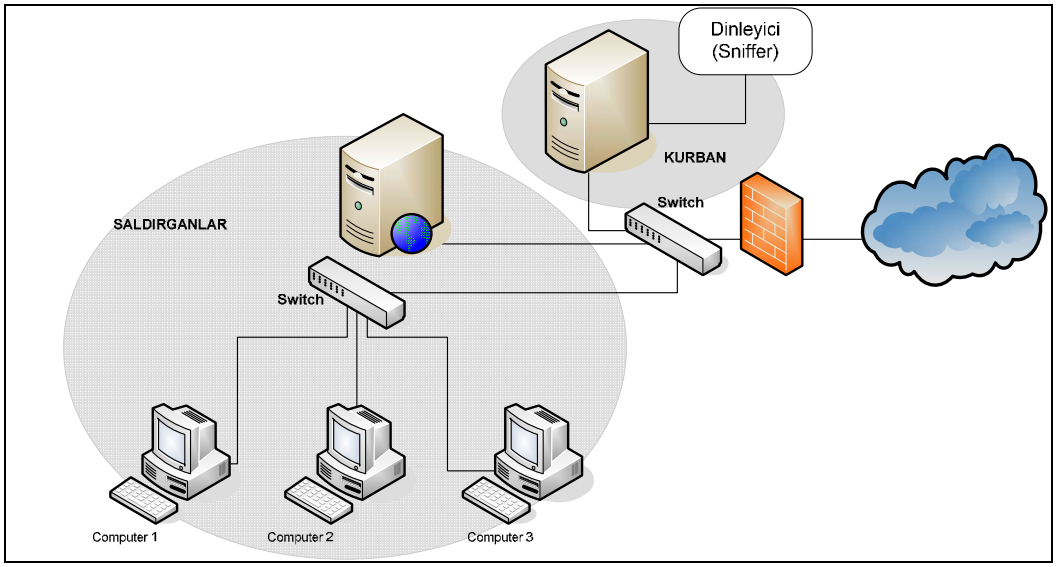
Toplanan verilerin analizi ve saldırı tipi ve zamanı bilgileri, aynı zaman aralığında çalıştırılan bir açık kaynak kodlu, bir de ticari STS yazılımı ile karşılaştırılarak bulunur. Daha sonra bir güvenlik uzmanı tarafından incelenerek, bu yazılımların tanıyamadığı saldırılarında etiketlenmesi işlemi gerçekleştirilir.

Elde edilen etiketlenmiş verinin, standart ve tek satırda ifade edilebilir hale gelmesi için gerekli önişlemler yapılır. Bunun sebebi, saldırıların genelde tek satırda anlaşılamaması ve YSA gibi tekniklerin kullanılması sırasında karşılaşılabilecek güçlükleri en aza indirmektir.

6.5. Ağ Tasarımı

Çizelge 6.3'de sunulan çalışma planı kapsamında, 5. görev olan "ağ yapısının oluşturulması" için literatürdeki çalışmalar incelenerek gerekli donanım ve yazılım gereksinimleri sağlanmalıdır. Bu aşamada, saldırı yapacak makinelerin ve kurban olan makinenin belirlenmesi gerekir.

Bu tez çalışmasında edinilen bilgi ve deneyimlerle oluşturulmaya başlanan yapı ve sağlanan donanımlar Şekil 6.3'de gösterilmiştir. Bu yapı sayesine örnek veritabanının elde edilmesi için kullanılabilecek bir platform oluşturulmuştur. Örnek veritabanı elde edildikten sonra, görülebilecek eksiklikler neticesinde, ağ yapısının iyileştirilmesi, saldırı araçlarının çeşitlendirilmesi, ağ dinleyici programının tekrar gözden geçirilmesi gibi değişiklikler yapılmalıdır.



Şekil 6.3. Saldırı veri kümesi ağ simülasyonu

6.6. Saldırı Veri Kümesini Özellik Vektörüne Dönüştürme

Ağ paketlerinden oluşturulan veri kümelerinin, önişlem yapılmadan kullanılması, bilinen bazı saldırılar haricinde, olası saldırı hakkında çok fazla bilgi vermez. Tek bir pakette anlaşılabilen saldırılardan en çok bilinenlerden biri, DoS tipinde olan Land saldırısıdır. Bu saldırıda, kaynak ve hedef IP adresleri aynı olduğundan kolayca ayırt edilebilir. Ancak birçok saldırının tespit edilebilmesi için, o paketten önceki belli bir zaman aralığında analiz yapılması gerekir.

Belirlenecek olan zaman aralığında, bazı kriterler dikkate alınarak ve literatürdeki örnekleri incelenerek elde edilecek kurallar doğrultusunda bir özellik vektörü oluşturmak zeki sistemlerde bu saldırı veritabanının kullanılabilirliğini sağlamak açısından önemlidir.

Özellik vektörü oluşturulurken, bir ağ paketinde saldırıyı tanımlamada önemli olan bilgilerin dikkate alınması gerekir. Literatürde "özellik seçme (feature selection)" olarak isimlendirilen bu işlemi gerçekleştirebilmek için, performans ve kullanılabilirlik açısından önemli bilgiler veren metriklerin bilinerek, buna göre özelliklerin önem dereceleri belirlenmelidir.

6.7. Eğitim ve Test Veri Kümeleri Oluşturma

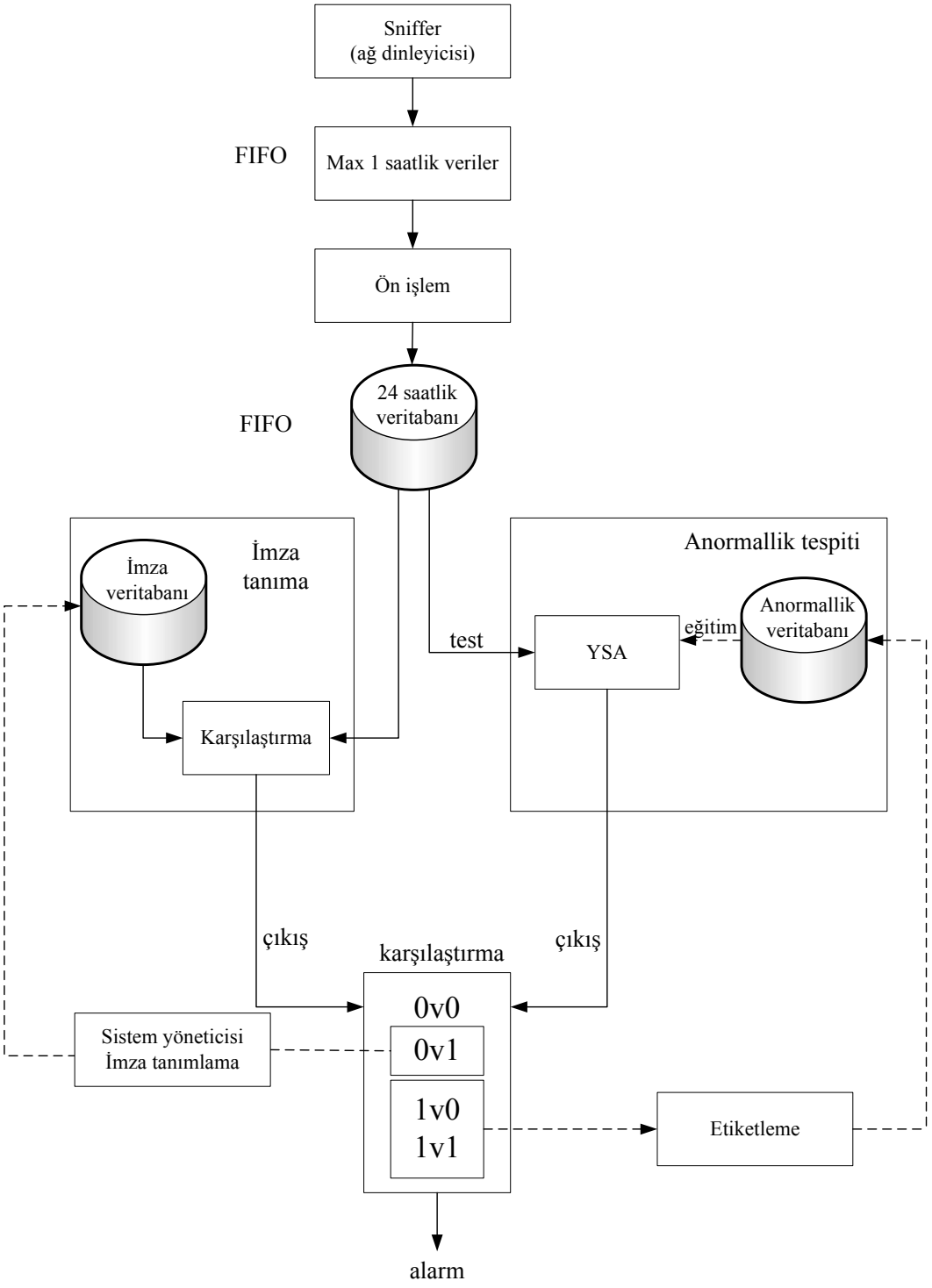
Oluşturulan saldırı veri kümesinin, yapılan ön işlem ile özellik vektörü elde edildikten sonra, bu vektörün değişkenlerine göre, tüm saldırı veritabanı, eğitim ve test için kullanılabilir hale getirilmelidir.

6.8. Zeki STS Tasarımı

Bu tez çalışması esnasında edinilen bilgi birikimi ve deneyimi göz önüne alınarak Şekil 6.4'de sunulan yapı verilmiştir.

Bu tasarım ile sunucu tabanlı, hibrit (anormallik ve kötüye kullanım) saldırı tespit yaklaşımı kullanılarak yanlış alarm oranının düşürülmesi amaçlanmaktadır.

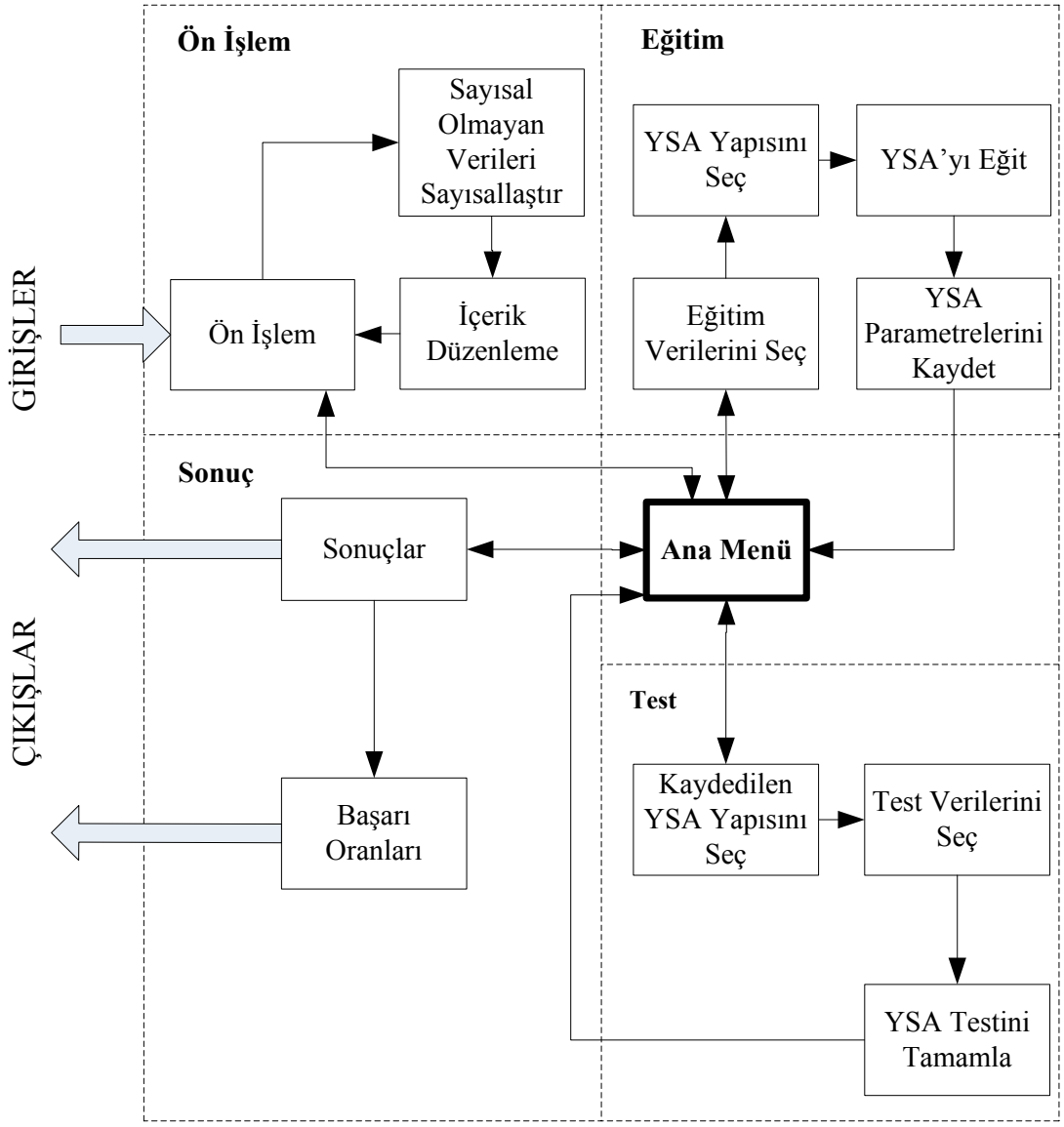
Sunucuya gelen ağ paketleri geliştirilen sniffer modülü ile alınarak ayrıştırılır. Ayrıştırılan bu veriler saatlik veri depolayabilen bir hafızaya atılır. Özellik vektörü oluşturmak için, kaydedilen her yeni veri kendisinden 5 saniye önceki veriye kadar geriye dönük işlem yapılarak 24 saatlik veri tutan veritabanına aktarılır. Elde edilen özellik vektörü aynı anda hem anormallik tespiti yapan modüle hem de imza tanıma tespiti yapan modüle gönderilir. Her iki modülün sonuçları karşılaştırıldığında saldırıya rastlanmadığında veri normaldir. Her iki modülden biri ya da ikisi saldırı sonucu döndürdüğünde saldırı olduğu tespit edilir. Eğer modüllerden biri tespit ettiği halde diğeri etmemişse, bu durum tespit edemeyen modülün güncelleştirilmesi için kullanılır.



Şekil 6.4. Önerilen zeki STS tasarımı

7. ZEKİ STS YAZILIMI GELİŞTİRME

Bu tez kapsamında, Bölüm 5.2'de incelediğimiz KDD'99 verileri ile çalışılarak, zeki bir STS uygulaması gerçekleştirilmiştir. Yapay sinir ağları kullanılarak gerçekleştirilen bu zeki STS yazılımının blok şeması Şekil 7.1'de verilmiştir. Blok şemadan da görülebileceği gibi geliştirilen bu uygulama yazılımı, ön işlem, eğitim, test, sonuç-başarı oranı ve ana menü olmak üzere beş modülden oluşmaktadır. Bu modüller kesikli çizgiler ile birbirinden ayrılmıştır.



Şekil 7.1. Geliştirilen zeki STS yazılımının blok şeması

Bu tez kapsamında gerçekleştirilen, zeki STS tasarımı için yapılan işlemler bu bölümde anlatılmıştır. Öncelikle eğitim ve test için kullanılan veri kümeleri hakkında detaylı bilgi verilmiştir. Zeki STS'nin en önemli kısmını oluşturan YSA yapısı açıklanmış ve elde edilen en iyi YSA modelinin test sonuçları ise bu bölümün sonunda sunulmuştur.

7.1. Eğitim Veri Kümeleri

Uygulama sırasında kullanılan eğitim veri kümeleri 3 farklı grup olarak tasarlanmış ve her biri KDD'99 veri kümesinin bir alt kümesi olan %10'luk veri kümesinden rasgele alınmış 65536 adet örnekten oluşturulmuştur. Bu eğitim kümeleri ile üç farklı eğitim yapılmıştır. İlk eğitim kümesinde yer alan veriler ve örnek sayıları Çizelge 7.1'de gösterilmiştir.

Çizelge 7.1. Eğitim kümesi #1

Saldırı tipi	**Saldırı adı**	**Veri sayısı**	**Saldırı tipine göre toplam**	**Tüm saldırılar toplam**	**Toplam eğitim verisi sayısı**
DoS	Land	16	42681	44098	65536
	Neptune	29086			
	Pod	20			
	Smurf	13459			
	Teardrop	100			
U2R	Buffer-overflow	2	4		
	Loadmodule	1			
	Perl	1			
R2L	Imap	11	34		
	Multihop	1			
	Warezmaster	20			
	Phf	2			
Probe	Ipsweep	102	1379		
	Nmap	101			
	Portsweep	238			
	Satan	938			
Normal	Normal	21438	21438	21438	

Oluşturulan eğitim kümesi incelendiğinde DoS saldırı tipinde 42681 veri varken, U2R saldırı tipinde sadece 4 veri olduğu görülür. Bu durum, 4 farklı saldırı tipini de

içeren eğitim verilerinden oluşmasına rağmen, veri kümesinin tamamının doğasında olan orantısız dağılımdan kaynaklanmaktadır. Diğer iki eğitim veri kümeleri de sırasıyla Çizelge 7.2 ve Çizelge 7.3'de gösterilmiştir.

Çizelge 7.2. Eğitim kümesi #2

Saldırı tipi	**Saldırı adı**	**Veri sayısı**	**Saldırı tipine göre toplam**	**Tüm saldırılar toplam**	**Toplam eğitim verisi sayısı**
DoS	Back	101	52493	55046	65536
	Neptune	80			
	Pod	62			
	Smurf	52052			
	Teardrop	198			
U2R	Buffer-overflow	7	20		
	Loadmodule	6			
	Rootkit	7			
R2L	Multihop	1	1023		
	Spy	2			
	Warezclient	1020			
Probe	Ipsweep	102	1510		
	Portsweep	361			
	Satan	1047			
Normal	Normal	10490	10490	10490	

Çizelge 7.3. Eğitim kümesi #3

Saldırı tipi	**Saldırı adı**	**Veri sayısı**	**Saldırı tipine göre toplam**	**Tüm saldırılar toplam**	**Toplam eğitim verisi sayısı**
DoS	Land	1	58267	58751	65536
	Neptune	41522			
	Pod	140			
	Smurf	16122			
	Teardrop	482			
U2R	Buffer-overflow	5	6		
	Perl	1			
R2L	-	-	-		
Probe	Ipsweep	257	478		
	Portsweep	220			
	Satan	1			
Normal	Normal	6785	6785	6785	

7.2. Test Veri Kümeleri

Farklı eğitim veri kümeleriyle eğitilen üç YSA yapısı, Bölüm 7.1'de sunulan eğitim veri kümelerinden farklı olan beş farklı test veri kümesi ile test edilmiştir. Test kümeleri, KDD'99 doğrulanmış test veri kümelerinden alınan 65536 örnekli verilerden oluşmaktadır. Bu çalışmada örnek olarak kullanılan ve başarı oranları gösterilen beş farklı test veri kümesi sırasıyla Çizelge 7.4, Çizelge 7.5, Çizelge 7.6, Çizelge 7.7 ve Çizelge 7.8'te gösterilmiştir.

Çizelge 7.4. Test kümesi #1

Saldırı tipi	**Saldırı adı**	**Veri sayısı**	**Saldırı tipine göre toplam**	**Tüm saldırılar toplam**	**Toplam test verisi sayısı**
DoS	Pod	22	57256	59168	65536
	Smurf	56835			
	Apache2	398			
	Udpstorm	1			
U2R	Buffer_overflow	3	3		
R2L	Guess_passwd	2	1100		
	Multihop	4			
	Phf	1			
	Named	10			
	Sendmail	6			
	Snmpgetattack	1069			
	Xlock	6			
	Xsnoop	2			
Probe	Ipsweep	80	809		
	Portsweep	106			
	Saint	623			
Normal	Normal	6367	6368	6368	

Eğitim ve test veri kümelerinde bulunan örnek sayıları incelendiğinde, birbirine yakın olmayan değerler gözlenmiştir. Örneğin bir veri kümesinde bir saldırı türünden 1000 örnek varken diğer veri kümesinde 5 adet vardır. Bu durum, test veri kümesi için önemsizken, YSA'nın öğrenmesini etkileyeceğinden eğitim veri kümesi için önemlidir. Bu çalışmada rasgele alınan veriler kullanıldığı için karşılaşılan bir durum olabileceği de göz önüne alınmasına rağmen, KDD'99 veri kümesinde genel tarama yapıldığında da benzer bir durum olduğu görülmüştür.

Çizelge 7.5. Test kümesi #2

Saldırı tipi	**Saldırı adı**	**Veri sayısı**	**Saldırı tipine göre toplam**	**Tüm saldırılar toplam**	**Toplam test verisi sayısı**
DoS	Back	99	41791	48739	65536
	Neptune	20561			
	Pod	23			
	Smurf	15700			
	Udpstorm	1			
	Mailbomb	5000			
	Prpcesstable	506			
U2R	Buffer_overflow	13	37		
	Loadmodule	2			
	Perl	1			
	Rootkit	4			
	Ps	8			
	Xterm	9			
R2L	Guess_passwd	163	4257		
	İmap	1			
	Multihop	9			
	Warezmaster	1078			
	Sendmail	6			
	Snmpgetattack	2938			
	Xlock	3			
	Xsnoop	2			
	Httptunnel	56			
	Spy	1			
Probe	Ipsweep	176	2654		
	Nmap	43			
	Portsweep	139			
	Satan	1243			
	Mscan	1053			
Normal	Normal	16698	16698	16698	

Çizelge 7.6. Test kümesi #3

Saldırı tipi	**Saldırı adı**	**Veri sayısı**	**Saldırı tipine göre toplam**	**Tüm saldırılar toplam**	**Toplam test verisi sayısı**
DoS	Land	8	39661	47100	65536
	Neptune	20580			
	Pod	23			
	Smurf	18648			
	Teardrop	6			
	Apache2	396			
U2R	Buffer_overflow	3	24		
	Rootkit	9			
	Ps	8			
	Xterm	4			
R2L	Ftp_write	3	6752		
	Guess_passwd	4202			
	Multihop	2			
	Phf	1			
	Warezmaster	524			
	Sendmail	3			
	Snmpgetattack	1919			
	Httptunnel	99			
	Spy	1			
Probe	Ipsweep	50	663		
	Nmap	1			
	Portsweep	109			
	Satan	390			
	Saint	113			
Normal	Normal	18434	18434	18434	

Çizelge 7.7. Test kümesi #4

Saldırı tipi	**Saldırı adı**	**Veri sayısı**	**Saldırı tipine göre toplam**	**Tüm saldırılar toplam**	**Toplam test verisi sayısı**
DoS	Back	999	38235	40673	48885
	Neptune	16860			
	Smurf	20220			
	Teardrop	3			
	Processtable	253			
U2R	Buffer_overflow	1	1		
R2L	Snmpgetattack	2422	2422		
Probe	Nmap	15	15		
Normal	Normal	8212	8212	8212	

Çizelge 7.8. Test kümesi #5

Saldırı tipi	**Saldırı adı**	**Veri sayısı**	**Saldırı tipine göre toplam**	**Tüm saldırılar toplam**	**Toplam test verisi sayısı**
DoS	Land	1	52811	54655	65536
	Pod	19			
	Smurf	52788			
	Teardrop	3			
U2R	Buffer_overflow	2	3		
	Perl	1			
R2L	Multihop	3	1816		
	Named	7			
	Sendmail	2			
	Httptunnel	3			
	Snmpgetattack	1801			
Probe	Nmap	25	25		
Normal	Normal	10881	10881	10881	

7.3. Ön işlemler

Zeki STS uygulama örneğini geliştirmek için KDD'99 veri kümelerinden alınan verilerin formatı Çizelge 7.9'da gösterilmiştir. DARPA'nın tcpdump'dan elde ettiği verileri önişlemden geçirerek 41 özellik ve 1 çıkış haline getiren KDD'99 verilerinin, YSA ile kullanılabilmesi için bazı önişlemlerin yapılması gereklidir.

Çizelge 7.9. Veri kümesi örneği

```
0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,
1.00,0.00,0.00,255,252,0.99,0.01,0.00,0.00,0.00,0.00,0.00,0.00,0.00,snmpgetattack.
1,tcp,smtp,SF,3170,329,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,2,0.00,0.00,0.00,0.00,
1.00,0.00,1.00,54,39,0.72,0.11,0.02,0.00,0.02,0.00,0.09,0.13,normal.
0,tcp,http,SF,297,13787,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,
1.00,0.00,0.00,177,255,1.00,0.00,0.01,0.01,0.00,0.00,0.00,0.00,normal.
0,tcp,http,SF,291,3542,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,12,12,0.00,0.00,0.00,0.00,
1.00,0.00,0.00,187,255,1.00,0.00,0.01,0.01,0.00,0.00,0.00,0.00,normal.
```

KDD'99 veri kümesinde, tek bir örneğe ait olan her veri satırındaki 42 farklı alan, birbirlerinden ayrılmıştır. Bu 42 özelliğin neler olduğu Bölüm 5.2'de ayrıntılı olarak

anlatılmıştır. Çizelge 7.10'da özelliklerine göre sütunlara ayrılmış veriler gösterilmiştir.

Çizelge 7.10. Özelliklerin ayrıştırılmış formu

0	udp	private	SF	105	146	…	…	…	0.00	0.00	0.00	0.00	snmpgetattack.
1	tcp	smtp	SF	3170	329	…	…	…	0.02	0.00	0.09	0.13	normal.
0	tcp	http	SF	297	13787	…	…	…	0.00	0.00	0.00	0.00	normal.
0	tcp	http	SF	291	3542	…	…	…	0.00	0.00	0.00	0.00	normal.

Çizelge 7.11. Saldırı isimlerinin sayısal forma dönüştürülmesi

Saldırı	**Sayısal Değeri**	**Saldırı**	**Sayısal Değeri**
normal	0	teardrop	20
back	1	warezclient	21
buffer_overflow	2	warezmaster	22
ftp_write	3	apache2	23
guess_passwd	4	named	24
imap	5	saint	25
ipsweep	6	sendmail	26
land	7	snmpgetattack	27
loadmodule	8	udpstorm	28
multihop	9	xlock	29
neptune	10	xsnoop	30
nmap	11	mailbomb	31
perl	12	processtable	32
phf	13	mscan	33
pod	14	httptunnel	34
portsweep	15	ps	35
rootkit	16	xterm	36
satan	17	snmpguess	37
smurf	18	worm	38
spy	19	sqlattack	39

Sütunlara ayrılan bu verilerden bazıları sayısal formatta olmadığından, YSA'da giriş olarak kullanılabilmesi için, sayısal formata çevrilmesi gerekmiştir. Ancak bu çevirimlerin elle yapılması, veri setinin büyüklüğünden dolayı mümkün değildir. Bu nedenle, önişlemlerin yapılması için bir ön işleme yazılım modülü geliştirilmiştir. Bu modülde, veri kümesinde yer alan protokol, servis, bayrak (flag) ve saldırı tipleri alanlarının sayısal forma dönüştürülmesi gerçekleştirilmiştir. Bu alanlardaki veriler

ve sayısal karşılıkları Çizelge 7.11, Çizelge 7.12, Çizelge 7.13 ve Çizelge 7.14'de verilmiştir.

Çizelge 7.12. Servis isimlerinin sayısal forma dönüştürülme tablosu

Servis	**Sayısal Değeri**	**Servis**	**Sayısal Değeri**
http	0	exec	33
smtp	1	printer	34
finger	2	efs	35
domain_u	3	courier	36
auth	4	uucp	37
telnet	5	klogin	38
ftp	6	kshell	39
eco_i	7	echo	40
ntp_u	8	discard	41
ecr_i	9	systat	42
other	10	supdup	43
private	11	iso_tsap	44
pop_3	12	hostnames	45
ftp_data	13	csnet_ns	46
rje	14	pop_2	47
time	15	sunrpc	48
mtp	16	uucp_path	49
link	17	netbios_ns	50
remote_job	18	netbios_ssn	51
gopher	19	netbios_dgm	52
ssh	20	sql_net	53
name	21	vmnet	54
whois	22	bgp	55
domain	23	Z39_50	56
login	24	ldap	57
imap4	25	netstat	58
daytime	26	urh_i	59
ctf	27	X11	60
nntp	28	urp_i	61
shell	29	pm_dump	62
IRC	30	tftp_u	63
nnsp	31	tim_i	64
http_443	32	red_i	65

Çizelge 7.13. Protokol isimlerinin sayısal dönüşümleri

Protokol	**Sayısal Değeri**
Tcp	0
Udp	1
Icmp	2

Çizelge 7.14. Bayrak (Flag) isimlerinin sayısal forma dönüştürülmesi

Bayrak (Flag)	**Sayısal Değeri**
S0	0
S1	1
S2	2
S3	3
SF	4
SH	5
OTH	6
REJ	7
RSTO	8
RSTOSO	9
RSTR	10

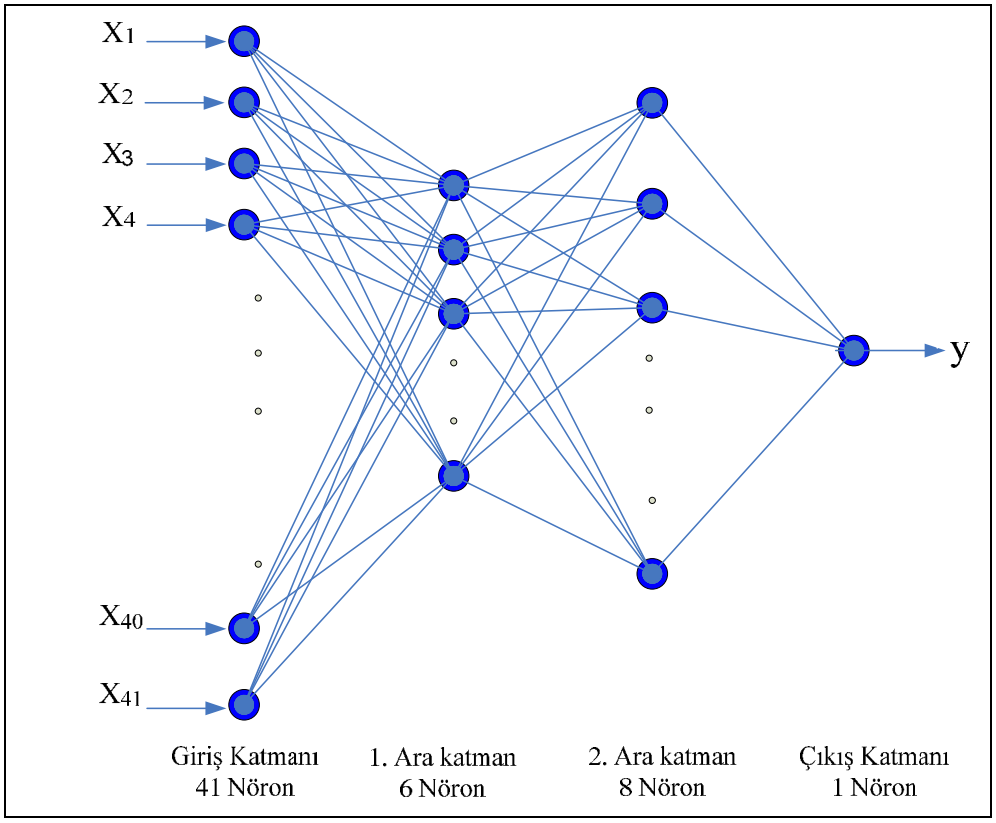
7.4. YSA Yapısı

Bu tez çalışmasında gerçekleştirilen zeki STS yazılımında bulunan YSA yapısı "deneme-yanılma" yolu tercih edilerek belirlenmiştir [69]. Eğitim veri kümesinin seçimine ve miktarına da bağlı olarak elde edilen sonuçlardan, uygun ve basit bir yapı tercih edilmiştir.

KDD'99 verilerinden elde edilen tüm girişler temel alınarak, 41 giriş nöronu kullanılmıştır. Birinci ara katman 6 ve ikinci ara katman ise 8 nörondan oluşmaktadır. YSA'dan elde edilen çıkış "saldırı" (0) ya da "normal" (1) olarak belirlendiğinden, tek bir çıkış nöronu kullanılmıştır.

YSA yapısı olarak MLP kullanılırken, öğrenme algoritması olarak Levenberg-Marquardt (LM) öğrenme algoritması tercih edilmiştir. MLP yapısının seçilmesinin nedeni, bilinen en eski YSA modellerinden olması ve sınıflandırma problemlerinde başarılı sonuçlar üretmesidir. Literatürde farklı yapılar ve öğrenme algoritmaları ile gerçekleştirilen zeki STS uygulamaları Bölüm 3'te tanıtılmıştır.

Tez kapsamında gerçekleştirilen bu uygulamanın asıl amacı olan YSA tabanlı zeki STS'lerin tasarımı ve uygulaması için temel bilgileri içeren ve yol gösteren bir çalışma yapılmasına özellikle özen gösterilmiştir.



Şekil 7.2. Zeki STS için kullanılan YSA yapısı

7.5. Eğitim ve Test Simülasyonları

Bu tez çalışmasında geliştirilen zeki STS'nin YSA modülü eğitimi gerçekleştirilirken takip edilen adımlar:

(1) YSA yapısının oluşturulması

(2) Eğitim veri setinin okunması ve normalize edilmesi

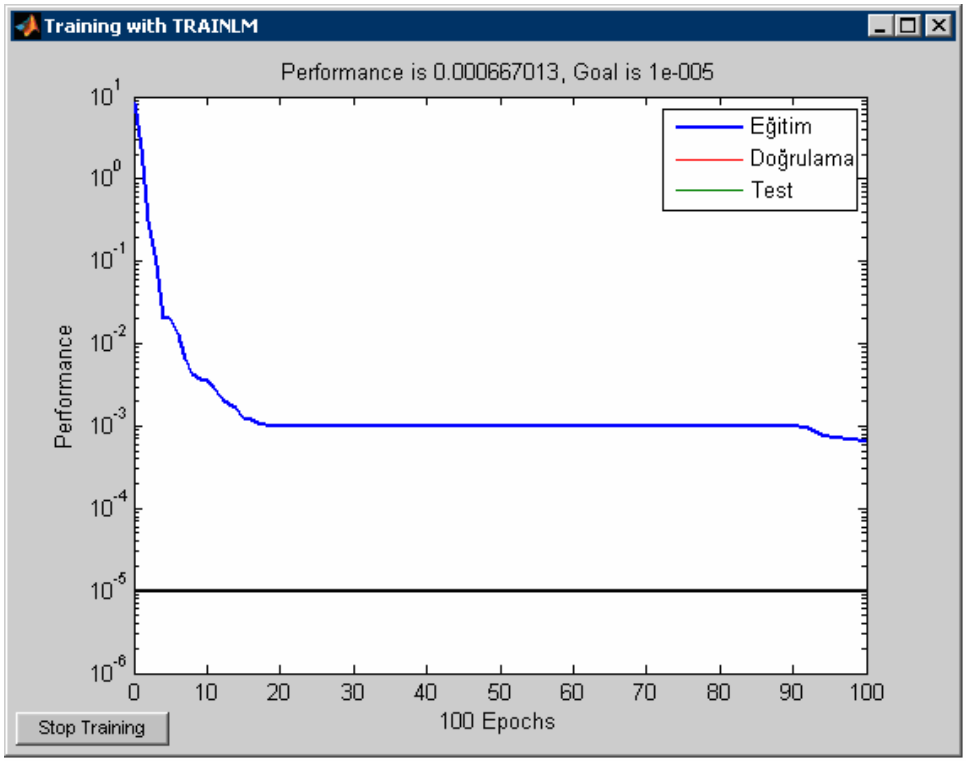
(3) İlk ağırlık değerlerinin atanması ve

(4) Eğitime başlanması,

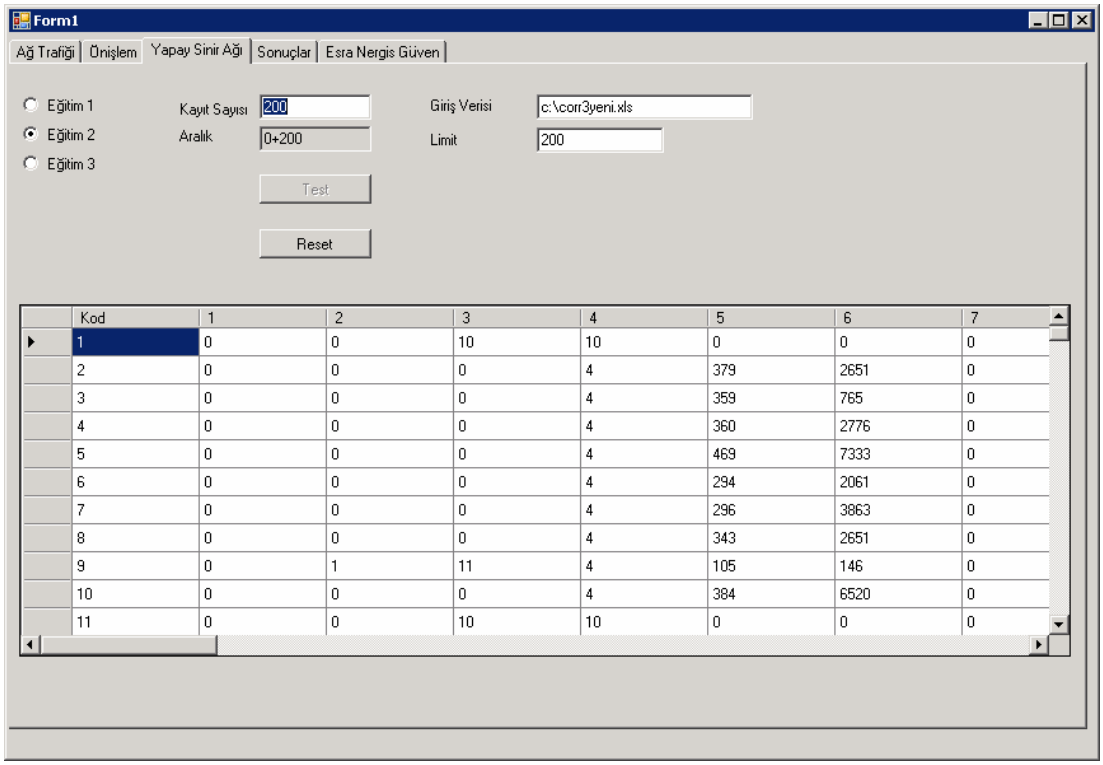
(5) Belirlenen epok sayısına kadar eğitimin devam ettirilmesi

(6) Sonuçların değerlendirilmesi

Geliştirilen zeki STS'nin YSA modülü eğitim veri kümelerinden 1.sine ait eğitim simülasyonu Şekil 7.3'de gösterilmiştir.



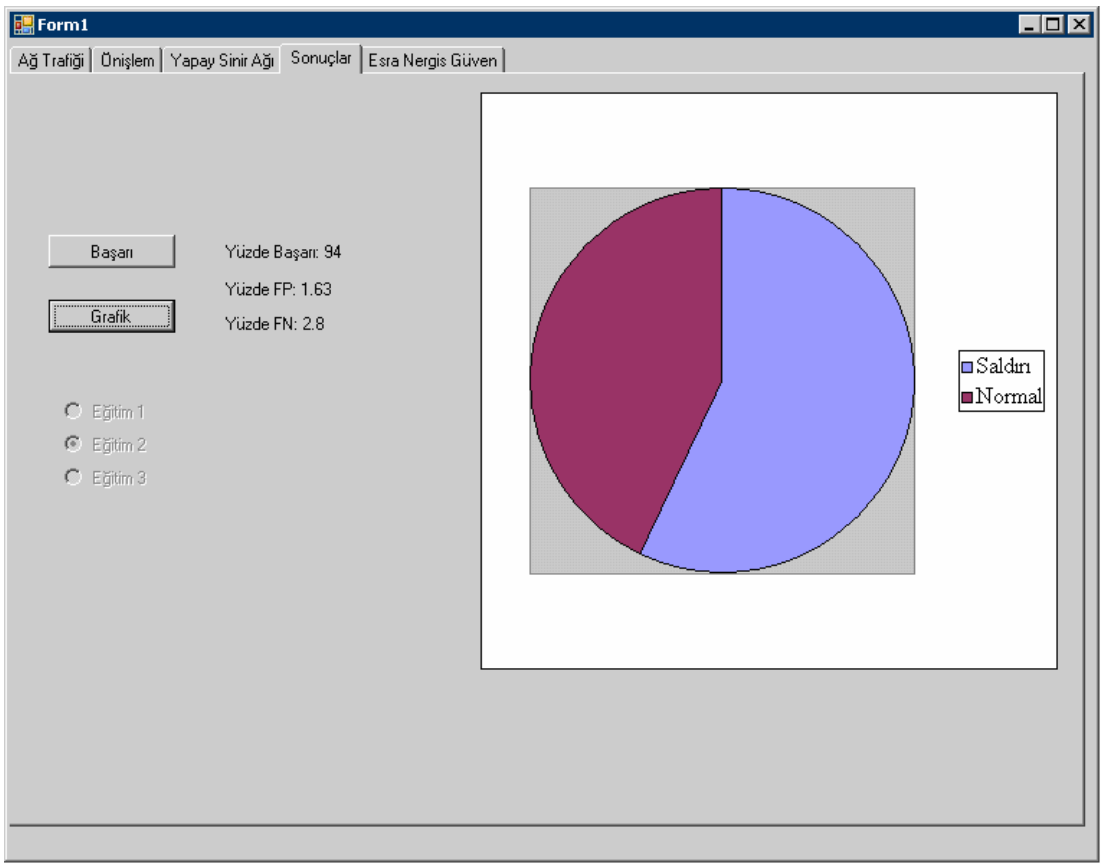
Şekil 7.3. Eğitim simülasyonu



Şekil 7.4. Uygulama yazılımı arayüzü

Zeki STS içerisinde bulunan YSA modülü eğitimi Matlab 2007 kullanılarak yapılmış ve veri kümelerinin tüm test işlemleri C# ile geliştirilen uygulama yazılımı ile hesaplanmıştır.

Şekil 7.4'de verilen test arayüzü, daha önceden eğitilmiş üç YSA yapısından birisi seçilerek, istenilen test kümesinden belirlenen aralık sayısı miktarında veri alınarak test işleminin kolaylıkla yapılmasını sağlamaktadır.



Şekil 7.5. Sonuç ve başarı değerlendirme arayüzü

Şekil 7.5'de gösterilen sonuç ve başarı değerlendirme arayüzü, test edilen veri kümesi sonuçlarının, elde edilmek istenen sonuçlar ile karşılaştırılarak, tespit etme oranı, yanlış alarm ve yanlış negatif oranlarının hesaplanması için geliştirilmiştir. Geliştirilen arayüzde gösterilen grafik, test verilerinin ne kadarının saldırı, ne kadarının normal veri olduğunu göstermektedir.

7.6. Eğitim ve Test Sonuçları

Geliştirilen uygulama sonucunda eğitim verilerinin test edilmesi ile elde edilen sonuçlar Çizelge 7.15'de sunulmuştur.

Çizelge 7.15'den görülebileceği gibi kullanılan eğitim veri kümelerinin sistem tarafından öğrenildiği yapılan test ile sonucunda doğrulanmıştır. Çok yüksek başarı ile öğrenen sistemin Çizelge 7.16'da gösterilen test veri kümesi sonuçları ise sistemin başarılı olduğunu ve eğitim setinde olmayan yeni saldırıları tiplerinin de algılandığını göstermektedir. Farklı eğitim veri kümeleri ile eğitilen YSA'ların, beş farklı test kümesi ile test edilmesinin sonuçları Çizelge 7.16'da sunulmuştur.

Çizelge 7.15. Eğitim verilerinin sonuçları

Eğitim veri kümesi	Doğruluk oranı	Yanlış alarm (yanlış pozitif)	Yanlış negatif
Eğitim #1	%99,12	%1,04	%0,99
Eğitim #2	%96,45	%2,63	%0,73
Eğitim #3	%98,57	%0,35	%0,92
Ortalama	%98,05	%1,34	%0,88

Çizelge 7.16. Test verilerinin sonuçları

Test veri kümesi	YSA#1	YSA#2	YSA#3
Test veri kümesi #1	%97,80	%97,59	%97,92
Test veri kümesi #2	%83,48	%83,66	%84,76
Test veri kümesi #3	%88,57	%81,93	%89,10
Test veri kümesi #4	%96,11	%88,97	%96,09
Test veri kümesi #5	%92,57	%93,31	%94,61
Ortalama	%91,71	%89,09	%92,50

Elde edilen sonuçlar sistemin başarılı olduğunu ve YSA'nın zeki STS'ler için oldukça başarılı bir yöntem olduğunu göstermektedir.

8. SONUÇ VE ÖNERİLER

Bu çalışmada, STS'ler genel olarak incelenmiş, klasik ve zeki STS'ler gözden geçirilmiş, STS tasarımında kullanılan veri kümeleri değerlendirilmiştir.

Literatür incelemesinden elde edilen bilgiler doğrultusunda, YSA tabanlı zeki bir STS tasarımı yapılarak elde edilen sonuçlar sunulmuştur. Elde edilen sonuçlardan;

- YSA'nın zeki STS tasarımında başarılı olduğu,
- YSA'nın başarısında saldırıların doğru şekilde tespit edilebilmesi için uygun veri kümesi tasarımının önemli olduğu,
- YSA'ların ara katman sayısı, nöron sayısı ve seçilecek olan fonksiyon tipine bağlı olarak başarıların arttığı,
- Eğitim ve test verilerinin sayısının fazla olmasına karşın tamamının kullanılmasının yüksek hesaplama zamanı gerektirdiği ve yapılan testler sonucunda az sayıda örneğin olduğu veri kümeleri kullanıldığında saldırı tespitinde başarının düştüğü,
- YSA'ların farklı veri tiplerinde başarılarının ölçülmesinin şart olduğu,
- Eğitim kümesi içerisinde bulunmayan yeni saldırı tiplerinin test esnasında tespit edilebildiği,
- Tasarlanan YSA yapısının kolaylıkla gerçek-zamanlı uygulamalarda kullanılabileceği,

Literatürde incelenen çalışmaların birçoğunda öğrenme algoritması olarak BP kullanılmışken, bu çalışmada farklı bir öğrenme algoritmasının da kullanılabileceği görülmüştür.

Çalışma genel olarak değerlendirildiğinde;

- Ülkemizde zeki saldırı tespit sistemlerine yönelik yeterli çalışma bulunmadığı,
- Ülkemizde geliştirilecek olan STS'lerin testinde kullanılabilecek bir veri kümesi bulunmadığı,
- Literatürde, veri kümesi oluşturmak için yapılan çalışmaların güncel olmadığı,

- Literatürde birçok çalışma mevcut olsa da çalışmalarda veri paketlerinin ayrıştırılması ve anlaşılabilir bir forma dönüştürülmesi için kolaylıkla uygulanabilecek bir yazılım bulunmadığı anlaşılmıştır.

Bundan dolayı bu çalışmanın, ülkemizde ilk kapsamlı çalışma olması, YSA uygulamasının tüm adımlarının açıkça sunulması, veri kümelerinin anlamlı bir hale getirilmesi için bir yazılım geliştirilmesi, tasarlanan YSA'nın farklı test kümeleri için bile yüksek başarı sunması, bu çalışmanın önemini daha da arttırmaktadır.

Bölüm 6 ve 7'de sunulan örneklerin, ülkemizde zeki STS'lerin tasarlanması ve test edilmesine yönelik olarak oluşturulacak yapıları, veri kümelerini ve kullanılacak yaklaşımları barındırması ise bu çalışmanın diğer katkıları olarak sıralanmaktadır.

Bu çalışmada tasarımı sunulan ulusal saldırı veri kümelerinin oluşturulması için çalışmalara başlanmış ve bu kapsamda BAP projesi ile alınan yazılım ve donanımlar ile gerekli ağ yapısı oluşturularak, örnek veri kümesi oluşturulmaya başlanmıştır. Daha kapsamlı ve gerçek trafik verilerinden elde edilen veri kümesi oluşturabilmek için, gerekli girişimlerde bulunulmuş ve bu konuda olumlu gelişmeler sağlanmıştır. Ortaklaşa yapılacak olan bu projenin sonuçları araştırmacıların kullanımına açılarak, ülkemize katkı sağlaması hedeflenmektedir.

Bu araştırmada elde ettiğimiz bilgi ve deneyimlere dayanarak, saldırıları tespit etme yöntemlerine göre ikiye ayrıldığını bildiğimiz STS'lerin, hangi yöntemin seçileceğine dair dikkat edilmesi gereken noktası, anormallik tespiti yönteminin, bütün kötü davranışları tespit etmeye çalışırken, kötüye kullanım tespiti yönteminin kötü olarak bilinen davranışları tanımaya çalışmak olduğudur. Her iki yöntemin de avantaj ve dezavantajları olduğu göz önünde bulundurularak, avantajları bir araya toplayan hibrit yaklaşımlardan faydalanmak daha gerçekçi çözümler üretmemizi sağlayabilir. Bu amaçla STS'lerde zeki yaklaşımlar kullanmak, false pozitif oranını düşürürken, atak tespit oranını artırabilir. Bu yöntemdeki asıl amaç normalde doğru orantılı olan bu değerlerin, ters orantılı hale geldiği bir sistem oluşturmaktır. Bu çalışmada sunulan önerinin buna katkı sağlayacağı değerlendirilmektedir.

Gerçekleştirilen çalışmada STS'ler incelenmiş, bilgisayar sistemleri için önemi örneklerle gösterilmiş, kullanılan yaklaşımlara ayrıntılı olarak değinilmiş ve bir zeki STS tasarımı yapılmıştır. Geliştirilen zeki STS'nin tespit ettiği saldırıların ve başarı oranlarının önceki çalışmalar ile karşılaştırılarak değerlendirmesi yapılmıştır. Bu çalışmada uygulanan yaklaşımın başarı oranları değerlendirildiğinde elde edilen sonuçlardan, zeki yaklaşımın bu tip problemlere uygulanabilir olduğu gösterilmiştir. Uygulama aşamasında çalışmayı yavaşlatan en büyük problem veritabanları olmuştur. Bunun için ayrı bir dinleyici modülü geliştirilerek bu sorunun üstesinden gelinmiştir. Geliştirilen tasarımla birlikte yazılımın gerçek veri paketleri üzerinde de çalışabilmesi için ortam sağlanmıştır.

Bu tez çalışması kapsamında gerçekleştirilen zeki STS uygulamasında, farklı eğitim ve test kümeleriyle yapılan testlerden elde edilen en iyi sonuç, %97,92 başarılı iken, en düşük sonuç %81,93 olarak tespit edilmiştir.

Gerçekleştirilen sistemin test sonuçları, diğer çalışmalarla karşılaştırıldığında oldukça başarılı sonuçlar ürettiği görülmüştür. Elde edilen bu başarının sonucunda LM algoritmasının, STS veri kümelerinde uygulanabilir olduğunu göstermiştir.

Gerçekleştirilen çalışma, bilgi güvenliğinin korunması açısından farklı bakış açıları kazanılmasını sağlamıştır. Literatürde de belirtildiği gibi zeki yaklaşımların STS tasarımında kullanılmasının doğru bir tercih olduğu, farklı yapılar seçerken MLP yapısının LM gibi güçlü algoritmalarla eğitilmesiyle YSA'ların STS tasarımında başarıyı artırdığı tespit edilmiştir.

Bu tez çalışmasında karşılaşılan güçlükler;

- STS'lerin geliştirilmesi için saldırı verilerinin toplanması,
- Toplanan verileri işlemenin zaman alıcı olması,
- Ticari yönü olmasından dolayı güncel saldırı verileri hakkında açıkça bilgi verilmemesi,
- Mevcut veri kümelerinin güncel olmaması,

- Saldırı araçlarıyla veri oluşturmanın ve bu saldırıları etiketleme işlemlerinin uzun zaman alması,
- Saldırı veri kümeleri ve STS için altyapı oluşturmanın maliyetli ve zaman alıcı olması, olarak sıralanabilir.

KAYNAKLAR

1. Sağıroğlu, Ş., Alkan, M., "Her yönüyle elektronik imza (e-imza)", ***Grafiker Yayınları***, Ankara, 1-100 (2005).

2. Endorf, C., Schultz E., Mellander J., "Intrusion Detection & Prevention", Jenn Tust, Jody McKenzie, Elizabeth Seymour, ***McGraw-Hill***, California, 10-150 (2004).

3. Mahoney, M.V., Chan, P.K., "An analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for network anomaly detection", ***Recent Advances in Intrusion Detection (RAID2003),*** Lecture Notes in Computer Science., Springer-Verlag, 2820, 220-237 (2003).

4. Anderson, J.P., "Computer security threat monitoring and surveillance", ***Technical Report, Fort Washington, Pennsylvania,*** 1-30 (1980).

5. Pei, J., Upadhyaya, S.J., Farooq, F., Govindaraju, V., "Data mining for intrusion detection: techniques, applications and systems," ***20th International Conference on Data Engineering (ICDE'04)***, 1063-6382 (2004).

6. Lunt, T.F. "Automated audit trail analysis and intrusion detection: A survey", ***11th National Computer Security Conference,*** Baltimore, MD, 65-73 (1988).

7. Denning, D.E., "An intrusion detection model", ***IEEE Transactions on Software Engineering***, 13(2): 118–131 (1987).

8. Mukherjee, B., Heberlein, L.T., Levitt, K.N., "Network intrusion detection", ***IEEE Network***, 8(3): 26-41 (1994).

9. Crosbie, M., Spafford, E.H., "Defending a computer system using autonomous agents", ***Technical Report 95-022, Dept. of Comp. Sciences, Purdue University, West Lafayette,*** 1-11 (1995).

10. Endler, D., "Intrusion detection applying machine learning to solaris audit data", ***1998 Annual Computer Security Applications Conference (ACSAC'98)***, 268-269 (1998).

11. Axelsson, S., "Intrusion detection systems: A survey and taxonomy", ***Technical Report 99-15, Dept. of Computer Eng., Chalmers University of Technology, Göteborg, Sweden,*** 1-23 (2000).

12. Patcha, A., Park, J.M., "An overview of anomaly detection techniques: Existing solutions and latest technological trends", ***Computer Networks,*** 51(12): 3448-3470 (2007).

13. Anderson, D., Lunt, T.F., Javitz, H., Tamaru, A., Valdes, A., "Detecting unusual program behavior using the statistical component of the next-generation intrusion detection expert system (NIDES)", ***SRI-CSL-95-06, Menlo Park, California***, 1-22 (1995).

14. Hofmeyr, S.A., "An immunological model of distributed detection and its application to computer security", Doktora Tezi, ***Computer Science, University of New Mexico***, 1-69 (1999).

15. Dickerson, J.E., Dickerson J.A., "Fuzzy network profiling for intrusion detection" ***NAFIPS 19th International Conference of the North American Fuzzy Information Processing Society***, Atlanta, 301-306, (2000).

16. İnternet: Massachusetts Teknoloji Enstitüsü Lincoln Laboratuarları "Off-line intrusion detection evaluation data" http://www.ll.mit.edu/IST/ideval/ (2007).

17. İnternet: Knowledge Discovery and Delivery, "KDD Cup 1999: General Information",http://www.sigkdd.org/kddcup/index.php?section=1999&method=info (2007).

18. İnternet: MITRE-CWE, Common Weakness Enumeration, "Vulnerability Type Distributions in CVE", http://cwe.mitre.org/documents/vuln-trends/index.html (2007).

19. Sundaram, A., "An introduction to intrusion detection", ***Crossroads: The ACM Student Magazine,*** New York, USA, 2(4), 3-7 (1996).

20. Jones, A.K., "Computer System Intrusion Detection: A Survey", ***Technical Report, Computer Science Dept., University of Virginia***, Charlottesville, Virginia, 1-21 (2000).

21. Murali, A., Rao, M., "A survey on intrusion detection approaches", ***First International Conference on Information and Communication Technologies***, IEEE Communications Society Press, 233-240 (2005).

22. Lee, W., Stolfo, S.J., Chan, P.K., Eskin, E., Fan, W., Miller, M., Hershkop, S., Zhang, J., "Real time data mining-based intrusion detection", ***Second {DARPA} Information Survivability Conference and Exposition (DISCEX II)***, Anaheim, CA, 89-100 (2001).

23. Ilgun, K., Kemmerer, R., Porras, P., "State Transition Analysis: A RuleBased Intrusion Detection System", ***Software Engineering,*** 21(3): 181-199 (1995).

24. Porras, P.A., "STAT: A State Transition Analysis Tool for intrusion detection", Yüksek Lisans Tezi, ***Computer Science Department, University of California***, Santa Barbara, 1-150 (1992).

25. Cabrera, B.D., Cabrera, L. Lewis and R.K. Mehra, "Detection and classification of intrusions and faults using sequence of system calls", ***ACM SIGMOD record***, 30(4): 25-34 (2001).

26. Bace, R., Mell, P., "Intrusion detection systems", ***Technical Report, National Institute of Standards and Technology, NIST SP300-31, Scotts Valley, CA,*** 5-46 (2001).

27. Mukkamala, S., Janoski, G., Sung, A., "Intrusion detection using neural networks and support vector machines", ***IEEE International Joint Conference on Neural Networks***, IEEE Computer Society Press, 1702-1707 (2002).

28. Mell, P., Hu, V., Lipmann, R., Haines, J., Zissman, M., "An overview of issues in testing intrusion detection systems", ***Technical Report NIST IR 7007, National Institute of Standard and Technology,*** 1-18 (2003).

29. Sağıroğlu, Ş., Beşdok, E., Erler, M., "Mühendislikte yapay zeka uygulamaları-1: Yapay sinir ağları", ***Ufuk Kitabevi,*** Kayseri, 10-100 (2003).

30. Dickerson, J.E., Juslin, J., Koukousoula, O., Dickerson, J.A., "Fuzzy intrusion detection", ***IFSA World Congress and 20th North American Fuzzy Information Processing Society (NAFIPS) International Conference***, Vancouver, British Columbia, 3: 1506-1510 (2001).

31. K. Lee and L. Mikhailov: "Intelligent Intrusion Detection System", ***Second IEEE International. Conference on Intelligent Systems***, 2: 497-502 (2004).

32. Cho, S.B., "Incorporating soft computing techniques into a probabilistic intrusion detection System", ***IEEE Transactions on Systems, Man, and Cybernetics***, Part C 32(2): 154-160 (2002).

33. Stein, G., Chen, B., Wu, A.S., Hua, K.A., "Decision tree classifier for network intrusion detection with GA-based feature selection", ***ACM Southeast Regional Conference (2),*** Kennesaw, Georgia, 136-141 (2005).

34. Garrett, S.M., "How Do We Evaluate Artificial Immune Systems?", ***Evolutionary Computation***, MIT Press, 13(2): 145-178 (2005).

35. Esponda, F., Forrest, S., Helman, P., "A formal framework for positive and negative detection schemes", ***IEEE Transactions on Systems, Man, and Cybernetics, Part B, Cybernetics,*** 34(1): 357-373 (2004).

36. Mukkamala, S., Sung, A.H., Abraham, A., "Intrusion detection using ensemble of soft computing paradigms", ***Third International Conference On Intelligent Systems Design and Applications***, Germany: Springer, 239-248 (2003).

37. Wassmer, R.E., Fikus, J.H., "Advanced Intrusion Detection Techniques", ***Final rept., Defense Technical Information Center - ADA410454,*** 2. evre, 1-14, (2003).

38. Cannady J., “Artificial neural networks for misuse detection”, ***Proceedings of the 1998 National Information Systems Security Conference (NISSC'98),*** Arlington, VA, 443-456 (1998).

39. Frank, J., “Artificial intelligence and intrusion detection: current and future directions”, ***Division of Computer Science, University of California at Davis,*** 1-12 (1994).

40. Peddabachigari, S., Abraham, A., Grosan, C., Thomas, J., “Modeling intrusion detection system using hybrid intelligent systems”, ***Journal of Network and Computer Applications, Elsevier,*** 30:114–132 (2007).

41. Ryan, J., Lin, M.J., Mikkulainen, R., “Intrusion Detection with Neural Networks”, ***Advances in Neural Information Processing systems 10***, Cambridge, MA, MIT Press, 1-7 (1998).

42. Lee, S.C., ve Heinbuch, D.V., “Training a Neural-Network Based Intrusion Detector to Recognize Novel Attacks”, ***IEEE Transactions on systems, man, and Cybernetics-Part A: Systems and Humans,*** 31(4): 294-299 (2001).

43. Bivens, A., Palagiri, C., Smith, R., Symanski, B., Embrechts, M., “Network-Based Intrusion Detection Using Neural Networks”, ***Intelligent Engineering Systems through Artificial Neural Networks (ANNIE-2002),*** New York: ASME Press, 579-584 (2002).

44. Yang, X.R., Shen, J.Y., Wang, R., “Artificial Immune Theory Based Network Intrusion Detection System And The Algorithms Design”, ***First Internetional Conference on Machine Learning and Cybernetics***, Beijing, 73-77 (2002).

45. Moradi, M., Zulkernine, M., “A neural network based system for intrusion detection and classification of attacks,” 2004 IEEE International Conference on Advances in Intelligent Systems - Theory and Applications, Luxembourg-Kirchberg, Luxembourg, 148:1-6 (2004).

46. Mukkamala, S., Sung, A.H., “Artificial Intelligent Techniques for Intrusion Detection”, ***IEEE International Conference on Systems, Man and Cybernetics***, Washington D.C., USA, 2: 1266 - 1271 (2003).

47. Mukkamala, S., Sung, A.H., “Feature Selection for Intrusion Detection using Neural Networks and Support Vector Machines”, ***Journal of the Transportation Research Board of the National Academies***, Transportation Research Record No. 1822, Washington D.C., USA, 33-39 (2003).

48. Jackson, K.A., “Intrusion detection system (IDS) product survey”, ***Technical Report LA-UR-99-3883, Los Alamos National Laboratory, Los Alamos, New Mexico***, 1-96 (1999).

49. Smaha S.E., "Haystack: An intrusion detection system", ***In Fourth Aerospace Computer Security Applications Conference,*** , Tracor Applied Science Inc., Austin, Texas, 37-44 (1988).

50. İnternet: SRI International's Computer Science Lab, "IDES history", http://www.csl.sri.com/programs/intrusion/history.html#IDES (2007).

51. Anderson, D., Frivold, T., Tamaru, A., Valdes, A., "Next Generation Intrusion Detection Expert System (NIDES), SRI-CSL-95-07***, Menlo Park, California***, 10-74 (1994).

52. Vaccarro, H. S. ve Liepins, G. E. "Detection of Anomalous Computer Session Activity", ***IEEE Symposium on Research in Security and Privacy***, Oakland, California, 280-289 (1989).

53. McAuliffe, N., Wolcott, D., Schaefer, L., Kelem, N., Hubbard, B., Haley, T., "Is your computer being misused? A survey of current intrusion detection system technology," ***Sixth Annual Computer Security Applications Conference***, Tucson, AZ, 260-272 (1990).

54. Christoph, G.G., Jackson, K.A., Neuman, M.C., Siciliano, C.L.B., Simmonds, D.D., Stallings, C.A., Thompson, J.L., "UNICORN: Misuse Detection for UNICOS", ***Proceedings of the 1995 ACM/IEEE conference on Supercomputing,*** San Diego, California, United States, 56-80 (1995)

55. Ilgun, K., "Ustat: A real-time intrusion detection system for Unix", ***Proceedings of the 1993 IEEE Symposium on Research in Security and Privacy***, Oakland, California, 16-28, (1993)

56. Snapp, S.R., Brentano, J., Dias, G.V., Goan, T.L., Heberlein, L.T., Ho, C., Levitt, K.N., Mukherjee, B., Smaha, S.E., Grance, T., Teal, D.M., Mansur, D., "DIDS (Distributed Intrusion Detection System) - Motivation, Architecture, and an Early Prototype", ***14th National Computer Security Conference,*** 167-176 (1991).

57. Kumar, S., "Classification and detection of computer intrusions", Doktora Tezi, ***Department of Computer Science, Purdue University,*** West Lafayette, IN, USA, 1-75 (1995).

58. Porras, P.A., Neumann, P.G., "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances", ***20th NIST-NCSC National Information Systems Security Conference,*** Baltimore, Maryland, 353-365, (1997).

59. Paxson, V., "Bro: A system for detecting network intruders in real-time", ***7th USENIX Security Symposium,*** San Antonio, TX, 32-53 (1998).

60. İnternet: Snort - open source network intrusion prevention and detection system, "Snort Downloads", http://www.snort.org/dl/ (2007).

61. İnternet: Massachusetts Teknoloji Enstitüsü Lincoln Laboratuarları "1998 DARPA Intrusion Detection Evaluation Data Set Overview", http://www.ll.mit.edu/IST/ideval/data/1998/1998_data_index.html (2007).

62. İnternet: Massachusetts Teknoloji Enstitüsü Lincoln Laboratuarları "1999 DARPA Intrusion Detection Evaluation Data Set Overview" http://www.ll.mit.edu/IST/ideval/data/1999/1999_data_index.html (2007).

63. İnternet: Massachusetts Teknoloji Enstitüsü Lincoln Laboratuarları "2000 DARPA Intrusion Detection Evaluation Data Set Overview" http://www.ll.mit.edu/IST/ideval/data/2000/2000_data_index.html (2007).

64. Kayacik H. G., Zincir-Heywood A. N., Heywood M. I., "Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets", ***Third Annual Conference on Privacy, Security and Trust (PST-2005),*** St. Andrews, Canada, 85-89 (2005).

65. Kayacik H. G., Zincir-Heywood A. N., Heywood M. I., "A hierarchical SOM-based intrusion detection System", ***Engineering Applications of Artificial Intelligence***, Elsevier, 20(4): 439-451, (2007).

66. İnternet: Hping, "Security tool documentation" http://www.hping.org/documentation.php (2007).

67. İnternet: Tcpdump- dump traffic on a network, "Tcpdump documents", http://www.tcpdump.org/#documentation (2007).

68. İnternet: Ethereal, network protocol analyzer, "Ethereal documentation", http://www.ethereal.com (2007).

69. Roy, A., "Artificial neural Networks - A science in trouble", ***ACM SIGKDD Explorations,*** ACM SIGKDD, 1(2): 33-38 (2000).

EKLER

EK-1 DARPA 1998 örnek veri kümesi

1 01/23/1998 16:56:12 00:01:26 telnet 1754 23 192.168.1.30 192.168.0.20 0 -
2 01/23/1998 16:56:15 00:00:13 ftp 1755 21 192.168.1.30 192.168.0.20 0 -
3 01/23/1998 16:56:17 00:00:01 smtp 43493 25 192.168.0.40 192.168.1.30 0 -
4 01/23/1998 16:56:17 00:00:00 auth 1756 113 192.168.1.30 192.168.0.40 0 -
5 01/23/1998 16:56:19 00:00:01 smtp 43494 25 192.168.0.40 192.168.1.30 0 -
6 01/23/1998 16:56:19 00:00:00 auth 1761 113 192.168.1.30 192.168.0.40 0 -
7 01/23/1998 16:56:19 00:00:01 ftp-data 20 1762 192.168.0.20 192.168.1.30 0 -
8 01/23/1998 16:56:22 00:00:00 ftp-data 20 1767 192.168.0.20 192.168.1.30 0 -
9 01/23/1998 16:56:24 00:00:02 ftp-data 20 1768 192.168.0.20 192.168.1.30 0 -
10 01/23/1998 16:56:25 00:01:01 telnet 1769 23 192.168.1.30 192.168.0.20 0 -
11 01/23/1998 16:56:27 00:00:00 ftp-data 20 1770 192.168.0.20 192.168.1.30 0 -
12 01/23/1998 16:56:36 00:00:03 finger 1772 79 192.168.1.30 192.168.0.20 0 -
13 01/23/1998 16:56:42 00:00:03 smtp 1778 25 192.168.1.30 192.168.0.20 0 -
14 01/23/1998 16:56:43 00:00:03 smtp 1783 25 192.168.1.30 192.168.0.20 0 -
15 01/23/1998 16:56:45 00:00:00 http 1784 80 192.168.1.30 192.168.0.40 1 phf
16 01/23/1998 16:56:49 00:00:14 ftp 43504 21 192.168.0.40 192.168.1.30 0 -
17 01/23/1998 16:56:56 00:00:00 ftp-data 20 43505 192.168.1.30 192.168.0.40 0 -
18 01/23/1998 16:56:57 00:00:00 ftp-data 20 43506 192.168.1.30 192.168.0.40 0 -
19 01/23/1998 16:56:59 00:00:00 ftp-data 20 43508 192.168.1.30 192.168.0.40 0 -
21 01/23/1998 16:57:00 00:00:00 ftp-data 20 43509 192.168.1.30 192.168.0.40 0 -
22 01/23/1998 16:57:02 00:00:00 ftp-data 20 43510 192.168.1.30 192.168.0.40 0 -
24 01/23/1998 16:57:13 00:00:48 telnet 43516 23 192.168.0.40 192.168.1.30 0 -
25 01/23/1998 16:57:15 00:00:12 ftp 1787 21 192.168.1.30 192.168.0.20 0 -
26 01/23/1998 16:57:16 00:00:01 http 1788 80 192.168.1.30 192.168.0.40 0 -
27 01/23/1998 16:57:19 00:00:02 http 1789 80 192.168.1.30 192.168.0.40 0 -
29 01/23/1998 16:57:20 00:00:05 smtp 43519 25 192.168.0.40 192.168.1.30 0 -
30 01/23/1998 16:57:22 00:00:00 auth 1790 113 192.168.1.30 192.168.0.40 0 -
31 01/23/1998 16:57:23 00:00:02 http 1796 80 192.168.1.30 192.168.0.40 0 -
32 01/23/1998 16:57:24 00:00:00 ftp-data 20 1801 192.168.0.20 192.168.1.30 0 -
33 01/23/1998 16:57:26 00:00:00 ftp-data 20 1802 192.168.0.20 192.168.1.30 0 -
34 01/23/1998 16:57:27 00:00:02 http 43521 80 192.168.0.40 192.168.1.30 0 -
35 01/23/1998 16:57:27 00:00:03 http 1804 80 192.168.1.30 192.168.0.40 0 -
36 01/23/1998 16:57:31 00:00:01 http 43522 80 192.168.0.40 192.168.1.30 0 -
37 01/23/1998 16:57:34 00:00:02 http 1806 80 192.168.1.30 192.168.0.40 0 -
38 01/23/1998 16:57:37 00:00:02 http 43524 80 192.168.0.40 192.168.1.30 0 -
39 01/23/1998 16:57:37 00:00:02 http 1807 80 192.168.1.30 192.168.0.40 0 -
41 01/23/1998 16:57:40 00:00:02 http 43525 80 192.168.0.40 192.168.1.30 0 -
42 01/23/1998 16:57:41 00:00:02 http 1808 80 192.168.1.30 192.168.0.40 0 -
43 01/23/1998 16:57:44 00:00:03 http 43526 80 192.168.0.40 192.168.1.30 0 -
44 01/23/1998 16:57:45 00:00:01 http 1810 80 192.168.1.30 192.168.0.40 0 -
45 01/23/1998 16:57:47 00:00:00 finger 1811 79 192.168.1.30 192.168.0.20 0 -
46 01/23/1998 16:57:48 00:00:03 http 43527 80 192.168.0.40 192.168.1.30 0 -
47 01/23/1998 16:57:48 00:00:02 http 1814 80 192.168.1.30 192.168.0.40 0 -
48 01/23/1998 16:57:52 00:00:02 http 43528 80 192.168.0.40 192.168.1.30 0 -
49 01/23/1998 16:57:53 00:00:03 http 1816 80 192.168.1.30 192.168.0.40 0 -
50 01/23/1998 16:57:55 00:00:03 http 1818 80 192.168.1.30 192.168.0.40 0 -
51 01/23/1998 16:57:55 00:00:01 finger 1820 79 192.168.1.30 192.168.0.20 0 -
53 01/23/1998 16:57:57 00:00:02 smtp 1826 25 192.168.1.30 192.168.0.20 0 -
54 01/23/1998 16:57:59 00:00:02 http 1830 80 192.168.1.30 192.168.0.40 0 -

EK-1 (Devam) DARPA 1998 örnek veri kümesi

55 01/23/1998 16:57:59 00:00:04 smtp 1832 25 192.168.1.30 192.168.0.20 0 -
56 01/23/1998 16:57:59 00:00:03 http 1833 80 192.168.1.30 192.168.0.40 0 -
57 01/23/1998 16:58:02 00:00:03 finger 1834 79 192.168.1.30 192.168.0.20 0 -
58 01/23/1998 16:58:03 00:00:02 http 1835 80 192.168.1.30 192.168.0.40 0 -
59 01/23/1998 16:58:03 00:00:02 http 1836 80 192.168.1.30 192.168.0.40 0 -
60 01/23/1998 16:58:04 00:00:01 http 43529 80 192.168.0.40 192.168.1.30 0 -
61 01/23/1998 16:58:06 00:00:02 http 1837 80 192.168.1.30 192.168.0.40 0 -
62 01/23/1998 16:58:06 00:00:02 http 1838 80 192.168.1.30 192.168.0.40 0 -
63 01/23/1998 16:58:07 00:00:01 http 43530 80 192.168.0.40 192.168.1.30 0 -
64 01/23/1998 16:58:10 00:00:01 http 1839 80 192.168.1.30 192.168.0.40 0 -
65 01/23/1998 16:58:11 00:00:01 finger 1841 79 192.168.1.30 192.168.0.20 0 -
66 01/23/1998 16:58:13 00:00:02 http 1844 80 192.168.1.30 192.168.0.40 0 -
67 01/23/1998 16:58:13 00:00:19 ftp 43532 21 192.168.0.40 192.168.1.30 0 -
68 01/23/1998 16:58:16 00:00:03 http 1846 80 192.168.1.30 192.168.0.40 0 -
69 01/23/1998 16:58:18 00:00:04 finger 1847 79 192.168.1.30 192.168.0.20 0 -
70 01/23/1998 16:58:20 00:00:02 http 1848 80 192.168.1.30 192.168.0.40 0 -
71 01/23/1998 16:58:20 00:00:00 ftp-data 20 43534 192.168.1.30 192.168.0.40 0 -
72 01/23/1998 16:58:21 00:00:02 http 1849 80 192.168.1.30 192.168.0.40 0 -
73 01/23/1998 16:58:22 00:00:17 ftp 1850 21 192.168.1.30 192.168.0.20 0 -
74 01/23/1998 16:58:23 00:00:02 http 1851 80 192.168.1.30 192.168.0.40 0 -
75 01/23/1998 16:58:24 00:00:02 http 1852 80 192.168.1.30 192.168.0.40 0 -
76 01/23/1998 16:58:27 00:00:03 http 1853 80 192.168.1.30 192.168.0.40 0 -
77 01/23/1998 16:58:28 00:00:02 finger 1855 79 192.168.1.30 192.168.0.20 0 -
78 01/23/1998 16:58:28 00:00:00 ftp-data 20 43536 192.168.1.30 192.168.0.40 0 -
79 01/23/1998 16:58:28 00:00:01 ftp-data 20 1854 192.168.0.20 192.168.1.30 0 -
80 01/23/1998 16:58:31 00:00:00 ftp-data 20 43537 192.168.1.30 192.168.0.40 0 -
81 01/23/1998 16:58:31 00:00:02 http 1857 80 192.168.1.30 192.168.0.40 0 -
82 01/23/1998 16:58:31 00:00:01 ftp-data 20 1856 192.168.0.20 192.168.1.30 0 -
83 01/23/1998 16:58:34 00:00:00 ftp-data 20 1858 192.168.0.20 192.168.1.30 0 -
84 01/23/1998 16:58:34 00:00:02 http 1859 80 192.168.1.30 192.168.0.40 0 -
85 01/23/1998 16:58:36 00:00:00 ftp-data 20 1860 192.168.0.20 192.168.1.30 0 -
86 01/23/1998 16:58:38 00:00:02 http 1861 80 192.168.1.30 192.168.0.40 0 -
87 01/23/1998 16:58:38 00:00:00 ftp-data 20 1863 192.168.0.20 192.168.1.30 0 -
88 01/23/1998 16:58:41 00:00:02 http 1864 80 192.168.1.30 192.168.0.40 0 -
89 01/23/1998 16:58:44 00:00:02 http 1866 80 192.168.1.30 192.168.0.40 0 -
90 01/23/1998 16:58:45 00:00:22 telnet 1867 23 192.168.1.30 192.168.0.20 1 guess
91 01/23/1998 16:58:47 00:00:02 http 1868 80 192.168.1.30 192.168.0.40 0 -
92 01/23/1998 16:58:48 00:00:02 http 1869 80 192.168.1.30 192.168.0.40 0 -
93 01/23/1998 16:58:48 00:00:11 ftp 43540 21 192.168.0.40 192.168.1.30 0 -
94 01/23/1998 16:58:51 00:00:02 http 1870 80 192.168.1.30 192.168.0.40 0 -
95 01/23/1998 16:58:52 00:00:00 ftp-data 20 43542 192.168.1.30 192.168.0.40 0 -
96 01/23/1998 16:58:54 00:00:02 http 1873 80 192.168.1.30 192.168.0.40 0 -
97 01/23/1998 16:58:57 00:00:00 ftp-data 20 43544 192.168.1.30 192.168.0.40 0 -
98 01/23/1998 16:58:57 00:00:01 http 1874 80 192.168.1.30 192.168.0.40 0 -
100 01/23/1998 16:59:00 00:00:02 http 1875 80 192.168.1.30 192.168.0.40 0 -
101 01/23/1998 16:59:01 00:00:45 telnet 1876 23 192.168.1.30 192.168.0.20 0 -
102 01/23/1998 16:59:03 00:00:01 http 1877 80 192.168.1.30 192.168.0.40 0 -
103 01/23/1998 16:59:06 00:00:02 http 1878 80 192.168.1.30 192.168.0.40 0 -
104 01/23/1998 16:59:09 00:00:02 http 1879 80 192.168.1.30 192.168.0.40 0 -

EK-1 (Devam) DARPA 1998 örnek veri kümesi

105 01/23/1998 16:59:12 00:00:02 http 1880 80 192.168.1.30 192.168.0.40 0 -
107 01/23/1998 16:59:15 00:00:01 http 1881 80 192.168.1.30 192.168.0.40 0 -
108 01/23/1998 16:59:18 00:00:01 http 1882 80 192.168.1.30 192.168.0.40 0 -
109 01/23/1998 16:59:21 00:00:01 http 1883 80 192.168.1.30 192.168.0.40 0 -
110 01/23/1998 16:59:23 00:00:24 telnet 1884 23 192.168.1.30 192.168.0.20 1 guess
111 01/23/1998 16:59:24 00:00:01 http 1885 80 192.168.1.30 192.168.0.40 0 -
112 01/23/1998 16:59:26 00:00:02 http 1886 80 192.168.1.30 192.168.0.40 0 -
113 01/23/1998 16:59:29 00:00:02 http 1887 80 192.168.1.30 192.168.0.40 0 -
115 01/23/1998 16:59:33 00:00:02 http 1889 80 192.168.1.30 192.168.0.40 0 -
116 01/23/1998 16:59:33 00:01:41 telnet 1890 23 192.168.1.30 192.168.0.20 0 -
117 01/23/1998 16:59:36 00:00:02 http 1891 80 192.168.1.30 192.168.0.40 0 -
118 01/23/1998 16:59:36 00:00:12 ftp 1892 21 192.168.1.30 192.168.0.20 0 -
119 01/23/1998 16:59:42 00:00:00 ftp-data 20 1893 192.168.0.20 192.168.1.30 0 -
120 01/23/1998 16:59:45 00:00:01 ftp-data 20 1894 192.168.0.20 192.168.1.30 0 -
121 01/23/1998 16:59:47 00:00:00 ftp-data 20 1895 192.168.0.20 192.168.1.30 0 -
122 01/23/1998 16:59:53 00:00:01 smtp 1900 25 192.168.1.30 192.168.0.20 0 -
123 01/23/1998 16:59:57 00:00:16 ftp 43546 21 192.168.0.40 192.168.1.30 0 -
124 01/23/1998 17:00:01 00:00:00 ftp-data 20 43548 192.168.1.30 192.168.0.40 0 -
125 01/23/1998 17:00:02 00:00:02 rsh 1023 514 192.168.1.30 192.168.0.20 1 rcp
126 01/23/1998 17:00:03 00:00:22 telnet 1906 23 192.168.1.30 192.168.0.20 1 guess
127 01/23/1998 17:00:04 00:00:01 ftp-data 20 43550 192.168.1.30 192.168.0.40 0 -
128 01/23/1998 17:00:05 00:00:14 rlogin 1022 513 192.168.1.30 192.168.0.20 1 rlogin
129 01/23/1998 17:00:07 00:00:00 ftp-data 20 43552 192.168.1.30 192.168.0.40 0 -
130 01/23/1998 17:00:09 00:00:00 ftp-data 20 43554 192.168.1.30 192.168.0.40 0 -
131 01/23/1998 17:00:10 00:00:11 ftp 43555 21 192.168.0.40 192.168.1.30 0 -
132 01/23/1998 17:00:12 00:00:00 ftp-data 20 43558 192.168.1.30 192.168.0.40 0 -
133 01/23/1998 17:00:16 00:00:00 ftp-data 20 43562 192.168.1.30 192.168.0.40 0 -
134 01/23/1998 17:00:18 00:00:00 ftp-data 20 43563 192.168.1.30 192.168.0.40 0 -
135 01/23/1998 17:00:20 00:00:00 ftp-data 20 43564 192.168.1.30 192.168.0.40 0 -
136 01/23/1998 17:00:21 00:00:02 rsh 1022 514 192.168.1.30 192.168.0.20 1 rsh
137 01/23/1998 17:00:22 00:00:01 rsh 1022 1021 192.168.0.20 192.168.1.30 1 rsh
138 01/23/1998 17:00:25 00:00:02 http 1908 80 192.168.1.30 192.168.0.40 0 -
139 01/23/1998 17:00:28 00:00:02 http 1909 80 192.168.1.30 192.168.0.40 0 -
140 01/23/1998 17:00:32 00:00:01 http 1910 80 192.168.1.30 192.168.0.40 0 -
141 01/23/1998 17:00:35 00:00:02 http 1912 80 192.168.1.30 192.168.0.40 0 -
143 01/23/1998 17:00:39 00:00:02 http 1913 80 192.168.1.30 192.168.0.40 0 -
144 01/23/1998 17:00:41 00:00:40 telnet 1914 23 192.168.1.30 192.168.0.20 1 guess
145 01/23/1998 17:00:43 00:00:01 http 1915 80 192.168.1.30 192.168.0.40 0 -
146 01/23/1998 17:00:45 00:00:02 smtp 33017 25 192.168.0.20 192.168.1.30 0 -
147 01/23/1998 17:00:46 00:00:02 http 1916 80 192.168.1.30 192.168.0.40 0 -
148 01/23/1998 17:00:46 00:00:00 auth 1917 113 192.168.1.30 192.168.0.20 0 -
149 01/23/1998 17:00:49 00:00:02 http 1922 80 192.168.1.30 192.168.0.40 0 -
151 01/23/1998 17:00:52 00:00:02 http 1923 80 192.168.1.30 192.168.0.40 0 -
152 01/23/1998 17:00:56 00:00:02 http 1924 80 192.168.1.30 192.168.0.40 0 -
153 01/23/1998 17:01:00 00:00:02 http 1925 80 192.168.1.30 192.168.0.40 0 -
154 01/23/1998 17:01:03 00:00:02 http 1926 80 192.168.1.30 192.168.0.40 0 -
155 01/23/1998 17:01:03 00:01:23 telnet 43570 23 192.168.0.40 192.168.1.30 0 -
157 01/23/1998 17:01:06 00:00:03 http 1927 80 192.168.1.30 192.168.0.40 0 -
158 01/23/1998 17:01:14 00:01:32 telnet 43571 23 192.168.0.40 192.168.1.30 0 -

EK-1 (Devam) DARPA 1998 örnek veri kümesi

159 01/23/1998 17:01:24 00:00:02 http 1931 80 192.168.1.30 192.168.0.40 0 -
160 01/23/1998 17:01:25 00:00:13 ftp 1932 21 192.168.1.30 192.168.0.20 0 -
161 01/23/1998 17:01:25 00:00:02 finger 1933 79 192.168.1.30 192.168.0.20 0 -
162 01/23/1998 17:01:28 00:00:01 http 1934 80 192.168.1.30 192.168.0.40 0 -
163 01/23/1998 17:01:31 00:00:02 http 1936 80 192.168.1.30 192.168.0.40 0 -
164 01/23/1998 17:01:31 00:00:01 ftp-data 20 1937 192.168.0.20 192.168.1.30 0 -
165 01/23/1998 17:01:33 00:00:00 ftp-data 20 1938 192.168.0.20 192.168.1.30 0 -
166 01/23/1998 17:01:34 00:00:01 finger 1939 79 192.168.1.30 192.168.0.20 0 -
167 01/23/1998 17:01:34 00:00:00 ftp-data 20 1940 192.168.0.20 192.168.1.30 0 -
168 01/23/1998 17:01:34 00:00:02 http 1941 80 192.168.1.30 192.168.0.40 0 -
169 01/23/1998 17:01:35 00:00:00 ftp-data 20 1942 192.168.0.20 192.168.1.30 0 -
170 01/23/1998 17:01:36 00:00:01 ftp-data 20 1943 192.168.0.20 192.168.1.30 0 -
171 01/23/1998 17:01:37 00:00:02 http 1944 80 192.168.1.30 192.168.0.40 0 -
172 01/23/1998 17:01:41 00:00:01 http 1945 80 192.168.1.30 192.168.0.40 0 -
173 01/23/1998 17:01:42 00:00:01 finger 1946 79 192.168.1.30 192.168.0.20 0 -
174 01/23/1998 17:01:44 00:00:02 http 1947 80 192.168.1.30 192.168.0.40 0 -
175 01/23/1998 17:01:47 00:00:03 http 43581 80 192.168.0.40 192.168.1.30 0 -
176 01/23/1998 17:01:47 00:00:02 http 1949 80 192.168.1.30 192.168.0.40 0 -
177 01/23/1998 17:01:50 00:00:00 finger 1951 79 192.168.1.30 192.168.0.40 0 -
178 01/23/1998 17:01:50 00:00:02 http 1952 80 192.168.1.30 192.168.0.40 0 -
179 01/23/1998 17:01:51 00:00:02 http 43583 80 192.168.0.40 192.168.1.30 0 -
180 01/23/1998 17:01:52 00:00:02 http 1953 80 192.168.1.30 192.168.0.40 0 -
182 01/23/1998 17:01:54 00:00:02 http 43584 80 192.168.0.40 192.168.1.30 0 -
183 01/23/1998 17:01:55 00:00:02 http 1954 80 192.168.1.30 192.168.0.40 0 -
184 01/23/1998 17:01:56 00:00:00 finger 1956 79 192.168.1.30 192.168.0.40 0 -
185 01/23/1998 17:01:58 00:00:02 http 1957 80 192.168.1.30 192.168.0.40 0 -
186 01/23/1998 17:02:01 00:00:03 http 1958 80 192.168.1.30 192.168.0.40 0 -
187 01/23/1998 17:02:01 00:00:49 telnet 1959 23 192.168.1.30 192.168.0.20 0 -
188 01/23/1998 17:02:02 00:00:00 finger 1961 79 192.168.1.30 192.168.0.40 0 -
189 01/23/1998 17:02:05 00:00:02 http 1962 80 192.168.1.30 192.168.0.40 0 -
190 01/23/1998 17:02:08 00:00:02 http 1963 80 192.168.1.30 192.168.0.40 0 -
191 01/23/1998 17:02:11 00:00:02 http 1964 80 192.168.1.30 192.168.0.40 0 -
192 01/23/1998 17:02:15 00:00:02 http 1965 80 192.168.1.30 192.168.0.40 0 -
193 01/23/1998 17:02:17 00:00:02 http 43588 80 192.168.0.40 192.168.1.30 0 -
194 01/23/1998 17:02:18 00:00:02 http 1966 80 192.168.1.30 192.168.0.40 0 -
195 01/23/1998 17:02:20 00:00:02 http 43589 80 192.168.0.40 192.168.1.30 0 -
196 01/23/1998 17:02:21 00:00:54 telnet 1967 23 192.168.1.30 192.168.0.20 0 -
197 01/23/1998 17:02:21 00:00:04 smtp 43590 25 192.168.0.40 192.168.1.30 0 -
198 01/23/1998 17:02:22 00:00:00 auth 1968 113 192.168.1.30 192.168.0.40 0 -
199 01/23/1998 17:02:25 00:00:02 smtp 1976 25 192.168.1.30 192.168.0.20 0 -
201 01/23/1998 17:02:33 00:00:02 http 43591 80 192.168.0.40 192.168.1.30 0 -
202 01/23/1998 17:02:36 00:00:02 http 43592 80 192.168.0.40 192.168.1.30 0 -
203 01/23/1998 17:02:39 00:00:02 http 43593 80 192.168.0.40 192.168.1.30 0 -
205 01/23/1998 17:02:41 00:00:36 telnet 1978 23 192.168.1.30 192.168.0.20 0 -
206 01/23/1998 17:02:42 00:00:02 http 43594 80 192.168.0.40 192.168.1.30 0 -
207 01/23/1998 17:02:45 00:00:02 http 43595 80 192.168.0.40 192.168.1.30 0 -
208 01/23/1998 17:02:48 00:00:02 http 43596 80 192.168.0.40 192.168.1.30 0 -
209 01/23/1998 17:02:51 00:00:02 http 43597 80 192.168.0.40 192.168.1.30 0 -
210 01/23/1998 17:02:54 00:00:02 http 43599 80 192.168.0.40 192.168.1.30 0 -

EK-1 (Devam) DARPA 1998 örnek veri kümesi

211 01/23/1998 17:02:58 00:00:01 http 1982 80 192.168.1.30 192.168.0.40 0 -
212 01/23/1998 17:02:59 00:00:01 http 1983 80 192.168.1.30 192.168.0.40 0 -
213 01/23/1998 17:02:59 00:00:11 ftp 1984 21 192.168.1.30 192.168.0.20 0 -
214 01/23/1998 17:03:01 00:00:02 http 1985 80 192.168.1.30 192.168.0.40 0 -
215 01/23/1998 17:03:02 00:00:02 http 1986 80 192.168.1.30 192.168.0.40 0 -
216 01/23/1998 17:03:03 00:00:42 telnet 43600 23 192.168.0.40 192.168.1.30 0 -
217 01/23/1998 17:03:04 00:00:00 ftp-data 20 1987 192.168.0.20 192.168.1.30 0 -
218 01/23/1998 17:03:04 00:00:02 http 1988 80 192.168.1.30 192.168.0.40 0 -
219 01/23/1998 17:03:05 00:00:02 http 1989 80 192.168.1.30 192.168.0.40 0 -
220 01/23/1998 17:03:07 00:00:00 ftp-data 20 1990 192.168.0.20 192.168.1.30 0 -
221 01/23/1998 17:03:08 00:00:02 http 1991 80 192.168.1.30 192.168.0.40 0 -
222 01/23/1998 17:03:08 00:00:01 ftp-data 20 1992 192.168.0.20 192.168.1.30 0 -
223 01/23/1998 17:03:09 00:00:03 http 1993 80 192.168.1.30 192.168.0.40 0 -
224 01/23/1998 17:03:11 00:00:15 ftp 43602 21 192.168.0.40 192.168.1.30 0 -
225 01/23/1998 17:03:12 00:00:02 http 1994 80 192.168.1.30 192.168.0.40 0 -
226 01/23/1998 17:03:14 00:00:02 http 1995 80 192.168.1.30 192.168.0.40 0 -
227 01/23/1998 17:03:15 00:00:02 http 1996 80 192.168.1.30 192.168.0.40 0 -
228 01/23/1998 17:03:16 00:00:02 ftp-data 20 43605 192.168.1.30 192.168.0.40 0 -
229 01/23/1998 17:03:17 00:00:02 http 1997 80 192.168.1.30 192.168.0.40 0 -
230 01/23/1998 17:03:18 00:00:01 http 43606 80 192.168.0.40 192.168.1.30 0 -
231 01/23/1998 17:03:18 00:00:02 http 1998 80 192.168.1.30 192.168.0.40 0 -
232 01/23/1998 17:03:20 00:00:01 ftp-data 20 43607 192.168.1.30 192.168.0.40 0 -
233 01/23/1998 17:03:21 00:00:01 http 43608 80 192.168.0.40 192.168.1.30 0 -
234 01/23/1998 17:03:21 00:00:02 http 1999 80 192.168.1.30 192.168.0.40 0 -
235 01/23/1998 17:03:21 00:00:02 http 2000 80 192.168.1.30 192.168.0.40 0 -
236 01/23/1998 17:03:23 00:00:00 ftp-data 20 43609 192.168.1.30 192.168.0.40 0 -
237 01/23/1998 17:03:24 00:00:02 http 43610 80 192.168.0.40 192.168.1.30 0 -
238 01/23/1998 17:03:24 00:00:02 http 2002 80 192.168.1.30 192.168.0.40 0 -
239 01/23/1998 17:03:25 00:00:02 http 2003 80 192.168.1.30 192.168.0.40 0 -
240 01/23/1998 17:03:25 00:00:00 ftp-data 20 43611 192.168.1.30 192.168.0.40 0 -
241 01/23/1998 17:03:28 00:00:02 http 2004 80 192.168.1.30 192.168.0.40 0 -
242 01/23/1998 17:03:28 00:00:02 http 2005 80 192.168.1.30 192.168.0.40 0 -
243 01/23/1998 17:03:28 00:01:18 telnet 43612 23 192.168.0.40 192.168.1.30 0 -
245 01/23/1998 17:03:31 00:00:04 http 2006 80 192.168.1.30 192.168.0.40 0 -
246 01/23/1998 17:03:32 00:00:02 http 2007 80 192.168.1.30 192.168.0.40 0 -
247 01/23/1998 17:03:35 00:00:02 http 2009 80 192.168.1.30 192.168.0.40 0 -
248 01/23/1998 17:03:36 00:00:02 http 2010 80 192.168.1.30 192.168.0.40 0 -
249 01/23/1998 17:03:38 00:00:03 http 2011 80 192.168.1.30 192.168.0.40 0 -
250 01/23/1998 17:03:39 00:00:02 http 2012 80 192.168.1.30 192.168.0.40 0 -
251 01/23/1998 17:03:42 00:00:02 http 43613 80 192.168.0.40 192.168.1.30 0 -
252 01/23/1998 17:03:43 00:00:02 http 2014 80 192.168.1.30 192.168.0.40 0 -
253 01/23/1998 17:03:45 00:00:42 telnet 2016 23 192.168.1.30 192.168.0.20 0 -
254 01/23/1998 17:03:45 00:00:02 http 2017 80 192.168.1.30 192.168.0.40 0 -
255 01/23/1998 17:03:46 00:00:02 http 43614 80 192.168.0.40 192.168.1.30 0 -
256 01/23/1998 17:03:48 00:00:02 http 2018 80 192.168.1.30 192.168.0.40 0 -
258 01/23/1998 17:03:51 00:00:02 http 2019 80 192.168.1.30 192.168.0.40 0 -
259 01/23/1998 17:03:52 00:00:05 telnet 2020 23 192.168.1.30 192.168.0.20 1 port-scan
260 01/23/1998 17:03:52 00:00:00 ssh 2021 22 192.168.1.30 192.168.0.20 1 port-scan
261 01/23/1998 17:03:52 00:00:05 ftp 2022 21 192.168.1.30 192.168.0.20 1 port-scan

EK-1. (Devam) Darpa 1998 Örnek Veriseti

262 01/23/1998 17:03:52 00:00:04 finger 2023 79 192.168.1.30 192.168.0.20 1 port-scan
263 01/23/1998 17:03:52 00:00:00 http 2024 80 192.168.1.30 192.168.0.20 1 port-scan
264 01/23/1998 17:03:52 00:00:00 sunrpc 2025 111 192.168.1.30 192.168.0.20 1 port-scan
265 01/23/1998 17:03:52 00:00:00 pop-3 2026 110 192.168.1.30 192.168.0.20 1 port-scan
266 01/23/1998 17:03:52 00:00:00 pop-2 2028 109 192.168.1.30 192.168.0.20 1 port-scan
267 01/23/1998 17:03:52 00:00:00 lpr 2029 515 192.168.1.30 192.168.0.20 1 port-scan
268 01/23/1998 17:03:52 00:00:06 rsh 2030 514 192.168.1.30 192.168.0.20 1 port-scan
269 01/23/1998 17:03:52 00:00:06 rlogin 2031 513 192.168.1.30 192.168.0.20 1 port-scan
270 01/23/1998 17:03:52 00:00:06 exec 2032 512 192.168.1.30 192.168.0.20 1 port-scan
271 01/23/1998 17:03:52 00:00:00 nfsd 2033 2049 192.168.1.30 192.168.0.20 1 port-scan
272 01/23/1998 17:03:53 00:00:00 unknown 2034 3000 192.168.1.30 192.168.0.20 1 port-scan
273 01/23/1998 17:03:53 00:00:00 x-server 2035 6000 192.168.1.30 192.168.0.20 1 port-scan
274 01/23/1998 17:03:54 00:00:00 telnet 2037 23 192.168.1.30 192.168.0.40 1 port-scan
275 01/23/1998 17:03:54 00:00:00 ssh 2038 22 192.168.1.30 192.168.0.40 1 port-scan
276 01/23/1998 17:03:54 00:00:00 ftp 2039 21 192.168.1.30 192.168.0.40 1 port-scan
277 01/23/1998 17:03:54 00:00:00 finger 2040 79 192.168.1.30 192.168.0.40 1 port-scan
278 01/23/1998 17:03:54 00:00:00 http 2041 80 192.168.1.30 192.168.0.40 1 port-scan
279 01/23/1998 17:03:54 00:00:00 sunrpc 2042 111 192.168.1.30 192.168.0.40 1 port-scan
280 01/23/1998 17:03:54 00:00:00 pop-3 2043 110 192.168.1.30 192.168.0.40 1 port-scan
281 01/23/1998 17:03:54 00:00:00 pop-2 2044 109 192.168.1.30 192.168.0.40 1 port-scan
282 01/23/1998 17:03:54 00:00:00 lpr 2045 515 192.168.1.30 192.168.0.40 1 port-scan
283 01/23/1998 17:03:54 00:00:00 rsh 2046 514 192.168.1.30 192.168.0.40 1 port-scan
284 01/23/1998 17:03:54 00:00:00 rlogin 2047 513 192.168.1.30 192.168.0.40 1 port-scan
285 01/23/1998 17:03:54 00:00:00 exec 2048 512 192.168.1.30 192.168.0.40 1 port-scan
286 01/23/1998 17:03:54 00:00:00 nfsd 2052 2049 192.168.1.30 192.168.0.40 1 port-scan
287 01/23/1998 17:03:54 00:00:00 unknown 1029 3000 192.168.1.30 192.168.0.40 1 port-scan
288 01/23/1998 17:03:54 00:00:00 x-server 1030 6000 192.168.1.30 192.168.0.40 1 port-scan
289 01/23/1998 17:03:55 00:00:01 http 1031 80 192.168.1.30 192.168.0.40 0 -
290 01/23/1998 17:03:58 00:00:03 http 1032 80 192.168.1.30 192.168.0.40 0 -
291 01/23/1998 17:03:58 00:00:02 http 1033 80 192.168.1.30 192.168.0.40 0 -
292 01/23/1998 17:03:59 00:00:02 http 43617 80 192.168.0.40 192.168.1.30 0 -
293 01/23/1998 17:04:02 00:00:02 http 1034 80 192.168.1.30 192.168.0.40 0 -
294 01/23/1998 17:04:02 00:00:02 http 43618 80 192.168.0.40 192.168.1.30 0 -
295 01/23/1998 17:04:03 00:00:02 http 1035 80 192.168.1.30 192.168.0.40 0 -
296 01/23/1998 17:04:05 00:00:02 http 43619 80 192.168.0.40 192.168.1.30 0 -
297 01/23/1998 17:04:06 00:00:01 http 1036 80 192.168.1.30 192.168.0.40 0 -
298 01/23/1998 17:04:06 00:00:02 http 1037 80 192.168.1.30 192.168.0.40 0 -
299 01/23/1998 17:04:08 00:00:02 http 43620 80 192.168.0.40 192.168.1.30 0 -
300 01/23/1998 17:04:09 00:00:01 http 1038 80 192.168.1.30 192.168.0.40 0 -
301 01/23/1998 17:04:11 00:00:02 http 43621 80 192.168.0.40 192.168.1.30 0 -
302 01/23/1998 17:04:12 00:00:02 http 1039 80 192.168.1.30 192.168.0.40 0 -
303 01/23/1998 17:04:14 00:00:02 http 43622 80 192.168.0.40 192.168.1.30 0 -
304 01/23/1998 17:04:15 00:00:02 http 1040 80 192.168.1.30 192.168.0.40 0 -
305 01/23/1998 17:04:17 00:00:01 http 43623 80 192.168.0.40 192.168.1.30 0 -
306 01/23/1998 17:04:18 00:00:02 http 1041 80 192.168.1.30 192.168.0.40 0 -
307 01/23/1998 17:04:20 00:00:01 http 43624 80 192.168.0.40 192.168.1.30 0 -
308 01/23/1998 17:04:31 00:00:38 telnet 1042 23 192.168.1.30 192.168.0.20 0 -

EK-1. (Devam) DARPA 1998 örnek veri kümesi

309 01/23/1998 17:04:33 00:01:19 telnet 43625 23 192.168.0.40 192.168.1.30 0 -
310 01/23/1998 17:04:53 00:00:01 smtp 1048 25 192.168.1.30 192.168.0.20 0 -
311 01/23/1998 17:05:23 00:00:01 finger 1050 79 192.168.1.30 192.168.0.20 0 -

EK-2 Saldırılar ve açıklamaları

Back	Denial of service attack against apache webserver where a client requests a URL containing many backslashes.
Dict	Guess passwords for a valid user using simple variants of the account name over a telnet connection.
Eject	Buffer overflow using eject program on Solaris. Leads to a user->root transition if successful.
Ffb	Buffer overflow using the ffbconfig UNIX system command leads to root shell
Format	Buffer overflow using the fdformat UNIX system command leads to root shell
ftp-write	Remote FTP user creates .rhost file in world writable anonymous FTP directory and obtains local login.
Guest	Try to guess password via telnet for guest account.
Imap	Remote buffer overflow using imap port leads to root shell
Ipsweep	Surveillance sweep performing either a port sweep or ping on multiple host addresses.
Land	Denial of service where a remote host is sent a UDP packet with the same source and destination
Loadmodule	Non-stealthy loadmodule attack which resets IFS for a normal user and creates a root shell
Multihop	Multi-day scenario in which a user first breaks into one machine
Neptune	Syn flood denial of service on one or more ports.
Nmap	Network mapping using the nmap tool. Mode of exploring network will vary--options include SYN
Perlmagic	Perl attack which sets the user id to root in a perl script and creates a root shell
Phf	Exploitable CGI script which allows a client to execute arbitrary commands on a machine with a misconfigured web server.
Pod	Denial of service ping of death
Portsweep	Surveillance sweep through many ports to determine which services are supported on a single host.
Rootkit	Multi-day scenario where a user installs one or more components of a rootkit
Satan	Network probing tool which looks for well-known weaknesses. Operates at three different levels. Level 0 is light
Smurf	Denial of service icmp echo reply flood.
Spy	Multi-day scenario in which a user breaks into a machine with the purpose of finding important information where the user tries to avoid detection. Uses several different exploit methods to gain access.
Syslog	Denial of service for the syslog service connects to port 514 with unresolvable source ip.
Teardrop	Denial of service where mis-fragmented UDP packets cause some systems to reboot.
Warez	User logs into anonymous FTP site and creates a hidden directory.
Warezclient	Users downloading illegal software which was previously posted via anonymous FTP by the warezmaster.
Warezmaster	Anonymous FTP upload of Warez (usually illegal copies of copywrited software) onto FTP server.

EK-3 Simülasyon ortamındaki bilgisayar ve sunucuların isimleri ve IP adresleri

Outside Hosts

IP Address	Hostname	Operating System	Notes
135.13.216.191	alpha.apple.edu	Redhat 5.0	kernel 2.0.32
135.8.60.182	beta.banana.edu	Solaris 2.5.1	
194.27.251.21	gamma.grape.mil	SunOS 4.1.4	
194.7.248.153	delta.peach.mil	Redhat 5.0	kernel 2.0.32
195.115.218.108	epsilon.pear.com	Solaris 2.5.1	
195.73.151.50	lambda.orange.com	SunOS 4.1.4	
196.37.75.158	jupiter.cherry.org	Redhat 5.0	kernel 2.0.32
196.227.33.189	saturn.kiwi.org	Solaris 2.5.1	
197.182.91.233	mars.avocado.net	SunOS 4.1.4	
197.218.177.69	pluto.plum.net	Redhat 5.0	kernel 2.0.32
192.168.1.30	monitor	MacOS	SNMP monitor
192.168.1.10	calvin.world.net		Outside gateway
192.168.1.20	aesop.world.net		Outside Web Server

Routers & Hubs

IP Address	Hostname	Operating System	Notes
172.16.112.1	loud.world.net		Cisco 2514 Router
192.168.1.1	loud.world.net		Cisco 2514 Router
172.16.112.5	None.		Hewlett-Packard EtherTwist Hub
192.168.1.2	None.		Hewlett-Packard EtherTwist Hub

Inside Hosts

IP Address	Hostname	Operating System	Notes
172.16.12.10	plato.eyrie.af.mil	Solaris	Not part of simulation
172.16.112.10	locke.eyrie.af.mil	SunOS	inside sniffer
172.16.112.20	hobbes.eyrie.af.mil	Redhat 5.0	Inside gateway
172.16.112.50	pascal.eyrie.af.mil	Solaris 2.5.1	
172.16.112.100	hume.eyrie.af.mil	Windows NT 4.0	
172.16.112.149	eagle.eyrie.af.mil	Redhat 5.0	kernel 2.0.32
172.16.112.194	falcon.eyrie.af.mil	Solaris 2.5.1	

EK-3 (Devam) Simülasyon ortamındaki bilgisayar ve sunucuların isimleri ve ip adresleri

172.16.112.207	robin.eyrie.af.mil	SunOS 4.1.4	
172.16.113.50	zeno.eyrie.af.mil	SunOS 4.1.4	
172.16.113.84	duck.eyrie.af.mil	SunOS 4.1.4	
172.16.113.105	swallow.eyrie.af.mil	Redhat 5.0	kernel 2.0.32
172.16.113.204	goose.eyrie.af.mil	Solaris 2.5.1	
172.16.114.50	marx.eyrie.af.mil	Redhat 4.0	kernel 2.0.27
172.16.114.148	crow.eyrie.af.mil	Redhat 5.0	kernel 2.0.32
172.16.114.168	finch.eyrie.af.mil	SunOS 4.1.4	
172.16.114.169	swan.eyrie.af.mil	Solaris 2.5.1	
172.16.114.207	pigeon.eyrie.af.mil	Redhat 5.0	kernel 2.0.32
172.16.115.5	pc1.eyrie.af.mil	Windows 95	
172.16.115.87	pc2.eyrie.af.mil	Windows 95	
172.16.115.234	pc0.eyrie.af.mil	Window NT 4.0	Build 1381, Service Pack 1
172.16.116.44	pc5.eyrie.af.mil	Windows 3.1	
172.16.116.194	pc3.eyrie.af.mil	Windows 95	
172.16.116.201	pc4.eyrie.af.mil	Windows 95	
172.16.117.52	pc7.eyrie.af.mil	Windows 3.1	
172.16.117.103	pc9.eyrie.af.mil	MacOS	
172.16.117.111	pc8.eyrie.af.mil	MacOS	
172.16.117.132	pc6.eyrie.af.mil	Windows 3.1	
172.16.118.10	linux1.eyrie.af.mil	Redhat 5.0	kernel 2.0.32
172.16.118.20	linux2.eyrie.af.mil	Redhat 5.0	kernel 2.0.32
172.16.118.30	linux3.eyrie.af.mil	Redhat 5.0	kernel 2.0.32
172.16.118.40	linux4.eyrie.af.mil	Redhat 5.0	kernel 2.0.32
172.16.118.50	linux5.eyrie.af.mil	Redhat 5.0	kernel 2.0.32
172.16.118.60	linux6.eyrie.af.mil	Redhat 5.0	kernel 2.0.32
172.16.118.70	linux7.eyrie.af.mil	Redhat 5.0	kernel 2.0.32
172.16.118.80	linux8.eyrie.af.mil	Redhat 5.0	kernel 2.0.32
172.16.118.90	linux9.eyrie.af.mil	Redhat 5.0	kernel 2.0.32
172.16.118.100	linux10.eyrie.af.mil	Redhat 5.0	kernel 2.0.32

EK-4 Saldırı takvimi

Hafta	Gün	Atak Adı	Zaman	Kaynak Makina	Hedef Makina	Kullanıcı	Veri	Varyant
1	Mon	format	08:05:07	135.8.60.182	pascal	tristank	tcp, bsm	clear
1	Mon	ffb	08:07:13	135.8.60.182	pascal	tristank	tcp,bsm	clear
1	Tues	loadmodule	10:12:06	135.8.60.182	zeno	tristank	tcp	clear
1	Tues	perlmagic	20:19:49	135.8.60.182	marx	tristank	tcp	clear
1	Wed	smurf	08:12:16	*	marx	-	tcp	-
1	Wed	neptune	11:55:15	1.2.3.4	pascal	-	tcp	-
1	Thurs	pod	09:06:03	1.2.3.4	marx	-	tcp	-
1	Thurs	dict	11:00:10	135.8.60.182	zeno	alie	tcp	-
1	Fri	teardrop	10:11:55	172.16.112.20	marx	-	tcp	-
2	Mon	guest	15:34:10	135.13.216.191	marx	tristank	tcp	-
2	Mon	portsweep	17:27:57	192.168.1.10	marx	-	tcp	fast,ports 1-100
2	Tues	ipsweep	19:00:31	135.13.216.191	*	-	tcp,bsm	-
2	Wed	land	09:32:03	zeno	zeno	-	tcp	-
2	Fri	ftp-write	09:10:47	195.73.151.50	pascal	ftp	tcp,bsm	-
2	Fri	imap	12:55:28	195.73.151.50	marx	-	tcp	-
2	Fri	back	13:55:30	135.8.60.182	marx	-	tcp	-
2	Fri	syslog	22:20:29	10.0.1.20	pascal	-	tcp	-
3	Mon	satan	09:36:59	1,52169E+11	pascal	-	tcp,bsm	level1
3	Mon	phf	11:14:34	197.218.177.69	marx	-	tcp	-
3	Mon	ffb	11:32:20	202.247.224.89	pascal	clintonl	tcp,bsm	clear
3	Mon	portsweep	19:28:06	207.75.239.115	marx	-	tcp	many,ramp
3	Wed	nmap	09:03:21	202.72.1.77	pascal	-	tcp	slow(12hr), pascal,-U,stealthy
3	Wed	ftp-write	09:19:29	206.48.44.18	pascal	ftp	tcp,bsm	-
3	Wed	smurf	11:55:57	*	pascal	-	tcp	-
3	Wed	ipsweep	12:31:00	202.77.162.213	*	-	tcp	fast,ping
3	Wed	multihop	14:22:31	206.229.221.82	marx	guest	tcp	guest,runs crack
3	Wed	back	16:18:44	202.77.162.213	marx	-	tcp	-
3	Thurs	multihop	09:25:58	206.229.221.82	zeno,marx	darleent	tcp	perlmagic
3	Thurs	neptune	15:10:26	10.20.30.40	pascal	-	tcp	all ports,1hr
3	Thurs	warez	23:12:17	192.168.1.10	pascal	ftp	tcp,bsm	-
3	Fri	imap	08:16:42	202.49.244.10	marx	-	tcp	-
3	Fri	imap	08:49:21	202.77.162.213	marx	-	tcp	-
3	Fri	nmap	14:18:25	208.240.124.83	172.16.112.*	-	tcp	echo flood
3	Fri	warezmaster	19:01:31	206.186.80.111	pascal	ftp	tcp,bsm	-
3	Fri	land	22:56:41	src.same.as.dst	several	-	tcp	-
4	Mon	warezclient	08:05:15	all.attackers	pascal	ftp	tcp,bsm	-
4	Mon	pod	11:23:24	207.103.80.104	pascal	-	tcp	-
4	Mon	rootkit	13:03:38	207.230.54.203	marx	imap/v0m	tcp	imap,trojan login: stage 1
4	Mon	smurf	14:03:35	199.174.194.*	marx	-	tcp	1.75 min
4	Tues	warezclient	08:05:11	all.attackers	pascal	ftp	tcp,bsm	-
4	Tues	satan	08:50:13	192.168.1.10	zeno	-	tcp	level2
4	Tues	neptune	11:55:38	9.9.9.9	pascal	-	tcp	ports 20,23,79,80 1hr
4	Tues	rootkit	12:17:38	207.230.54.203	marx	v0z	tcp	compile sniffer: stage 2

EK-4 (Devam) Saldırı takvimi

Hafta	Gün	Atak Adı	Zaman	Kaynak Makina	Hedef Makina	Kullanıcı	Veri	Varyant
4	Tues	spy	15:37:41	2,08254E+11	pascal	huws	tcp,bsm	adds user rsmith: stage 2
4	Tues	pod	20:11:31	207.103.80.104	allpc,vlinux	-	tcp	-
4	Tues	teardrop	23:15:08	2,22222E+11	marx	-	tcp	-
4	Wed	pod	09:11:58	207.103.80.104	allvin	-	tcp	-
4	Wed	ipsweep	16:32:26	197.182.91.233	172.16.114.*	-	tcp	udp scan -- nmap -u -n -p53
4	Wed	portsweep	20:00:29	194.27.251.21	marx	-	tcp	fast,0-300 -- scantcp 0
4	Thurs	warezclient	08:05:07	all.attackers	pascal	ftp	tcp,bsm	
4	Thurs	syslog	09:52:16	1.1.1.1	pascal	-	tcp	disable syslog
4	Thurs	portsweep	15:15:18	194.7.248.153	zeno	-	tcp	ports 1-2000 one every 20 sec,scantcp 2
4	Fri	warezclient	08:05:05	all.attackers	pascal	ftp	tcp,bsm	
4	Fri	rootkit	09:11:02	207.230.54.203	marx	v0z	tcp	put sniffer in rc.local: stage 3
4	Fri	ffb	09:22:12	135.13.216.191	pascal	jaroslan	tcp,bsm	add .rhosts, stealthy
4	Fri	format	10:55:44	192.168.1.10	pascal	alie	tcp,bsm	stealthy
4	Fri	multihop	13:00:20	206.229.221.82	zeno,marx	darleen7	tcp	Start up tftp on 6543 using suid rootshell: stage 3
4	Fri	loadmodule	14:10:11	135.13.216.191	pascal	jaroslan	tcp,bsm	ftp loadmodule, and run, failed
4	Fri	ipsweep	15:10:18	197.218.177.69	172.16.112.*	-	tcp	nmap -P 172.16.112.*
5	Mon	teardrop	08:15:02	1.1.1.1	pascal	-	tcp	port23,100packets
5	Mon	satan	10:11:57	209.154.98.104	marx	-	tcp	level 2
5	Mon	smurf	12:53:15	252.169.215, 202.77.162	marx	-	tcp	~5 minutes
5	Mon	ffb	14:10:20	135.13.216.191	pascal	jaroslan	tcp,bsm	ftp's over exploit files
5	Mon	smurf	15:33:28	all.outside	marx	-	tcp	~25 minutes
5	Mon	ffb	16:22:25	135.13.216.191	pascal	jaroslan	bsm	chmods files. No sniffing
5	Mon	ffb	17:47:29	135.13.216.191	pascal	jaroslan	bsm	executes attack No sniffing
5	Mon	format	20:14:14	139.134.61.42	pascal	tristank	bsm	clear No Sniffing
5	Tues	ipsweep	10:11:45	196.37.75.158	172.16.112-114.*	-	tcp	-
5	Tues	eject	14:43:08	209.12.13.144	pascal	raeburnt	tcp,bsm	clear
5	Tues	eject	16:39:11	209.17.189.98	pascal	alie	tcp,bsm	clear
5	Tues	portsweep	20:02:12	200.27.121.118	pascal	-	tcp	uses SYNs
5	Tues	perlmagic	20:13:16	1,95115E+11	marx	raeburnt	tcp	-
5	Tues	pod	21:30:11	207.103.80.104	all.vin	-	tcp	-
5	Wed	syslog	11:11:20	197.218.177.69	pascal	-	tcp	-
5	Wed	teardrop	12:46:20	1,11111E+11	marx	-	tcp	-
5	Wed	eject	22:42:37	197.182.91.233	pascal	alie	tcp,bsm	stealthy
5	Wed	ipsweep	23:46:35	196.37.75.158	all	-	tcp	one ping every 3 sec
5	Thurs	pod	08:10:06	209.30.71.165	all.pcs.and.linux	-	tcp	1 packet
5	Thurs	smurf	08:16:59	152.168.215, 202.77.162	marx	-	tcp	5 mins

EK-4 (Devam) Saldırı takvimi

Hafta	Gün	Atak Adı	Zaman	Kaynak Makina	Hedef Makina	Kullanıcı	Veri	Varyant
5	Thurs	pod	09:41:00	207.103.80.104	pascal	-	tcp,	10 pings
5	Thurs	teardrop	09:48:16	2,22222E+11	marx	-	tcp	port 23
5	Thurs	neptune	10:05:00	10.20.30.40	pascal	-	tcp	all port`s 1 hr
5	Thurs	land	10:06:14	pascal	pascal	-	tcp	port 79
5	Thurs	teardrop	10:09:56	1.1.1.1	pascal	-	tcp	port 23
5	Thurs	pod	10:16:17	199.174.194.16	allpcsandlinux	-	tcp	1 ping each
5	Thurs	teardrop	10:36:48	1,23123E+11	zeno	-	tcp	-
5	Thurs	satan	10:42:53	208.240.124.83	zeno	-	tcp	level0
5	Thurs	portsweep	11:23:12	207.136.86.223	pascal	-	tcp	1 SYN every 1 min
5	Thurs	portsweep	11:25:11	196.227.33.189	zeno	-	tcp	1 ACK every 20 sec
5	Thurs	neptune	11:42:31	9.9.9.9	pascal	-	tcp	ports 20,23,79,80 for 1 hr
5	Thurs	smurf	11:57:45	all.attackers	marx	-	tcp	about 50 mins
5	Thurs	teardrop	12:05:56	1.1.1.1	pascal	-	tcp	-
5	Fri	format	08:50:38	199.227.99.125	pascal	lucyj	tcp,bsm	ftp over files, Stage
5	Fri	loadmodule	11:12:16	197.218.177.69	zeno	wardc	tcp	clear
5	Fri	portsweep	11:46:39	2,0516E+11	zeno	-	tcp	port 1-100, one every 3 min, FIN scan
5	Fri	eject	12:34:29	206.48.44.18	pascal	bramy	tcp,bsm	run self contained exploit, more the uuencoded results
5	Fri	format	13:07:22	199.227.99.125	pascal	lucyj	tcp,bsm	chmod exploit files, Stage 2
5	Fri	neptune	17:27:07	230.1.10.20	pascal	-	tcp	port 1-1024 every 5 min for 1 hr
5	Fri	smurf	18:00:15	8subnets	linux3	-	tcp	800,000 ICMP replies
5	Fri	rootkit	22:53:38	207.230.54.203	marx	-	tcp	tftp sniffer logs out, Stage 4
6	Mon	phf	13:02:04	135.8.60.182	marx	-	tcp	-
6	Mon	satan	17:29:28	202.72.1.77	zeno	-	tcp	level 0
6	Mon	neptune	19:19:17	9.9.9.9	pascal	-	tcp	port 20,23,79,80
6	Tues	portsweep	08:16:51	206.48.44.18	pascal	-	tcp	ACK, every 38 secs, ports 1-2000
6	Tues	pod	13:04:56	207.103.80.104	marx	-	tcp	10 oversize ping packets
6	Tues	land	17:53:49	zeno	zeno	-	tcp	port 23
6	Wed	ipsweep	08:29:08	209.30.70.14	marx	-	tcp	nmap -u -n -p53 (udp scan for named)
6	Wed	neptune	10:41:42	135.13.216.191	zeno	-	tcp	all ports for an hour
6	Wed	back	14:11:52	135.8.60.182	marx	-	tcp	-
6	Thurs	ipsweep	08:27:03	205.231.28.163	172.16.114.*	-	tcp	
6	Thurs	ipsweep	08:28:43	196.37.75.158	172.16.112.*	-	tcp	
6	Thurs	eject	08:41:50	202.247.224.89	pascal	raeburnt	tcp	
6	Thurs	ffb	09:06:46	199.174.194.16	pascal	alie	tcp,bsm	
6	Thurs	eject	09:32:03	135.8.60.182	pascal	alie	tcp,bsm	

EK-4 (Devam) Saldırı takvimi

Hafta	Gün	Atak Adı	Zaman	Kaynak Makina	Hedef Makina	Kullanıcı	Veri	Varyant
6	Thurs	eject	09:50:46	195.73.151.50	pascal	alie	tcp,bsm	
6	Thurs	eject	10:00:14	135.8.60.182	pascal	alie	tcp,bsm	
6	Thurs	pod	10:11:06	135.13.216.191	pc0	-	tcp	
6	Thurs	pod	10:20:11	209.30.71.165	linux10	-	tcp	
6	Thurs	pod	10:27:24	207.103.80.104	pascal	-	tcp	
6	Thurs	dict	10:34:46	206.186.80.111	marx	kiaraa	tcp	
6	Thurs	ipsweep	10:37:42	202.72.1.77	172.16.112.*	-	tcp	
6	Thurs	phf	11:15:53	209.74.60.168	marx	-	tcp	
6	Thurs	neptune	11:32:23	230.1.10.20	pascal	-	tcp	
6	Thurs	portsweep	12:03:45	202.247.224.89	zeno	-	tcp	
6	Thurs	eject	12:21:55	209.12.13.144	pascal	raeburnt	tcp,bsm	
6	Thurs	portsweep	12:29:51	207.103.80.104	marx	-	tcp	
6	Thurs	smurf	12:48:13	*	marx	-	tcp	
6	Thurs	land	13:31:05	*	*	-	tcp	
6	Thurs	neptune	13:31:08	10.20.30.40	pascal	-	tcp	
6	Thurs	teardrop	13:30:00	2,22222E+11	marx	-	tcp	
6	Thurs	satan	13:57:45	1,95115E+11	marx	-	tcp	
6	Thurs	ipsweep	14:10:09	197.218.177.69	172.16.112.*	-	tcp	
6	Thurs	eject	14:14:56	199.227.99.125	pascal	raeburnt	tcp,bsm	
6	Thurs	portsweep	14:41:47	206.48.44.18	pascal	-	tcp	
6	Thurs	ffb	14:43:31	209.154.98.104	pascal	alie	tcp,bsm	
6	Thurs	ipsweep	15:08:20	209.1.12.46	172.16.112.*	-	tcp	
6	Thurs	land	15:08:42	*	*	-	tcp	
6	Thurs	teardrop	15:23:47	1.1.1.1	pascal	-	tcp	
6	Thurs	pod	16:15:20	207.75.239.115	linux10	-	tcp	
6	Thurs	pod	16:35:20	197.182.91.233	pc0	-	tcp	
6	Thurs	perlmagic	16:47:24	196.37.75.158	marx	raeburnt	tcp	
6	Thurs	satan	16:57:23	128.223.199.68	marx	-	tcp	
6	Thurs	perlmagic	17:02:54	209.74.60.168	marx	raeburnt	tcp	
6	Thurs	eject	17:50:09	207.253.84.13	pascal	alie	tcp,bsm	
6	Thurs	smurf	17:53:26	*	marx	-	tcp	
6	Thurs	eject	19:50:15	202.49.244.10	pascal	raeburnt	bsm	no sniffing
6	Thurs	ffb	20:30:59	2,08254E+11	pascal	alie	bsm	no sniffing
6	Thurs	eject	20:39:41	206.222.3.197	pascal	darleent	bsm	no sniffing
6	Thurs	eject	20:47:35	2,09117E+11	pascal	alie	bsm	no sniffing
6	Thurs	eject	23:43:29	202.72.1.77	pascal	raeburnt	bsm	no sniffing
6	Fri	teardrop	08:32:12	2.11.32.45	marx	-	tcp	crashed machine
6	Fri	neptune	09:31:52	10.20.30.40	pascal	-	tcp	ports 23,25 for 1 hr
6	Fri	smurf	19:12:37	all.attackers	marx	-	tcp	2 hrs
7	Mon	satan	08:04:25	207.230.54.203	marx	-	tcp	
7	Mon	syslog	12:42:51	1.1.1.1	pascal	-	tcp	
7	Mon	phf	15:53:54	197.182.91.233	marx	-	tcp	
7	Mon	land	18:48:49	several	several	-	tcp	
7	Tues	portsweep	13:05:46	194.27.251.21	marx	-	tcp	
7	Tues	pod	16:34:52	207.103.80.104	marx	-	tcp	

EK-4 (Devam) Saldırı takvimi

Hafta	Gün	Atak Adı	Zaman	Kaynak Makina	Hedef Makina	Kullanıcı	Veri	Varyant
7	Tues	ffb	17:10:01	206.47.98.151	pascal	alie	tcp	No bsm
7	Tues	eject	19:16:45	209.74.60.168	pascal	raeburnt	tcp	No bsm
7	Wed	phf	08:56:59	205.180.112.36	marx	-	tcp	
7	Wed	loadmodule	09:41:51	209.12.13.144	zeno	simple1	tcp	
7	Wed	teardrop	12:53:40	2,22222E+11	marx	-	tcp	
7	Wed	ipsweep	18:01:10	207.114.237.57	172.16.114.*	-	tcp	
7	Wed	portsweep	23:10:42	128.223.199.68	zeno	-	tcp	
7	Thurs	smurf	17:46:58	*	marx	-	tcp	
7	Thurs	satan	21:40:40	207.75.239.115	marx	-	tcp	
7	Thurs	perlmagic	22:09:33	135.8.60.182	marx	alie	tcp	
7	Thurs	ipsweep	23:53:54	153.37.134.17	172.16.112.*	-	tcp	
7	Fri	neptune	09:48:11	10.20.30.40	pascal	-	tcp	
7	Fri	smurf	17:16:23	10.different	pascal	-	tcp	
7	Fri	neptune	19:21:20	9.9.9.9	pascal	-	tcp	
7	Fri	back	22:51:38	207.75.239.115	marx	-	tcp	

EK-5 Anormallikler ve kullanıcılar

Hafta-Gün	Kullanıcı	Kullanıcı Adı	Başlangıç Zamanı	Hedef	Anormallik Tanımı
6-0	programmer2	franko	14:57:10	beta	pascal logs in from beta
6-0	secretary	georgeb	20:03:39	alpha	pascal logs in at night
6-1	sysadm	janes	08:29:48	jupiter	pascal logs in from jupiter
6-1	programmer1	fredd	08:52:15	alpha	pascal becomes a secretary
6-2	secretary	georgeb	08:16:01	alpha	pascal becomes a manager
6-2	programmer1	fredd	21:09:24	alpha	pascal logs in at night
6-3	sysadm	janes	08:54:53	alpha	pascal becomes a programmer
6-3	manager1	williamf	23:59:11	alpha	pascal logs in at midnight (see note)
6-4	manager1	williamf	08:11:12	alpha	pascal becomes a sysadm
6-4	manager2	donaldh	08:42:52	pluto	pascal logs in from pluto

sysadm: A system administrator named Jane Steinberg who logs in as root, cats the password file and runs commands such as top.

programmer1: A programmer named Fred Dunmeyer who writes public domain C code using a vi editor, compiles the C code (sometimes successfully), reads and sends mail, and executes unix commands.

programmer2: Another programmer named Frank Orlando with a similar user profile, except that he works afternoons and evenings.

secretary: A secretary named George Belliard who edits latex files, runs latex, reads mail, and sends mail.

manager1: A manager named William Finchley who reads and sends mail

manager2: A manager named Donald Hershey who reads mail.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : GÜVEN, Esra Nergis

Uyruğu : T.C.

Doğum tarihi ve yeri : 10.01.1981, Kırıkkale

Medeni hali : Bekar

Telefon : 0 (312) 231 74 00

Faks : 0 (312) 230 65 03

e-mail : eng@gazi.edu.tr

Eğitim

Derece	**Eğitim Birimi**	**Mezuniyet tarihi**
Yüksek lisans	Gazi Üniversitesi / Bilgisayar Müh. Bölümü	2007
Lisans	Erciyes Üniversitesi/ Bilgisayar Müh. Bölümü	2004
Lise	Afyon Kocatepe Anadolu Lisesi	1999

İş Deneyimi

Yıl	**Yer**	**Görev**
2005-2007	Gazi Üniversitesi	Araştırma Görevlisi

Yabancı Dil

İngilizce

Yayınlar

Güven, E.N., Onur, H., Sağıroğlu, Ş., "Yapay Sinir Ağları ile Web İçeriklerini Sınıflandırma", ***Değişen Dünyada Bilgi Yönetimi Sempozyumu,*** ANKARA, 42-49 (2007).

Sağıroğlu, Ş., Yavanoğlu, U., Güven, E.N., "Web based Machine Learning for Language Identification and Translation" ***International Conference on Machine Learning and Applications (ICMLA 2007),*** Ohio, Cincinnati (2007).