BAKANLIKTA KULLANILAN LOG SİSTEMLERİNİN MERKEZİLEŞTİRİLMESİ VE YÖNETİMİ

- UZMANLIK TEZİ -

HAZIRLAYAN: SAMET CAN

ANKARA-2017

T.C.
ÇEVRE VE ŞEHİRCİLİK BAKANLIĞI

BAKANLIKTA KULLANILAN LOG SİSTEMLERİNİN MERKEZİLEŞTİRİLMESİ VE YÖNETİMİ

Tez Hazırlayanın Adı Soyadı : Samet CAN
Tez Danışmanının Adı Soyadı : Kudret Aslı BABACAN
Birim Amirinin Adı Soyadı : Ömer ALAN

Samet CAN tarafından hazırlanan Bakanlıkta Kullanılan Log Sistemlerinin Merkezileştirilmesi ve Yönetimi adlı bu tezin Çevre ve Şehircilik Uzmanlık tezi olarak uygun olduğunu onaylarım.

Çevre ve Şehircilik Uzmanı, Kudret Aslı, BABACAN
Tez Danışmanı

Bu çalışma, tez savunma komisyonumuz tarafından Çevre ve Şehircilik Uzmanlık tezi olarak kabul edilmiştir.

Başkan : Genel Müdür V., Ömer, ALAN

Üye : Genel Müdür Yrd. V., Mustafa Yavuz, TORUN

Üye : Daire Başkanı V., İskender, ERMİŞ

Üye : Daire Başkanı V., Sibel, ASLAN

Üye : Çevre ve Şehircilik Uzmanı, Kudret Aslı, BABACAN

Bu tez, Çevre ve Şehircilik Uzmanlığı Tez Hazırlama Yönergesi' ne uygundur.

İÇİNDEKİLER

ÖZET

ÇEVRE VE ŞEHİRCİLİK BAKANLIĞI	
Tezin Adı	Bakanlıkta Kullanılan Log Sistemlerinin Merkezileştirilmesi ve Yönetimi
Türü	Çevre ve Şehircilik Bakanlığı Uzmanlık Tezi
Yazar	Samet CAN
Teslim Tarihi	10/05/2017
Anahtar Kelimeler	Log, Olay, Yönetim
Tez Danışmanı	Kudret Aslı BABACAN
Sayfa Adedi	152

Özet

Günümüzde artan siber saldırılar ve uyumluluk gereksinimleri nedeniyle kurumlar, bilgi güvenliği ve olay yönetimi konusunda gerekli çalışmaları yapmaktadır. Bu çalışmaları yaparken ihtiyaç duydukları bir kavram da log yönetimidir. Logların merkezileştirilmesi ve yönetiminin sağlanabilmesi için loglama türleri, log biçimleri, log kaynakları, log toplama mekanizmaları, log saklama stratejileri, log analizi, loglama kuralları ve uyumluluk gereksinimleri gibi konularda temel bilgi sahibi olmak gereklidir. Logların belirli standartlarda oluşturularak toplanması, kurumun ihtiyaçlarına, uyumluluk gereksinimlerine ve düzenleyici kuruluşlara göre belirlenir.

Bu tez çalışması kapsamında logların merkezileştirilmesi ve yönetimi ile birlikte dolaylı olarak bilgi güvenliği ve olay yönetimi konusunda farkındalık oluşturmak, kurumda yapılan çalışmalara ışık tutmak, bu çalışmalardaki eksiklerin belirlenmesine yardımcı olarak loglama ile ilgili ne tür eylemlerin alınması gerektiği konusunda bilgi sağlamak amaçlanmaktadır.

ABSTRACT

MINISTRY OF ENVIRONMENT AND URBANIZATION	
Thesis	Centralization and Management of Log Systems in the Ministry
Type	Ministry of Environment and Urbanization Expertise Thesis
Author	Samet CAN
Submission Date	10/05/2017
Key Words	Log, Event, Management
Advisor	Kudret Aslı BABACAN
Total Page	152

Abstract

Today, due to increased cyber attacks and compliance requirements, organizations are working on information security and event management. Log management is a concept they need when doing these studies. In order to centralize and manage logs, it is necessary to have basic knowledge about log types, log formats, log sources, log collection mechanisms, log retention strategies, log analysis, logging rules and compliance requirements. Collection of logs by specific standards is determined by the needs of the organization, compliance requirements and regulatory agencies.

In this thesis study, it is aimed to provide information about what kind of actions should be taken about logging by helping to indirectly establish awareness about security information and event management together with centralization and management of logs, to shed light on the work done in the organization and to determine the shortcomings in these studies.

TEŞEKKÜR

Tez çalışmam boyunca desteğini esirgemeyen başta anneme, Daire Başkanım İskender ERMİŞ'e, danışmanım Kudret Aslı BABACAN'a, ve mesai arkadaşlarıma teşekkür ederim.

Samet CAN

TABLO LİSTESİ

ŞEKİL LİSTESİ

KISALTMALAR

API : Application Programming Interface
BT : Bilişim Teknolojileri
CIFS : Common Internet File System
DNS : Domain Name System
EBCDIC : Extended Binary Coded Decimal Interchange Code
FISMA : Federal Information Security Management Act
FSMO : Flexible Single Master Operation
HDFS : Hadoop Distributed File System
HIDS : Host-based Intrusion Detection System
HIPS : Host-based Intrusion Prevention System
HIPAA : Health Insurance Portability and Accountability Act
IDS : Intrusion Detection System
IP : Internet Protocol
ISO : International Organization for Standardization
MIB : Management Information Base
NERC : North American Electric Reliability Corporation
NFS : Network File System
NIST : National Institute of Standards and Technology
NMS : Network Management Station
NSM : Network Security Monitoring
NTP : Network Time Protocol
PCI DSS : Payment Card Industry Data Security Standard
RDBMS : Relational Database Management System
SaaS : Software as a Service
SCP : Secure Copy
SIEM : Security Information and Event Management
SOAP : Simple Object Access Protocol
SSH : Secure Shell
SSL : Secure Socket Layer
TCP : Transmission Control Protocol
TTL : Time to Live
UDP : User Datagram Protocol
URL : Uniform Resource Locator

GİRİŞ

Loglar, sistem yönetimi (işletim sistemleri, disk sistemleri, yazıcılar gibi), kullanıcı ve uygulama yönetimi (giriş/çıkış, uygulama erişimi gibi) ve güvenlik (saldırı tespit sistemi, güvenlik duvarı gibi) için çok yararlı bir bilgi kaynağı olmakla birlikte çoğunlukla az değer görürler.

Çeşitli disk depolama ürünleri, donanım hataları oluştuğunda log mesajları üretmektedir. Bu bilgilere erişmek çoğu kez küçük sorunların büyük bir kâbusa dönüşmeden çözülmesi anlamına gelebilir.

Bir kullanıcı bir Windows ortamında oturum açtığında, bu işlem bir yerde oturum açma kaydı olarak loglanır. Buna kullanıcı yönetimi log verisi denilebilir. Bu kullanıcı ağın çeşitli bölümlerine her eriştiğinde bir güvenlik duvarı, ağ paketlerinin kullanıcının bilgisayarından ağın belirli bir bölümüne akışının izin verilip verilmemesiyle ilgili olarak ağ erişimini kaydeder. Buna güvenlik log verisi denilebilir. Bu kullanıcının oturum açma verileri, kullanıcının sunucuya erişmeye çalıştığını gösteren güvenlik duvarı kaydı ile eşleştirilebilir. Böyle bir eşleştirme, doğru bilgilere erişilebildiği sürece faydalı işlemlerin yapılmasını sağlayabilir; ancak bu doğru bilgileri elde etmek biraz zaman ve yeterli çalışma gerektirir.

Son yıllarda siber güvenlikle ilgili olaylar oldukça artmaktadır. Bununla ilgili olarak kurumlar, Bilgi Güvenliği ve Olay Yönetimi (SIEM) ürünleri alıp işletmeye başlamaktadırlar. SIEM ürünleri, log toplamakta ve topladığı loglardan korelasyonlar yaparak analizlerle birlikte olaya müdahalenin gerçekleştirilebilmesi için fayda sağlamaktadır. Bu nedenle olay yönetiminin iyi yapılabilmesi için kaynağın, yani logun iyi bilinmesi gerekir.

Bu uzmanlık tezi kapsamında logların merkezileştirilmesi ve yönetimi konusunda kurumdaki yazılım geliştiricilerin, güvenlik yöneticilerinin ve diğer paydaşların genel olarak ihtiyaç duyacağı kavramlar, kurallar, düzenlemeler ve çeşitli konular ele alınacaktır. Bununla birlikte kurumdaki mevcut durum değerlendirilerek nelerin yapılması gerektiği konusunda öneriler sunulacaktır. Bu tez çalışması on iki bölümden oluşmakta olup ilgili bölümlerin detayları aşağıda verilmektedir.

Birinci bölümde, log mesajının ne olduğu ve neden önemli olduğu açıklanarak loglamayla ilgili temel kavramlar verilecektir.

İkinci bölümde, syslog, SNMP ve Windows Olay Günlüğü'nden bahsedilerek log kaynakları sınıflandırılacaktır.

Üçüncü bölümde, log tutma, saklama biçimleri ve logları ilişkisel bir veritabanı yönetim sisteminde depolama konularında bilgiler verilecektir.

Dördüncü bölümde, log verilerini analiz etmek için hedeflerin belirlenmesi, planlama ve hazırlık konularına değinilerek basit log analizi konusunda bilgiler verilecek ve daha sonra basit log analizinin göz ardı edebileceği sorunları bulmaya yardımcı olan filtreleme, normalleştirme ve korelasyon adımlarından bahsedilecektir.

Beşinci bölümde, bir tür log analizi olan log veri madenciliği veya başka bir deyişle log bilgisi bulma çalışmalarından bahsedilecektir.

Altıncı bölümde, log analiziyle ilgili olarak raporlamadan bahsedilecek ve log verileri için en iyi raporların neler olduğu hakkında bilgiler verilecektir.

Yedinci bölümde, kurumların loglarla ilgili olarak yaptığı yaygın hatalar ele alınarak genel kurallardan bahsedilecektir.

Sekizinci bölümde, geliştiricilere yönelik olarak log mesajlarının iyi üretilebilmesi konusunda kavramlardan ve yöntemlerden bahsedilecektir.

Dokuzuncu bölümde, düzenlemelere yönelik olarak loglama gereksinimleri ile yönetmelik ve politikalar hakkında bilgiler verilecektir.

Onuncu bölümde, tez çalışması kapsamında bahsedilen konularla ilgili olarak kurumun mevcut durumu hakında bilgiler verilecek ve diğer kurumlardaki durumlar ele alınarak ortaya çıkarılan sonuçlarla birlikte kuruma öneriler sunulacaktır.

1. LOG TANIMI VE GENEL KAVRAMLAR

Loglama, log analizi ve log yönetimi için kullanılan terimlerin çoğu anlaşılamayan, yanıltıcı veya birden fazla anlam içerebilmektedir. Bu çalışma boyunca kullanılacak terimlerin tanımları aşağıda yer almaktadır.

- Bir olay, bir ortamda meydana gelen bir hadise olup genellikle durum değişikliği içerir. Bir olay genel olarak, olayın sebeplerini veya etkilerini açıklamaya veya anlamaya yardımcı olabilecek zaman kavramı, oluşum ve olay veya ortam ile ilgili herhangi bir ayrıntıyı içerir.
- Bir olay alanı, olayın bir özelliğini tanımlar. Bir olay alanına örnek olarak tarih, saat, kaynak IP, kullanıcı kimliği ve sunucu kimliği verilebilir.
- Bir olay kaydı, birlikte tek bir olayı tanımlayan olay alanları topluluğudur. Olay kayıtlarına "denetim kaydı" ve "log kaydı" da denilir.
- Log, olay kayıtlarının bir toplamıdır. Genellikle "veri logu", "etkinlik logu", "denetim logu", "denetim izi", "log kaydı" ve "olay logu" gibi terimler logla aynı anlamda kullanılır.
- Denetim, logları bir ortamda (örneğin; bir elektronik sistem içinde) değerlendirme sürecidir. Denetimin amacı, genel durumu değerlendirmek ya da önemli veya problemli herhangi bir etkinliği belirlemektir.
- Kaydetme, tek bir olayla ilişkili olay alanlarını içeren bir olay kaydı oluşturma eylemidir.
- Loglama, olay kayıtlarını logların içine toplama eylemidir. Örnek olarak log girdilerini bir metin log dosyasına, ikili (binary) dosyalara veya veritabanlarına saklamak loglamadır. (Mitre, 2010)

Bir güvenlik vakası, ortamda kötü bir şeyler olduğuna işaret eden bir veya daha fazla güvenlik olayının bir oluşumudur. Bu kötü şeyler, bir sisteme yetkisiz erişim, bilgi hırsızlığı, hizmeti engelleme veya kuruma özgü bir dizi etkinlik olabilir.

Log kayıtları, denetimin tamamında veya bir parçasında kullanılabilir. İyi loglama denetime yardımcı olur; ancak loglama sadece denetim için değildir. Siber saldırı veya bir felaket senaryosunda yapılabilecek tek iş olan loglara bakmak dışında, bu tür felaketlerle karşılaşmamak adına log içeriğini kullanabilmeyi öğrenmek gereklidir. Log analiz araçları, bir şeyler ters gittiğinde değil de aylık, haftalık, günlük

hatta anlık olarak kullanılmalı ve alışkanlık haline getirilmelidir. Günümüzde yer alan düzenlemeler, kurumlarda bu tür iyi alışkanlıkları benimsemeye zorlamaktadır.

Uyarı veya alarm, genellikle dikkat edilmesi gereken bir olaya karşılık olarak yapılan bir eylemdir. Log dosyalarının uyarılar içerdiği düşünülür. Özellikle, bazı saldırı tespit logları, aslında IDS uyarılarının toplamıdır. Bu çalışmada uyarı, belirli bir log mesajının hızlı bir şekilde bir kullanıcıya bildirilmesi gerektiğinde yapılan bir eylem olarak ele alınacaktır.

1.1 Loglama Türleri

Çoğu işletim sistemi, birkaç farklı türde loglama yeteneğine sahiptir ve farklı log mesajları üretir. Bununla birlikte, sistem yöneticileri de dâhil birçok bilgisayar kullanıcısı, var olan logların ne olduğunu bilmemektedir. Bütün log kaynakları hesaba katıldığında loglamanın dört temel sebebi vardır:

- **Güvenlik Loglama:** Saldırıların, zararlı yazılımların, veri hırsızlığının ve diğer güvenlik konularının tespit edilerek gerekli karşılığın verilmesine odaklanmıştır. Güvenlik odaklı loglamanın klasik örneği, uygun bir yetkilendirmeye sahip olmayan kimsenin kaynağa erişimi olup olmadığını analiz etmek amacıyla kimlik doğrulamasını ve diğer erişim isteklerini kaydetmektir.
- **Operasyonel Loglama:** Sistem operatörlerine hata ve olası işlem durumları gibi yararlı bilgileri sağlamak amacıyla yapılır. Operasyonel loglama, erişim tabanlı fiyatlandırma ve web sunucusu erişim logları gibi yalnızca bilgi teknolojileri için değil ticari amaçlı hizmet sağlamak ve finansal kararlar almak için de kullanılabilir. Bu kategori oldukça geniştir ve çok sayıda log türünü kapsar.
- **Uyumluluk Loglama:** Güvenlik loglama ile çakışır; çünkü düzenlemeler genel olarak sistemlerin ve verilerin güvenliğini artırmak amacıyla yazılmaktadır. Uyumluluk loglamanın iki türü vardır. Bunlardan ilki bilgi teknolojilerini etkileyen düzenlemeler (PCI DSS, HIPAA, ISO vb.) ve direktiflerdir. İkincisi ise ortak ölçütler (common criteria) gibi sistem düzenlemeleri ve sistem tasarımı ve güvenliği ile ilgili diğer direktiflerdir.

- **Hata Ayıklama Loglama:** Sistem operatörleri için değil de uygulama ve sistem geliştiricileri için yararlı olan özel bir loglama türüdür. Bu loglama, üretim (production) sistemlerinde genellikle devre dışıdır; ancak istek üzerine etkinleştirilebilmektedir. Hata ayıklama loglarının içerisindeki mesajların birçoğu, uygulamanın yapısı hakkında tam bilgi sahibi olan ve bazen de uygulama kaynak koduna sahip geliştiriciler tarafından analiz edilebilir. (Chuvakin ve Schmidt, 2013)

Bu dört loglama türü neredeyse tüm log kaynakları (olay üreticileri) tarafından üretilir; ancak farklı sistemler (olay tüketicileri) tarafından analiz edilir ve tüketilirler.

İşletim sistemi, yukarıda bahsedilen spektrumdan loglar üretir. Windows işletim sistemlerinde Olay Günlüğü adı verilen bir loglama sistemi bulunur. Sistem üzerinde belirli işlemler gerçekleştirildiğinde, işletim sistemi veya uygulama Olay Günlüğü'ne bazı bilgiler yazar. 1993 yılında yayınlanan Windows NT sürümünden bu yana bütün Windows işletim sistemlerinde bu bilgileri görüntülemeye yardımcı olan Olay Görüntüleyici bileşeni yer almaktadır. (Event Viewer, 2016)

Unix sistemlerde, sendmail programlarının yapıldığı 1980'lerden beri syslog adı verilen bir loglama teknolojisi vardır. (syslog, 2016) Windows'taki gibi, işletim sistemi ve bazı uygulamalar syslog'a mesaj yazar. Örnek bir girdi aşağıda verilmektedir:

Feb 16 14:58:10 abc named[25]: sysquery: findns error (NXDOMAIN) on xyz.csb.gov.tr?

Bu mesaj, kendisinin yanı sıra tarih, sunucu adı ve mesajın üretildiği hizmetin ismini de içerir. Bu özel mesaj, mevcut olmayan bir alan üzerinde DNS araması yapma girişimini gösterir.

Bazı uygulamalar kendi loglarını oluştururlar. Web sitesi yöneticileri, müşteri profillerini araştırarak onların davranışlarını öğrenmek ve pazarlama ile ilgili görevleri yerine getirmek için web sunucusu loglarını incelerler. Web sunucusu logları, web sunucusu performansını ve web sitesinin erişilebilirliğini incelemek için de kullanılır.

Yukarıda bahsedilen loglama sistemlerinin tamamı, güvenlik logu analizi için faydalıdır. Windows, Unix ve Web sunucusu logları, kurumun güvenlik durumu ile ilgili önemli ipuçları içerebilir.

1.2 Loglama Bileşenleri

Yukarıda bahsedilen amaçlar için kullanılan loglama mekanizmalarının mantıksal olarak dört bileşeni vardır:

- Log taşıma
- Log sözdizimi ve biçimi
- Log içeriği (taksonomi)
- Loglama ayarları

1.2.1 Log Taşıma

Log taşıma, log mesajlarını bir yerden başka bir yere taşımanın basit bir yoludur. Syslog ve ürüne özgü taşıma protokolleri gibi birçok olay taşıma protokolü bulunmakla birlikte kendi taşıma yöntemi olmayan loglama mekanizmaları (yerel log dosyası gibi) da vardır. Uygun bir log taşıma mekanizması; log verisinin bütünlüğünü, erişilebilirliğini ve gerekirse gizliliğini muhafaza etmeli, log biçimini ve anlamını koruyarak oluşmuş olan tüm olayların doğru zamanlama ve olay sırası ile tutarlı bir şekilde gösterilmesine imkân sağlamalıdır. Log taşımadaki gereksinim, logların, olay kayıtlarının ve olay akışının bütünlüğünün korunmasıdır. Daha da önemlisi, her log girdisi doğru zaman damgasıyla muhafaza edilmelidir.

Bilinen bazı log taşıma mekanizmaları şunlardır:

- Syslog
- SNMP
- SOAP
- SCP, CIFS, NFS gibi dosya paylaşımı

Syslog, log mesajlarını 514 numaralı porttan UDP aracılığıyla taşımak için yapılandırılmış, bugüne kadar milyonlarca Unix türevi sistem ve ağ cihazları tarafından kullanılan en popüler log taşıma mekanizmasıdır. Birden fazla zafiyetine (garantili mesaj ve erişilebilirlik eksikliği gibi) rağmen syslog, log verilerini taşımanın en temel yoludur. (syslog, 2016)

1.2.2 Log Sözdizimi ve Biçimi

Herhangi bir biçimdeki her log dosyası bir sözdizimi içerir. Log sözdizimi kavramsal olarak Türkçe gibi bir dilin sözdizimine benzer. İnsan dilindeki bir

cümlenin sözdizimi, genellikle bir konu, bir yüklem, bazen de tamamlayıcılar ve nitelikler içermektedir. Cümlenin sözdizimi, cümle üyeleri ve anlamları arasındaki ilişkileri kapsar. Sözdizimi, mesaj içeriğini ele almaz. Başka bir deyişle sözdizimi, kullanılan belirli kelimelerle ilgili değil de neyin söyleneceği seçilerek nasıl yapılandırıldığı ile ilgilidir.

Log sözdizimi, log mesajlarının nasıl oluşturulduğu, taşındığı, saklandığı, incelendiği ve analiz edildiğini tanımlar. En önemsiz durumda bile her olay kaydı bir metin dizesi olarak ele alınabilir ve olay tüketicisi bu log üzerinde tam metin araması yaparak tutarlı olmasını bekleyebilir; ancak güvenilir bir otomatikleştirilmiş olay analizi için, olay tüketicileri ve üreticilerinin olay kaydının sözdizimini anlamaları ve hemfikir olmaları gereklidir.

Log mesajlarının her biri çeşitli türden bilgi kalıplarından oluşur. Yaygın olarak kullanılan alanlar şu şekildedir:

- Tarih/zaman
- Log girdisi türü
- Üreten sistem
- Üretildiği uygulama veya bileşen
- Başarı veya başarısızlık göstergesi
- Log mesajının önem derecesi veya önceliği
- Kullanıcı etkinlikleriyle ilgili loglar için kullanıcı adı

Log sözdizimi, log verilerinin her türlü otomatikleştirilmiş analizi için önemlidir. Log verilerinden mantıklı sonuçların çıkarılabilmesi için logu, sözdizimine göre kendi parçalarına ayırmak gerekir. Çoğu durumda otomatikleştirilmiş analiz, genellikle belirli bir şablonda kodlanmış bir log sözdizimine ihtiyaç duymaktadır.

Bilinen bazı log biçimleri şunlardır:

- W3C Extended Log File Format (W3C, Extended Log File Format)
- ArcSight Common Event Format (Hewlett Packard, 2016)
- IDMEF (The Intrusion Detection Message Exchange Format, 2007)
- Cisco SDEE/CIDEE (Cisco, 2009)
- Syslog (The Syslog Protocol, 2009)

Biçimlendirilmiş olay kaydının her alanı bazı gösterimlerde farklı bilgiler içerebilmektedir. Anlaşılabilir gelse de "Thu 10 6 2016 15:05pm" gösterimi ile "2016-10-06T15:05:00Z" gösteriminin aynı zamanı temsil ettiği belirgin değildir. İlk gösterimde, zaman dilimi bilgisi eksikliği nedeniyle zaman gerçekten bilinmemektedir ve bölgesel tarih seçimi nedeniyle 10 Haziran mı yoksa 6 Ekim mi belirsizdir.

Günümüzde log mesajlarının çoğu, zaman damgası gibi mesajın küçük bir bölümü hariç, ne yazık ki belirli bir biçimi takip etmemektedir. Bu sebeple loglar serbest biçimli metin olarak düşünülebilir.

Log biçimleri farklı yönlerden ele alınabilir; log dosyası ikili mi yoksa ASCII biçiminde mi, basit bir metin tarayıcı veya düzenleyicisiyle okunabiliyor mu gibi. Okunabilir ASCII logunun en yaygın örneği syslog'tur. Web sunucuları, güvenlik duvarları ve bütün platformlardaki birçok uygulama kolayca görüntülenebilen metin dosyalarına loglama yaparlar.

İkili dosyanın en yaygın örneği Windows Olay Günlüğüdür. Olay Günlüğü kolayca okunamaz. Olay Günlüğü'nün okunabilmesi için Olay Görüntüleyici'sinin kullanılması gereklidir. Bu yardımcı program, ikili dosyayı insan tarafından okunabilen bir olaya dönüştürür. Yaygın olarak kullanılan bir diğer ikili biçim ise oturum kayıtlarını içeren Unix utmp dosya türüdür. (utmp, 2016)

Bütün loglar metin tabanlı ve okunması kolay olsaydı, insanlar için daha basit olurdu. Fakat programcılar log verisi için ikili biçimi seçmektedir. Çünkü ikili loglama için performans ve alan gibi zorlayıcı nedenler vardır. Saniyede binlerce kaydın loglanması gerekiyorsa, bu görev güncel işlemcileri bile zorlayabilir. Loglama aleyhinde en yaygın iddialardan biri performansı etkilediğidir. Bunun doğruluk payı olsa da sistem çöktüğü zaman nedeni konusunda bir bilgi bulunamayacağı göz önüne alınırsa burada performanstan bahsedilmesinin hiçbir önemi yoktur.

İkili biçiminde log girdileri genellikle daha küçüktür. Böylelikle biçimlendirmek ve yazmak için daha az işlem gerekir. Mesajlar genellikle ASCII mesajından küçük olduğundan, log dosyaları diskte daha az yer kaplar ve taşınırken daha az giriş/çıkış işlemi oluşur.

İkili loglar genellikle ayrıştırmak için daha az işlem gerektirir ve analizleri daha verimli hale getirmek için açıkça tanımlanmış alanlar ve veri türlerine sahiptir. Bir ASCII log ayrıştırıcı daha fazla veri işlemek zorundadır ve yararlı bilgi parçalarını ayıklamak için çoğu zaman desen eşleme yöntemini kullanması gerekir.

Sıkıştırılabilir olması ikili loglama için diğer bir nedendir. Sıkıştırılmış bir log, ikili bir logu ifade etmektedir. Bir log dosyası şifrelenmiş bir biçimde ise, ikili de olmalıdır. İkili loglamanın bir diğer nedeni, okunmasının daha zor olmasıdır. Anlaşılmamanın log dosyalarını daha güvenli hale getirdiği düşünülmektedir.

Log biçiminin başka bir türü, metin dosyası veya ikili olmayan veya ikili biçimin süslü bir çeşidi olarak düşünülmüş ilişkisel veritabanıdır. Bir ilişkisel veritabanı, ikili kayıtları ekleme (insert) ve alma (select) işlemleri için veritabanı şemasında tanımlanan bir tabloya saklar. Bir veritabanı şeması, tüm veritabanındaki tabloları ve bir tablodaki kayıtları tanımlayan basit bir yöntemdir.

Log türleri arasında bir başka önemli ayrım da biçimin açık mı yoksa tescilli mi olduğudur. Açık biçim, biçimin bir yerde belki de standart bir belgede (ISO, ANSI gibi) veya bir başvuru belgesinde (RFC gibi) dokümante edildiği anlamına gelir. Tescilli biçim ise, belirli bir cihaz üreticisi tarafından kullanılır. Tescilli biçimler genellikle dokümante edilmemiştir ve log okuma ve işleme araçlarının kullanılabilmesi için üreticinin inisiyatifine kalınmaktadır. Bir kimse tescilli bir metin biçimini çok fazla gayret sarf ederek anlayabilir. Bununla birlikte, belirli bir alanın anlamını, veri türünü veya değer aralığını yanlış yorumlama riski de vardır ve oldukça yararsız veya yanıltıcı sonuçlar elde edilebilir.

Log analizi ile ilgili uygulamalardan bazıları, çeşitli log türlerinin sözdizimini tanımlamak için mesaj şemaları geliştirmektedir. Bu şemalar, cihazlar, sistemler, uygulama ve diğer kaynaklardan gelen çeşitli log dosyalarında oluşabilecek herhangi bir mesaja uyacak şekilde tasarlanmaktadır.

Log analiz ürünü tarafından mesajlar toplandığında bazı veriler de bu mesajlara eklenebilmektedir. *Event, EventType* gibi çeşitli alanlar saldırı tespit sistemleri tarafından oluşturulan olayı tamamlamak için bir log analiz sistemi tarafından ilgili mesajlara eklenmektedir.

Genel olarak çeşitli sistem ve cihaz üreticileri, ortak sözdizimi seçeneğine bağlı birkaç yaygın log türünü kullanmaktadır. Örneğin; bazı güvenlik loglamaları, daha kolay ve daha zengin bilgi entegrasyonu için XML ile yapılır. Birçok operasyonel loglamalar, yapılandırılmamış metin ya da yarı yapılandırılmış mesajlar kullanılarak syslog üzerinden yapılır. Benzer şekilde, hata ayıklama logları da syslog veya metin dosyalarıdır. Yüksek performanslı loglamalar genellikle ikili ve tescilli biçimdedir.

Tablo 1.1'de yaygın olarak kullanılan loglama mekanizmalarının karşılaştırması gösterilmektedir.

Tablo 1.1 Loglama Mekanizmalarının Karşılaştırılması

	XML Loglama	**Syslog Loglama**	**Metin Dosyası Loglama**	**Tescilli loglama**
Okunma yöntemi	Çoğunlukla makine okuyabilir	Çoğunlukla manuel okunabilir	Sadece manuel okunabilir	Sadece makine okuyabilir
Genel kullanım durumu	Güvenlik loglama	Operasyonel loglama, hata ayıklama loglaması	Hata ayıklama loglaması	Yüksek performanslı loglama
Kullanıldığı örnekler	IPS güvenlik cihazı	Yönlendirici ve anahtarlar	Uygulama hata ayıklaması	Güvenlik duvarı loglama, paket yakalama
Tavsiye	Yapılandırılmış bilgilerin aktarılması gerektiğinde kullanın	Otomatik analizi basitleştirmek için ad=değer gibi bir yapı ekleyin	Operasyonlar sırasında loglar etkin bırakılacaksa, otomatik analizi etkinleştirmek için belirli bir yapı ekleyin	Yalnızca yüksek performanslı kullanımlar için kullanın
Dezavantajları	Göreceli olarak düşük performans, geniş log mesajı boyutları	Log mesajı yapısının olmaması, otomatik analizin karmaşık ve maliyetli olmasına neden olur	Genellikle loglar yalnızca uygulama geliştiricileri tarafından anlaşılabilir	İkili metin haline dönüştürebilen özel bir uygulama olmadan okunabilir değil

(Chuvakin vd. 2013)

1.2.3 Log İçeriği (Taksonomi)

Logların içeriği, logların gerçekte ne anlama geldiğini gösteren bir taksonomidir. Log olay taksonomisi, loglanan olayları sınıflandırmanın bir yoludur. Aynı olayın birden fazla sistem tarafından loglanması durumunda, olayın taksonomi tanımlarının aynı olması beklenmelidir. Bir bilgisayar, iki logun aynı türdeki olaya ait olup olmadığını hemen belirleyebilmelidir. Bunun yapılabilmesini sağlamak için, öngörülebilir biçimde bir araya getirilen iyi tanımlanmış kelimelerden oluşan bir

koleksiyon olmalıdır. Bu ise log taksonomisidir. Belirlenen kelimeler, etkinlik türünü, katılan aktörleri, sonucu ve diğer olay verilerini tanımlamalıdır.

Olay taksonomilerine ilişkin henüz iyi tanımlanmış genel bir standart bulunmamaktadır. Bununla birlikte, günümüzde piyasadaki çoğu bilgi güvenliği ve olay yönetimi ve bazı log yönetimi araçlarının üreticileri, ürünlerinin içinde kullandıkları log taksonomisi geliştirmektedirler. Ne yazık ki, her üretici kendi taksonomisi için farklı temel ilkeleri kullanmaktadır.

Logların içerikleri genel olarak şunları sağlayabilir:

- Kimin oturum açtığı, neler yaptığı, aldığı ve gönderdiği e-postalar gibi kullanıcı etkinliği hakkında bilgiler içerebilir.
- Disk hataları gibi bozuk veya bozulacak şeyleri haber verebilir.
- Kaynak kullanımı ve performans hakkında bilgi verebilir.
- Durum değişiklikleri, başlatma ve durdurma gibi konular hakkında bilgi içerebilir.
- Saldırı girişimleri hakkında bilgi verebilir ve saldırı olduğu zaman başarılı bir şekilde bunu gösterebilir.

Aşağıda yer alan türler güvenlik, operasyon ve hata ayıklama mesajlarının bütün spektrumunu kapsar.

1. **Değişim Yönetimi**: Sistem değişiklikleri, bileşen değişiklikleri, güncellemeler, hesap değişiklikleri ve bir değişim yönetimi sürecine tabi tutulabilecek her şeyin kayıtlarıdır. Bu loglar genellikle ekleme, silme ve güncelleme gibi bölümlere ayrılır. Bunlar, güvenlik ve operasyonel türler arasında kesişebilir.
2. **Kimlik Doğrulama ve Yetkilendirme:** Kimlik doğrulama ve yetkilendirme kararlarının kayıtlarıdır ve her uygulama ve ağ cihazı tarafından üretilmesi gerekir. Bunlar, operasyonel kullanımları da olan güvenlik mesajlarıdır.
3. **Veri ve Sistem Erişimi:** Uygulama bileşenlerine ve verilere (dosya veya veritabanı tablosu gibi) erişim kayıtlarıdır ve ortak güvenlik konularında operasyonel kullanımları içerir. Bazı durumlarda bu mesajların her zaman etkin olmaması ve hassas ortamlarda üretilmemesi gerekebilir.

4. **Tehdit Yönetimi:** Geleneksel saldırı uyarılarından güvenlik politikalarını ihlal eden diğer faaliyetlere kadar bu tür mesajlar, özel bir güvenlik işlevine sahip ağ cihazları (güvenlik duvarı gibi) tarafından üretilir.
5. **Performans ve Kapasite Yönetimi:** Çeşitli eşikler, bellek ve hesaplama kapasitesi kullanımı ve diğer sınırlı kaynak kullanımı da dâhil olmak üzere sistem performansı ve kapasite yönetimi ile ilgili geniş bir mesaj kategorisidir. Bazen güvenlik kullanımını da olan çok yaygın operasyonel mesajlardır.
6. **İş Sürekliliği ve Erişilebilirlik Yönetimi:** Çoğu sistem, kapatıldığında veya başlatıldığında bir log oluşturur: yedekleme ile ilgili erişilebilirlik mesajları veya işin devam etmesiyle ile ilgili mesajlar. Bunlar, nadir güvenlik kullanımı olan çok yaygın operasyonel mesajlardır.
7. **Çeşitli hatalar ve arızalar:** Kullanıcıların dikkatini çekmeyi amaçlayan diğer sistem hataları burada sınıflandırılmıştır. Bunlar, cihaz yöneticisi tarafından bir işlem yapılmasını gerektiren veya gerektirmeyen, kritik olmayan operasyonel mesajlardır.
8. **Çeşitli hata ayıklama mesajları:** Hata ayıklama logları genellikle geliştiricilerin takdirine göre oluşturulur ve sınıflandırılması oldukça zordur. Çoğu hata ayıklama logları, operasyonel üretim ortamlarında aktif değildir. (Chuvakin vd. 2013)

1.2.4 Loglama Ayarları

Loglama ayarları, olay üreticileri ve sistem operatörleri için hangi olayların loglanacağı ve her bir cihazda hangi loglamanın etkinleştirileceğine karar vermeleri amacıyla kullanılan ortak bir yöntemdir. Olayların ortak bir ifadede belirlenmesi, ne tür olayların üretilmesi gerektiği konusunda fayda sağlamaktadır. Bir güvenlik duvarının engellenen bağlantı girişimleri gibi olayları loglaması beklenirken, günümüzde standart loglama kuralları bulunmamaktadır. Ayrıca, hangi olayların loglanacağı değil, aynı zamanda her bir olay için hangi detayların loglanacağı da önemlidir.

PCI DSS uyumluğuna göre loglamada, her bir log girdisi için kullanıcı adı, olay türü, tarih ve saat, başarı veya başarısızlık göstergesi, olayın kaynağı, etkilenen sistem bileşeninin adı belirtilmelidir. Şekil 1.2'de PCI DSS Standardı'nın 3.2 sürümü gösterilmektedir.

Şekil 1.1 PCI DSS Standardı v3.2

PCI DSS Requirements	Testing Procedures
10.3 Record at least the following audit trail entries for all system components for each event:	**10.3** Through interviews and observation of audit logs, for each auditable event (from 10.2), perform the following:
10.3.1 User identification	**10.3.1** Verify user identification is included in log entries.
10.3.2 Type of event	**10.3.2** Verify type of event is included in log entries.
10.3.3 Date and time	**10.3.3** Verify date and time stamp is included in log entries.
10.3.4 Success or failure indication	**10.3.4** Verify success or failure indication is included in log entries.
10.3.5 Origination of event	**10.3.5** Verify origination of event is included in log entries.
10.3.6 Identity or name of affected data, system component, or resource.	**10.3.6** Verify identity or name of affected data, system component, or resources is included in log entries.

(PCI DSS, 2016)

1.3 Loglama Kriterleri

Loglanan bilgilerin eksik ya da yararsız olduğu birçok durum vardır. İzinsiz giriş tespiti, kaynak yönetimi veya denetim amacıyla bir log mesajının kullanılabilmesi için hangi bilgilerin gerekli olduğu önemli bir konudur. Çok sayıda log tipi ve bundan daha fazla da log üreten cihaz vardır, bu nedenle tek bir ölçüt tanımlamak zordur.

Genel olarak loglar şunları söylemelidir:

- Ne oldu
- Nerede oldu
- Ne zaman oldu
- Kim katıldı
- Kişi nereden geldi

Yukarıdaki liste, mutlaka olması gerekenlerdir. Bunlar, kullanıldıkları diğer disiplinlerden alınırlar: gazetecilik, cezai soruşturma gibi. Ayrıca olayın bütününe gidilebilmesi için şunların da bilinmesi istenebilir:

- Etkilenen nedir
- Nereden daha fazla bilgi edinilir
- Verilen bilgilerin gerçekliğinden nasıl emin olunur
- Bundan sonra ne olacak
- Bu konuda ne yapılması gerekir
- Başka nelere dikkat edilmesi gerekir

Günümüzde kurumlar ağ cihazını ve bu kapsamda da sunucu loglamayı kontrol altında aldığı için, üzerinde çalışılması gereken önemli bir konu uygulama loglama olarak görünmektedir. Uygulama loglarıyla ilgili birçok sorun bulunmaktadır. Loglar genellikle eksik ve kritik ayrıntılar atlanmaktadır. Herhangi bir yerde bu konuyla ilgili standart form veya içerik bulunmamaktadır. Sekizinci bölümde bu konuyla ilgili ayrıntılı bilgi verilecektir.

Bilgi iki kategoriye ayrılır: şimdi ilgilenilmesi gereken (kritik) ve daha sonra ilgilenilmesi gereken (kritik olmayan). Kritik olan bilgi, açık ve işlenebilir, yüksek öncelikli logu ifade eder. Bu loga bir girdinin eklendiği her yerde, operatör veya analist derhal bir aksiyon almalıdır. Kritik olmayan bilgi ise hemen harekete geçilmesine gerek duyulmayan logları ifade eder; ancak bu loglar zamanı geldiğinde özetlenerek, denetim veya adli analiz için önemli bilgiler sağlayabilir. Tablo 1.2'de kritiklik durumuna göre ayrılan, loglanması gereken olayların bir listesi verilmektedir. Bu liste kapsamlı değildir; ancak ideal olarak loglanması gerekenler konusunda fikir sahibi olunmasını sağlar.

Tablo 1.2 Kritiklik Durumuna Göre Loglar

Kritik Log (derhal aksiyon alınmalı)	**Kritik Olmayan Log (bir kenarda tutulmalı)**
Sistem operasyonlarını etkileyen hatalar	Sistem durumu mesajları
Başarılı olan saldırılar	Saldırı girişimleri
Başarma şansı yüksek olan saldırılar	Etkisi az olan saldırılar
Maksimum sistem kapasitesine erişilmesi	Belirli değere yakın sistem kapasitesine ulaşılması
Güvenlik ve erişilebilirlik sorunlarına yol açan sistem değişiklikleri	Çeşitli sistem değişiklikleri
Sistem çökmesi	Sistem başlatma/kapatma
Başarısız giriş	Başarılı giriş
Donanım hatası	Donanım durumu mesajı
Güvenlikle ilgili yapılandırma değişikliği	Düzenli ve otomatik yapılandırma değişikliği
Yetkisiz bağlantı tespiti	Bağlantı kuruldu/sonlandırıldı

(Chuvakin vd. 2013)

2. LOG VERİ KAYNAKLARI

Log analiz sürecinin ilk adımı, log kaynağının ne olduğunu anlamaktır. Bazı uygulamalar ve sistemler loglamayı varsayılan olarak yapmaktadır. Uygulama log verileri gönderiyor olabilir; ancak karşısında onu alması gereken hiçbir şey bulunmayabilir. Bu bölümde, sistem ve uygulamalar için temel log kaydının nasıl etkinleştirileceği gösterilecektir. Bazı uygulamalar ve sistemler için varsayılan zorunlu ayarlar yeterli olmadığından, uygulama ayarlarını değiştirerek logların kalitesinin nasıl artırılacağından bahsedilecektir. Genel amaç, ortamda loglamanın nasıl kurulacağı hakkında fikir vermek ve hâlihazırda bulunan araçları güçlendiren örnekler sunmaktır.

2.1 Loglama Kaynakları

Log kaynakları genel olarak iki kategoriye ayrılır:

- İtme tabanlı
- Çekme tabanlı

İtme tabanlı log kaynakları ile cihaz veya uygulama yerel diske ya da ağa bir mesaj gönderir. Ağ üzerinden olursa, bu mesajı almaya hazır bir log toplayıcısına sahip olunması gerekir. Üç temel itme tabanlı kaynak; syslog, SNMP ve Windows Olay Günlüğü'dür. Bunlar, log mesajının iletildiği protokollerdir. Teknik olarak Windows Olay Günlüğü; protokol, aktarım mekanizması, saklama ve erişimi kapsar.

Çekme tabanlı log kaynakları ile uygulama log mesajını kaynaktan çeker. Bu yöntem, istemci-sunucu modeline dayanır. Bu şekilde çalışan çoğu sistem, log verilerini bazı tescilli biçimlerde saklar. Örneğin; Checkpoint ürünü, geliştiricilerin güvenlik duvarı loglarını alabilmesi için uygulamalar yazarken kullanabilecekleri OPSEC kütüphanesini sunar. Diğer ürünler MSSQL, Oracle, MySQL gibi veritabanlarını kullanarak verilerini saklamaktadır. Logları bir veritabanından çekmek kolaydır ve bu işlem bir komut dosyası veya program ile yapılabilir.

Burada, yaygın olarak kullanılan üç log kaynağı protokolü ele alınacaktır.

2.1.1 Syslog

Syslog, Unix/Linux da dâhil olmak üzere pek çok sistemde yaygın olarak kullanılan bir loglama biçimidir. Yoğun bir şekilde kullanılmasına rağmen resmi olarak standartlaştırılmamıştır. Standartlaştırma eksikliği nedeniyle, uygulamaları

sistemlerde çok çeşitli olabilir ve bazı ortamlarda olayların platformlar arası ilişkisini zorlaştırabilir. Syslog, ilk olarak hata ayıklama bilgilerini toplamak için oluşturulmuştur. Bu nedenle log analizi için uygun olmayan bazı sınırlamaları vardır.

Syslog, syslog arka plan programından (syslogd) oluşur. Sistemin açılışı ve kapanışında sırasıyla başlatılır ve durdurulur. Uygulamalar, syslog kütüphane çağrıları aracılığıyla syslogd ile iletişim kurar. Syslogd, bir Unix alan soketi üzerinden uygulamalardan ve çekirdekten log kayıtlarını alır. Syslogd, uzak makinelerden 514 numaralı port üzerinden UDP mesajları aracılığıyla isteğe bağlı olarak veri alabilir. Rsyslog ve syslog-ng gibi modern syslog türleri TCP ile de çalışır.

Syslogd'nin davranışı genellikle */etc/syslog.conf* gibi bir yapılandırma dosyası tarafından kontrol edilir. Bu arka plan programı (*daemon*), log mesajlarını Unix alan soketi */dev/log*'dan okur ve bir veya daha fazla çıktı dosyasına yazar ya da logları UDP aracılığıyla bir toplama sunucusuna iletir. Yapılandırma dosyasında bir değişiklik yapıldığında, yeni yapılandırmayı tekrar okuması için syslogd işlemine bir SIGHUP gönderilmesi gerekir.

Syslogd normal olarak önyükleme sırasında başlatılır, manuel olarak başlatmak için sadece *syslogd* komutu çalıştırılmalıdır. Syslogd için standart olarak gerekli olanlar; yapılandırma dosyasının konumu, işaret aralığı, ağdan gelen veriyi kabul edip etmeme ve Unix alan soketinin yolu olarak kabul edilir. Yapılandırma için belirlenen argümanlar, Unix/Linux'un farklı türlerine göre değişkenlik gösterir. Örneğin; Linux, diğer bilgisayarlardan logları kabul etmek için "*-r*" kullanırken OSX aynı işlem için "*-u*" kullanır. Bazı syslogd sürümleri, herhangi bir komut satırı argümanına gerek duymadan uzak mesajları otomatik olarak kabul eder. Bu seçenekler genellikle varsayılan değerlere sahiptir, varsayılanları geçersiz kılmak için komut satırı değişkenleri kullanılır. İşletim sistemine özgü bilgiler için yerel kılavuzlara bakılması gerekir.

Varsayılan syslogd yapılandırma dosyasının (*/etc/syslog.conf*) içeriği de dağıtımlar arasında değişir. Çoğu yapılandırma, varsayılan olarak herşeyi loglamazken bazıları da farklı şeyleri farklı dosyalara loglar. Loglamak istenilen şeylerin loglanıp loglanmadığını belirleyebilmek için syslog yapılandırma dosyası her zaman incelenmelidir.

2.1.1.1 Syslogd ile Temel Loglama

Syslogd ile loglamayı başlatabilmek için basit bir yaklaşım; her şeyi yerel bilgisayardaki tek bir dosyaya loglamaktır. Her şeyi tek bir dosyaya loglamak iki avantaja sahiptir. İlki, bu dosyaya göz atarak nelerin loglandığını görebilmektir. Neyin loglandığı bilinmiyorsa, nereye loglanacağına dair de bir kararın verilmesi zordur. İkincisi, bu mesaj merkezi bir log sunucusuna yönlendirildiğinde yerel kopya, sorun giderme ve yedekleme amacıyla kalmaya devam edecektir.

Her şeyi bir dosyaya loglayabilmek için, aşağıdaki komut *syslog.conf* dosyasına koyulmalıdır:

**.debug /var/log/messages*

İstenilirse farklı bir yol adı kullanılabilir. Fakat farklı bir yol adı kullanıldığında syslog yazmaya başlamadan önce ilgili dizinde boş bir dosya oluşturmak gerekebilir.

Aynı bilgiyi birden fazla dosyaya gereksiz yere yazmamak için dosyadaki diğer satırlar yorum satırına dönüştürülebilir. Yorum yapmak veya devre dışı bırakmak istenilen satırların önüne bir '#' ifadesi yerleştirilmelidir.

Syslog arka plan programının (syslogd) işlem kimliğini almak için ps komutu kullanılır:

ps -ef | grep syslog
root 675 1 0 Feb 16 ? 0:00 syslogd –m 0

syslogd'yi yeniden başlatmak için aşağıdaki SIGHUP gönderilmelidir:

kill -HUP 675

/var/log/messages dosyasına bakılırsa aşağıdaki gibi bir mesaj görülecektir:

Feb 16 17:50:05 hostname syslogd 1.4.1: restart.

Unix'in bazı sürümlerinde syslog arka plan programı, varsayılan olarak uzak sistemlerden gelen istekleri dinlememektedir. Bu, uygulama içeren uzak sistemlerin yayınladığı syslog log mesajlarını merkezi bir yere, yani merkezi bir log sunucusuna loglayamayacağı anlamına gelir. Bu nedenle log toplamaya başlamadan önce gereken özel argüman, syslog arka plan programına verilmelidir.

2.1.1.2 Syslog Mesajlarının Sınıflandırılması

Syslog mesajlarının, syslogd tarafından nasıl yönlendirileceğine karar vermek için iki özelliği vardır: tesis ve öncelik. Tesis, mesajın nerede üretildiği konusunda genel bir sınıflandırma yapmak için tasarlanmıştır. Tablo 2.1'de kullanılabilecek syslog tesisleri verilmektedir.

Tablo 2.1 Syslog Tesisleri

Tesis	Açıklama
auth	Kimlik doğrulama bilgisi
authpriv	Yalnızca ayrıcalıklı kişilerin görmesi gereken hassas kimlik doğrulama bilgisi
cron	Crontab'dan üretilen mesajlar
daemon	Herhangi bir arka plan programından gelen mesajlar
ftp	Dosya transfer protokolünden gelen mesajlar
kern	Kernel tarafından üretilen mesajlar
local0'dan local7'ye kadar	Özel uygulamalardan gelen mesajlar
mail	E-posta aktarım aracından gelen mesajlar
mark	Düzenli aralıklarla üretilen dâhili mesajlar
news	Ağ haber sisteminden gelen mesajlar
syslog	Syslog arka plan programından gelen mesajlar
user	Komut satırından gelen mesajlar
uucp	Uucp sisteminden gelen mesajlar

(Korrf ve Paco, 2005)

Bu tesislerin isimleri Unix/Linux dağıtımları arasında biraz farklılık gösterir ve bazı işletim sistemlerinde daha az veya daha fazla kategoriler bulunabilir. Örneğin; OSX'in *install* adında farklı bir tesisi daha vardır.

Tesisi uygulama programcısı seçer ve seçim konusunda herhangi bir sınırlama bulunmamaktadır. Bir programcı, programı bir e-posta programı olmasa bile *mail* tesisini kullanmayı seçebilir.

2.1.1.3 *Mark* Tesisi

Syslogd tarafından kullanılan ve *mark* olarak adlandırılan özel bir tesis vardır. Bu tesis aşağıdaki gibi düzenli aralıklarla bir işaret mesajı üretir. (Korff vd. 2015)

Feb 16 17:34:00 hostname -- MARK --

Mark tesisinin amacı, syslog arka plan programının kendisine herhangi bir mesaj gelmemiş olsa bile çalışmakta olduğunu doğrulamaktır. İşaret mesajları, bir sistemin çalışıp çalışmadığını denetlemek ve bir bilgisayarın ne zaman çöktüğünü belirlemek için kullanışlıdır. Çökme süresini yaklaşık olarak belirlemek için

çökmeden önceki son işaret mesajının saati kullanılabilir. Varsayılan aralık, komut satırından belirtilmezse yirmi dakikadır. İşaret mesajlarının doğru aralığını belirlemede bazı tereddütler vardır. Bir dakika gibi küçük bir değer, bir sunucunun ne zaman çalışır veya bozuk olduğuna ilişkin kesin zamanlamayı verecektir. Bununla birlikte, binlerce sunucu her dakika merkezi log sunucusuna işaret mesajları gönderirse, yalnızca bu iletiler için bile ağda çok fazla trafik yaşanılacaktır; ancak syslog'un gerçekten çalıştığını doğrulayabilmek için de belirli bir değerde işaret aralığına sahip olması gerekir.

2.1.1.4 Syslog Önceliği

Bir mesajın önceliği, mesajın önemini belirtmek için kullanılır. Tablo 2.2'de mevcut syslog öncelikler kümesi verilmektedir.

Tablo 2.2 Syslog Öncelikleri

Önem Derecesi	Anahtar Kelime	Açıklama
Acil	emerg	Sistem kullanılamıyor, panik durumu
Alarm	alert	Hemen harekete geçilmeli, veritabanı sorunu gibi
Kritik	crit	Kritik koşullar, donanım hatası gibi
Hata	err	Hata koşulları
Uyarı	warning	Uyarı koşulları
Bildirim	notice	Normal ama önemli koşullar
Bilgi	info	Bilgilendirici mesajlar
Hata Ayıklama	debug	Hata ayıklama mesajları

(syslog, 2016)

Herhangi bir olaya eklenen öncelik, uygulama programcısı tarafından seçildiğinden, önem derecesinin gerçekten anlamlı olduğunun garantisi yoktur.

Tesis ve öncelik birleşiminin pratik uygulaması, mesajların filtrelenmesinin ilkel bir formu olarak syslog arka plan programı tarafından kullanılır. Bu filtre *syslog.conf* dosyasında belirtilmektedir.

2.1.1.5 Syslog.conf Dosyası

Syslog.conf dosyası, aşağıda gösterilen biçimde bir veya daha fazla satır içermektedir.

<selector>... <tab> ... <action>

<selector> belirli satırın hangi mesaj türlerini uygulayacağını, *<action>* ise o mesaj ile ne yapılacağını belirtir.

<*selector*>, nokta ile ayrılan bir tesis ve önceliğin bir birleşimidir. Örneğin; *daemon.info*. Verilen bir seçici belirtilen tesisin tüm mesajlarını belirtilen önceliğe veya daha fazlasına eşleştirir. Yani *daemon.info* olarak verilen seçici, *daemon.info*'dan daha yüksek önceliğe sahip olan *daemon.crit* ile de eşleştirilmektedir. Çoklu tesislere tek bir öncelik virgüllerle ayrılarak verilebilir. Örneğin; *daemon,mail.info* ifadesi *daemon.info* ve *mail.info*'yu belirtir. Tüm tesislerin belirtilmesi için tesis adı yerine '*' kullanılabilir. Birden fazla seçici aynı satırda *';'* ile belirtilir. Örneğin; *daemon,info;mail.crit*. *none* anahtar kelimesi, belirli bir tesisin dışındaki her şeyi seçmek üzere '*' karakteri ile birlikte kullanılabilir. Örneğin; **.info;mail.none* ifadesi *mail* tesisinin mesajları hariç olacak şekilde *info* önceliği olan her şeyi seçecektir.

Seçici ve eylem, <*tab*> karakteriyle ayrılır. Syslogd'nin yeni sürümleri <*tab*>'a ek olarak <*space*>'i de kabul etmektedir.

Seçici için gerçekleştirilen eylem üç şeyden biri olabilir; mesajı bir dosyaya eklemek, mesajı başka bir bilgisayardaki syslogd'ye iletmek ve mesajı bir kullanıcının terminaline yazmak. Kullanılan en yaygın eylem, dosyanın tam yolunu belirleyerek mesajı bir dosyaya eklemektir.

mail.crit /var/log/eposta.log

Birçok syslogd sürümünde, syslogd'nin yazabilmesi için dosyanın bulunması gerekir. Dosya mevcut değilse syslogd dosyayı oluşturmaz. Dosya mevcut değilse, *touch <dosyaismi>* veya *cp /dev/null <dosyaismi>* komutu ile dosya oluşturulabilir.

Log mesajlarını başka bir sunucuya iletmek, ona bir "@" işareti eklenmiş bir sunucu adı vermek suretiyle yapılır.

**.crit @logserver.mydomain*

Logserver, bir bilgisayar adı veya bir IP adresi olabilir. Bir IP adresi belirtmenin avantajı, syslog arka plan programı başlatıldığında DNS çalışmıyorken bile mesajların iletilmesidir. Uzak bilgisayara gönderilen mesajlar UDP aracılığıyla gönderilir ve yanıt beklenmez. Bu nedenle uzak bilgisayarın bir arızası syslogd'nin yerel bilgisayarda çalışmasını engeller.

Üçüncü muhtemel eylem, mesajı bir kullanıcının terminaline yazmaktır. Mesajın görünmesi için kullanıcının oturum açmış olması gerekir. Bu işlem, kullanıcı adı belirterek yapılır:

mail.crit abc

Bu örnek, kritik ve daha yüksek olan öncelikteki tüm *mail* mesajlarının abc kullanısının sahip olduğu herhangi bir kullanıcı terminaline yazılmasına neden olacaktır. Burada kullanıcı adı olarak '*' belirtilmesi durumunda mail mesajları tüm kullanıcılara gönderilecektir:

*mail.crit **

Bazı syslog arka plan programlarının yapılandırma dosyası aracılığıyla kontrol edilebilecek ek özellikleri vardır. Örneğin; bazı syslogd'ler syslog mesajlarının log sunucusuna hangi bilgisayarlardan kabul edileceğini sınırlamaya izin verirler.

2.1.1.6 Syslogd Çıktısı

Syslogd, mesajları yeni satırlarla sonlandırılmış ASCII metni olarak yazar. Bu nedenle dosyaları okumak için özel bir izleyici gerekmez, herhangi bir metin görüntüleyici gösterebilir. Bununla birlikte, log dosyalarını incelemek için "vi" veya başka bir düzenleyici kullanılması önerilmez. Log dosyaları gerçekten büyük olabilir; ancak daha da önemlisi log dosyalarını bir metin düzenleyiciyle okurken, log dosyasını yanlışlıkla değiştirme riski oluşur. Bu da yasal veya iş amaçlı log verilerine ihtiyaç duyulduğunda kabul edilemez bir durumdur. Dosyayı değiştirmek planlanmadığı sürece, bir metin düzenleyiciyi bir dosyada kullanmamak iyi bir sistem yönetimi uygulamasıdır.

2.1.1.7 Syslog Protokolü

Uzun yıllar boyunca, syslog protokolü için kullanılan RFC standardı RFC3164'tür. (The BSD syslog Protocol, 2001) Günümüzde ise kullanılan standart RFC5424'tür. (The Syslog Protocol, 2009) RFC5424, eski syslog protokolünün bir revizyonudur. Protokoldeki en büyük değişikliklerden biri RFC3339'da problemlere neden olan zaman damgalarının belirtilmesidir. (Date and Time on the Internet: Timestamps, 2002) Eski protokolde zaman damgası konusunda fazla bir şey belirtilmemektedir. Alınan log mesajı ay, gün, saat ve saniye gibi minimum bilgileri

içermekte olup, yıl ve zaman dilimi bilgileri bulunmamaktadır. Bu şekilde olması analiz açısından zor bir durumdur. Ayrıca RFC5424'te, otomatik log analizini önemli ölçüde kolaylaştıracak olan ad=değer çiftleri gibi yapısal veriler bulunmaktadır.

2.1.2 SNMP

SNMP, cihazları sorgulamak ve yapılandırmak için kullanılan bir protokoldür. SNMP tuzakları (trap) ve bildirimleri (notification), belirli bir olay oluştuğunda bir cihaz tarafından üretilen bir SNMP mesajıdır. SNMP protokolü bir bütün olarak loglama sistemi olmadığı halde, SNMP tuzakları veya bildirimleri log mesajlarının türleri olarak kabul edilebilir. Birçok ağ cihazı syslog yoluyla olay bilgilerini gönderebilirken, bazı eski cihazlar gönderemezler. Dolayısıyla SNMP tuzakları ve bildirimleri, başka türlü toplanılamayan cihazlardan olay bilgisi alma yöntemidir ve bazı durumlarda SNMP yoluyla gönderilen bilgi türü syslog üzerinden gönderilenlerden farklıdır. SNMP'nin SNMPv1, SNMPv2 ve SNMPv3 olarak bilinen birden çok sürümü vardır.

2.1.2.1 Yöneticiler ve Ajanlar

SNMP tarafından yönetilen cihazlar genellikle bir ağ yönetim istasyonu (NMS) tarafından kontrol edilir. NMS, cihazları periyodik olarak yoklar, durum bilgisini sorgular, gerektiğinde yapılandırma değişiklikleri gönderir. NMS ayrıca tuzakları veya bildirimleri dinler. Bu şekilde NMS, merkezi bir log toplayıcıya benzer şekilde işlev görür.

Syslog'da olduğu gibi bir şey, SNMP tuzaklarını dinlemek zorundadır. Alıcı bir NMS veya kullanılan Unix dağıtımındaki SNMP arka plan programı olabilir. Net-SNMP, en popüler açık kaynaklı SNMP araç setidir. Kullanışlı komut satırı araçları ile Unix ve Windows'ta çalışan bir SNMP tuzak arka plan programı (snmpd) içerir.

2.1.2.2 SNMP Tuzakları ve Bildirimleri

SNMP tuzakları SNMPv1 protokolünün, SNMP bildirimleri ise SNMPv2 ve SNMPv3 protokollerinin bir parçasıdır. Tuzaklar ve bildirimler arasındaki en önemli fark, bildirimlerin alıcının bir bildirimi gönderene geri gönderme yeteneğini içermesidir. Cihaz, hangi olayların mesaj ürettiği ve mesajların nereye gönderileceği de dâhil olmak üzere mesajı oluşturmak üzere yapılandırılmalıdır.

Syslog'da olduğu gibi SNMP, UDP üzerinden uygulanır ve bu nedenle Syslog ile aynı güvenilirlik sorunlarından muzdariptir. SNMPv2 bildirimleri, bir alındı bilgisinin alıcıdan geri gönderilmesini sağlar, bu şekilde tek yönlü UDP iletilerinden daha güvenilir bir mesaj teslimatı sağlar.

Ayrıca syslog'da olduğu gibi SNMPv1 tuzakları, düz metin olarak gönderilir ve kimliği doğrulanmaz. Bu nedenle, aynı sahtecilik (spoofing) saldırılarına tabidirler. SNMPv2 bildirimleri de düz metin olarak gönderilir. SNMPv3 hem gönderici ve hem de alıcı tarafında fazladan CPU döngüsü oluşturmasına rağmen sahtecilik saldırılarına karşı koruma sağlayan isteğe bağlı bir kimlik doğrulama sertifikasına sahiptir.

2.1.2.3 SNMP Get

SNMP protokolü, bir cihazdan veya sistemden bilgi almaya yarayan *get* işlemi sağlar. Ne tür bilgi alınacağı, söz konusu cihazın neyi sağladığına bağlı olarak değişir. Ağ yönetiminde, tuzak yönlendirmeli çekme (*trap-directed polling*) adı verilen bir kavram vardır. Bu, bir çeşit SNMP tuzağı alındığında, yeni alınan tuzağı ilişkilendirmek veya doğrulamak için SNMP get kullanarak cihazın çekme işlemini başlatmak demektir.

2.1.2.4 SNMP Set

SNMP *set*, uzaktaki bir sistemde bir şeyin değerini değiştirmeyi sağlar. Örneğin; ağ köprüleri, SNMP'nin bir geçiş portunu yukarı ve aşağı ayarlayabilmesi için kullanabileceği belirli bir tanımlamayı sağlar. Bunu yapabilmek için belirli porta bağlı makinenin MAC adresinin bilinmesi gereklidir.

2.1.2.5 SNMP ile İlgili Sorunlar

Syslog mesajlarıyla ilgili büyük bir sorun standartlaştırılmış bir biçimin bulunmamasıdır. Bu ise, ABC üreticisinin log mesajlarının XYZ üreticisinden farklı olacağı anlamına gelir; ancak SNMP ile durum biraz daha iyidir. SNMP, MIB (Management Information Base)'i kullanır. MIB, belirli bir sistemin hangi tuzakları veya bildirimleri desteklediğinin bir tanımlamasıdır.

MIB'ler iyi biçimlendirilmiş, anlaşılır mesajlar vermeye yönelik güçlü bir altyapı sağlasa da üreticiler bunu nadiren desteklemektedir. MIB, cihaz veya sistem tarafından gönderilen tuzak veya bildirimle eşleşmeyebilir. SNMP uygulamasında

değişiklikler yapılır ancak MIB güncelleştirilmezse, tuzakların nasıl göründüğünü anlamak için MIB kullanılmaya başlandığında karışıklığa sebep olabilir.

Bir üretici MIB oluşturmak için zaman harcamak yerine, serbest biçimli bir metin dizesi olarak içinde tek bir değişken bulunan bir tuzak oluşturabilir. Sonuç, syslog mesajlarını ayrıştırmak gibi zordur. Bu nedenle SNMP, log bilgilerini toplamak için en iyi çözüm değildir; ancak bazı cihazlardan bu bilgileri alabilmenin tek yoludur.

2.1.3 Windows Olay Günlüğü

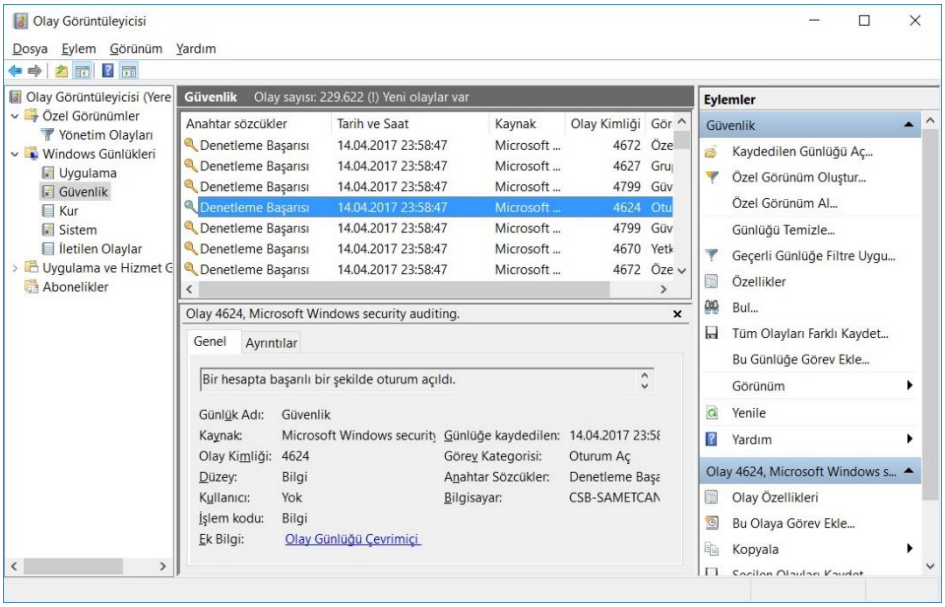
Microsoft'tun log hazırlama ve toplama sistemine Olay Günlüğü adı verilir. Olay Günlüğü öncelikli olarak iki tür log toplamak ve gözden geçirmek için kullanılır:

- Windows Günlükleri
- Uygulama ve Hizmet Günlükleri

Windows Günlükleri; Uygulama, Güvenlik ve Sistem'i kapsar. Güvenlik'te oturum açma, oturum kapatma, kaynak erişimi (paylaşımlar, dosyalar vb.) loglanır. Uygulama ve Hizmet Günlükleri'nde uygulamalar, aktarım durumunu, hataları ve diğer kayda değer öğeleri yazabilir.

Windows'ta Olay Görüntüleyicisi'ni çalıştırmak için: "Denetim Masası => Sistem ve Güvenlik => Yönetimsel Araçlar => Olay Görüntüleyicisi" yolu izlenir. Şekil 2.1'de Olay Görüntüleyicisi gösterilmektedir.

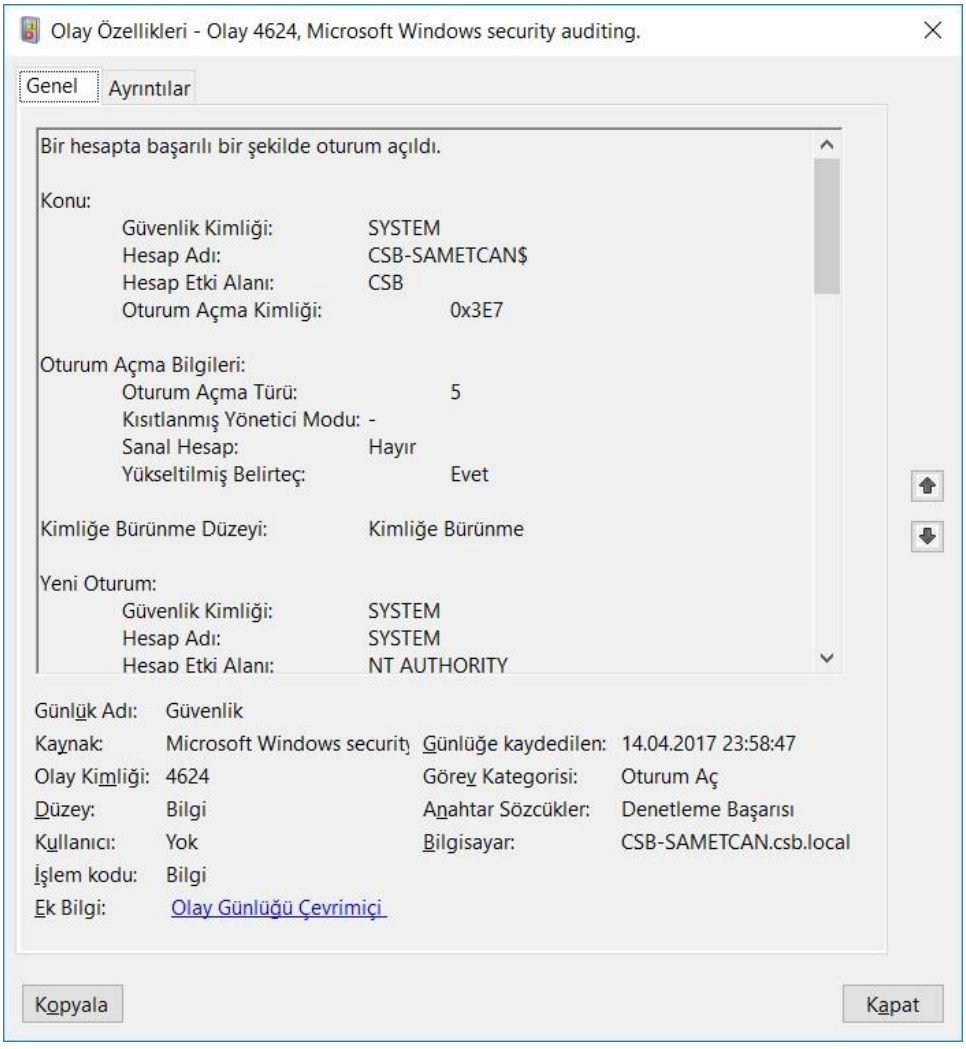
Şekil 2.1 Windows Olay Görüntüleyicisi



Olay Görüntüleyicisi ekranının sol bölmesinde çeşitli log mesajı türleri vardır. Ortada, seçilen türe ait log mesajları bulunur. Ortadaki log mesajının altında ilgili log mesajının ayrıntıları yer alır. Log mesajı çift tıklatılırsa Şekil 2.2'deki gibi bir pencere açılarak burada seçilen log mesajının ayrıntıları gösterilmektedir. Log mesajının türüne bağlı olarak şunlar görülebilir:

- Olay Kimliği
- Hesap Adı
- Etki Alanı
- Kaynak (dosya veya dizin erişimi durumunda)
- Durum (isteğin başarılı olup olmadığı)

Şekil 2.2 Bir Windows Olayının Ayrıntıları



2.2 Log Kaynaklarının Sınıflandırılması

Aşağıdaki bölümlerde, log verileri üreten uygulamaların ve sistemlerin örnekleri verilmektedir.

2.2.1 Güvenlikle İlgili Host Logları

Bu kategori, işletim sistemi bileşenleri tarafından üretilen logları, çeşitli ağ servislerinin loglarını ve bunların yanı sıra sistem üzerinde çalışan diğer uygulamaların loglarını kapsar. Mesajların birçoğu performans izleme, denetim veya sorun giderme nedenleriyle üretilirken, bunların büyük bir çoğunluğu güvenlik için faydalıdır.

2.2.1.1 İşletim Sistemi Logları

İşletim sistemleri çok fazla türde log mesajları üretmektedir. İşletim sistemleri tarafından üretilen bazı güvenlik mesajlarının türleri aşağıda verilmektedir.

- Kimlik doğrulama: Kullanıcının oturum açması gibi. Aşağıda SSH (Secure Shell) arka plan programıyla uzak kullanıcılarla ilgili kimlik doğrulaması yapan bir Linux syslog satırı verilmektedir.

Feb 18 09:55:21 abc sshd2[168]: User abc, coming from 72.127.65.155, authenticated.

- Sistemin başlatılması, kapatılması ve yeniden başlatılması. Aşağıda sistem kapanmasına ilişkin bir Linux syslog satırı verilmektedir.

Apr 9 02:16:20 localhost shutdown: shutting down for system reboot

- Hizmetlerin başlatılması, kapatılması ve durumunun değişikliği. Aşağıda sendmail arka plan programının başlatılmasıyla ilgili bir Linux syslog satırı verilmektedir.

May 7 15:08:11 solinst sendmail[359]: [ID 146987 mail.info] starting daemon (9.12.7+Thu): SMTP+queueing@02:34:00

- Hizmetin çökmesi. Aşağıda istemsiz olarak kapatılan FTP sunucusu ile ilgili bir Linux syslog satırı verilmektedir.

Apr 11 13:45:34 xyz ftpd: service shut down

- Çeşitli durum mesajları. Aşağıda bir zaman senkronizasyon arka plan programı (NTPD) ile ilgili bir Linux syslog satırı verilmektedir.

Feb 19 16:23:42 localhost ntpd[375]: precision = 12 usec

Genel olarak işletim sistemi mesajları, iki temel nedenden dolayı güvenlikle ilgili olarak kabul edilir:

1. Saldırı tespiti için kullanışlıdır; çünkü başarılı ve başarısız saldırılar genellikle loglarda benzersiz iz bırakır. Sunucu Tabanlı Saldırı Tespit Sistemleri (HIDS) ve Bilgi Güvenliği ve Olay Yönetim Sistemleri (SIEM) log mesajlarını toplayarak geçmiş ve yaklaşan tehditler hakkında kararlar verebilir.
2. Olaya müdahale için faydalıdır; çünkü çeşitli güvenlik önlemlerinin varlığına rağmen başarılı saldırılar meydana gelmektedir. Saldırı bulmacasının birbirine bitişik parçalarının bir araya getirilmesine imkân sağlaması açısından loglar, olaya müdahale için oldukça önemlidir.

2.2.1.2 Ağ Arka Plan Programı Logları

Ağ arka plan programları genellikle aşağıdaki kategorilerin güvenlikle ilgili mesajlarını loglar.

- Hizmet bağlantısı kuruldu. Aşağıda "abc" uzak kullanıcısı tarafından bir POP3 e-posta arka plan programına yapılan başarılı bir bağlantıyı gösteren bir Linux syslog satırı verilmektedir.

Feb 26 16:12:56 abc popper[726]: (v4.0.5) POP login by user "abc" at (72.127.65.155) 72.127.65.155

- Sunucuya bağlantı başarısız oldu. Aşağıda bir telnet servisine yapılan başarısız bir bağlantıyı gösteren bir Linux syslog satırı verilmektedir.

Feb 26 17:30:24 abc xinetd[513]: FAIL: telnet libwrap from =195.46.10.32

- Bağlantı kuruldu; ancak erişim izni verilmedi. Aşağıda SSH sunucusuna yapılan başarısız bir bağlantıyı gösteren bir Linux syslog satırı verilmektedir.

Feb 27 09:54:10 abc sshd2[1245]: connection lost: 'Connection closed.'

- Çeşitli hata mesajları. Aşağıda bir sendmail arka plan programının bir istemciyle konuşmaya devam etmesinin başarısız olduğunu gösteren bir Linux syslog satırı verilmektedir.

Feb 27 10:32:15 xyz sendmail[1476]: tSRAbEd16453: lost input channel from [10.51.165.6] to MTA after rcpt

- Çeşitli durum mesajları. Aşağıda başarılı bir e-posta transferini gösteren bir Linux syslog satırı verilmektedir.

Feb 27 12:05:45 abc sendmail[1548]: tSRAbEd16453: from=<hkjofheod@osdowjdf.com>, size=0, class=0, nrcpts=2, proto=SMTP, daemon=MTA, relay=[10.21.145.9]

Ağ arka plan programının logları, işletim sistemi logları kadar faydalıdır. Bunlar genellikle aynı yere loglarlar. Örneğin; Windows ve Unix'te aynı loglama mekanizması yaygın olarak kullanılır.

Sisteme giriş yerlerinin en yaygın bulunduğu ağ arka plan programları uzaktan izlenir ve saldırıların çoğu bunları hedef alınır. Bu nedenle, güçlü loglamaya sahip olmak bu ortamda çok önemlidir.

2.2.1.3 Uygulama Logları

Uygulamaların logladığı mesajlar çok çeşitlidir. Uygulama log türleri aşağıdaki gibi tek bir listede toplanabilir.

- Uygulama kullanıcı etkinliği
- Ayrıcalıklı kullanıcı etkinliği
- Rutin ancak kritik etkinlik
- Yeniden yapılandırma

2.2.2 Güvenlikle İlgili Ağ Logları

Bu kategori, ağ altyapısı tarafından oluşturulan ağ loglarını kapsar. Yönlendiriciler ve anahtarlar, içlerinden geçen trafikle ilgili işlemlerde çok çeşitli log mesajları üretirler.

2.2.2.1 Ağ Altyapısı Logları

Ağ altyapısı, ağları oluşturan ve masaüstü bilgisayarlar ile sunucuları birbirine bağlayan yönlendiriciler, anahtarlar ve diğer cihazları içerir. Bu cihazlardan gelen loglar güvenlik açısından kritik bir rol oynar.

Mesajların yer aldığı en yaygın kategoriler şunlardır:

- Girişler ve çıkışlar
- Servise kurulan bağlantı
- Gelen ve giden baytlar

- Yeniden başlatmalar
- Konfigürasyon değişiklikleri

2.2.3 Güvenlik Host Logları

Bu kategori, bir hostta çalışan güvenlik görevine sahip uygulamalardan gelen host loglarını kapsar. Her durumda güvenlik açısından alakalı olabilecek ya da olamayacak log kayıtlarının tutulmasının aksine buradaki güvenlik logları, saldırılar, müdahaleler, zararlı şeylerin bulaşması ile ilgili olduğundan dikkat edilmesi gerekir; ancak pek çok durumda güvenlik donanımları çeşitli tiplerde yanlış alarmlar (yanlış pozitifler gibi) üreterek gerçeği yansıtmayabilirler.

3. LOG SAKLAMA

Logların saklanması ve loglara hızlı bir şekilde erişilerek analiz edilmesi kurumlar için kritik bir konudur. Küçük ve orta ölçekteki kurumlar tarafından tutulan log kayıtları, terabayt veya petabayt veri boyutuna hızlı bir şekilde ulaşabilir. Bu veriler, bir kurumda çeşitli biçimlerde saklanabilir. Bu bölümde çevresel faktörler ve uyumluluk gereksinimleri dikkate alınarak saklama stratejileri incelenecektir.

3.1 Log Tutma Politikası

Bu bölümde log tutma politikası geliştirilirken ihtiyaç duyulacak bilgiler verilecektir. Kurum için tasarlanan tutma politikasında log verileri için saklama türü, boyut, maliyet, erişim hızı, arşivleme ve imha gereksinimleri hakkında yapılması gereken kararların yer alması gerekir.

Aşağıda açıklanan faktörleri temel alarak bir tutma stratejisi oluşturma süreci ele alınacaktır. Kurumun ihtiyaçlarına göre organizasyondaki güvenlik, uyumluluk ve işletme yönetimi gruplarından bir dizi paydaş, bu sürece dâhil olmalıdır. Log tutma politikası geliştirilirken aşağıdaki hususlar gözden geçirilmelidir.

1. **Uygulanabilir uyumluluk gereklilikleri değerlendirilmelidir:** Günümüzde pek çok endüstride güçlü bir dizi uyumluluk gereksinimi bulunmaktadır. Örnekler arasında, bir yıl gibi belirli bir log tutma süresi olan Ödeme Kartı Endüstrisi Veri Güvenliği Standardı (PCI DSS), belirli log türleri için belirli zamanlardan bahseden Kuzey Amerika Elektrik Güvenilirliği Şirketi (NERC) kuralları yer alır. Diğer düzenlemeler, belirli türdeki logları tutmayı gerektirir; ancak bir süre belirtmez.
2. **Kurumun risk durumu gözden geçirilmelidir:** Ağın çeşitli bölümleri için oluşan iç ve dış riskler, tutma süresini artırır. Logların süresi ve önemi, bir kurum için bu risk alanlarının her birinde çeşitlilik gösterebilir. İç tehditleri araştırmak için loglar kullanılmaya odaklanılırsa, bu tür olaylar genellikle yıllarca keşfedilemediğinden log tutma süresi uzun olmalıdır.
3. **Çeşitli log dosyalarına ve oluşturulan logların boyutlarına bakılmalıdır:** Güvenlik duvarları, sunucular, veritabanları gibi çeşitli sistemler düşünüldüğünde yalnızca ihtiyaca değil, tipik log hacmine ve üretilen her bir log kaydının boyutuna ve türüne bakılmalıdır. Her cihaz ya

da uygulamanın türünden elde edilebilecek log hacmi konusunda geniş bir çeşitlilik vardır. Örneğin; güvenlik duvarlarından gelen loglar, fazla miktarda log oluşturur ve uzun vadeli tutma gereksinimlerinden dolayı bu log verileri yalnızca 30 gün boyunca saklanır. Bununla birlikte, PCI DSS gibi bir kuruluşun uyumluluk gereksinimleri ve güvenlik duvarının kritikliği, daha uzun bir tutma süresinin gerekli olup olmadığını belirlemek için yakından değerlendirilmelidir. Ayrıca, özel uygulamalar ve desteklenmeyen işletim sistemleri gibi loglarını analiz etmek için gerekli araçlar bulunmayan log kaynakları da olabilir.

4. **Mevcut saklama seçenekleri gözden geçirilmelidir:** Log saklama seçenekleri, disklerden, DVD'lerden, kasetten, RDBMS'den, loga özgü saklama ve bulut tabanlı saklama seçeneklerinden oluşur. Makul bir süre içerisinde doğru log kayıtlarına erişebilmek için fiyat, kapasite ve erişim hızıyla ilgili konularda neyin önemli olduğuna karar verilmelidir. Kaset ucuz bir seçenek olarak bilinir; ancak saklanan kayıtların etkin bir şekilde aranamaması problemi vardır ve uygun kaseti bulmak ve analiz amacıyla verileri geri yüklemek için insan müdahalesi gerekebilir. Tutma süresi gereksinimini karşılamak için ne sıklıkla yeniden yazılması gerekeceği konusunda medyanın ömrünün de dikkate alınması gerekir. Örneğin; yazılabilir ucuz CD'ler veya DVD'ler için yedi yıl muhtemelen uzun bir süredir. Teknolojideki çağ ve geçerlilik de kasetleri ve diskleri okumak için sürücü satın alma olanağını zorlaştırabilir. (Chuvakin vd. 2013)

Tablo 3.1'de örnek bir log tutma politikası gösterilmektedir.

Tablo 3.1 Örnek Log Tutma Politikası

Tür	Ağ	Depolama Katmanı	Tutma Süresi
Saldırı Önleme Sistemi (IPS)	DMZ	Çevrimiçi	3 ay
Güvenlik Duvarı	DMZ	Çevrimiçi	1 ay
Sunucular	İç	Çevrimiçi	3 ay
Hepsi	DMZ	Arşiv	3 yıl
Kritik	İç	Arşiv	5 yıl
Diğerleri	İç	Arşiv	1 yıl

Bir kurumda logları tutma süresini ve alan gereksinimlerini belirlemek zor olabilir. Bu bölümün geri kalanında, karşılaşılacak log biçimlerinin türleri ve kurumun politikasına uygun olarak log dosyalarını korumak ve saklamak için kullanılan stratejiler incelenecektir.

3.2 Log Saklama Biçimleri

Bu bölümde, birkaç log saklama biçimi ele alınacaktır. Ağ cihazları, uygulamalar ve işletim sistemleri birçok farklı biçimde log üretir ve çoğu durumda loglar metin tabanlı, ikili veya sıkıştırılmış biçimde saklanır.

3.2.1 Metin Tabanlı Log Dosyaları

Günümüzdeki programlama dillerinde metin tabanlı logların oluşturulmasını kolaylaştıran birçok altyapı olması ve sistemlerin bu tür log dosyalarını üretmesinin düşük maliyetli olması nedeniyle en yaygın kullanılan loglama türüdür. Metin tabanlı dosyalara loglayan sistemlerin popülerliği ve bolluğu, bu biçimin pek çok faydasının bir sonucudur. Metin tabanlı loglamanın faydaları şunlardır:

- Bir uygulama için metin tabanlı bir log dosyasına yazmak, CPU ve G/Ç kaynakları açısından çok ucuz bir işlemdir.
- Biçimi, genellikle insan tarafından okunabilir ve Unix/Linux işletim sistemlerinin birçok çeşidinde yer alan *grep* ve *awk* gibi yaygın metin işleme araçları ile incelenebilir.
- Operasyon ve güvenlik ekiplerinin logları ortak bir şekilde merkezileştirip ayrıştırma ve daha güçlü bir log yönetimi sistemi oluşturma becerisini kolaylaştıran syslog gibi metin tabanlı yaygın log biçimleri vardır.

3.2.1.1 Düz Metin Dosyaları

Düz metin dosyaları, ortak deseni takip edebilen veya serbest form olarak şema içermeyen düz bir dosyadan oluşur. Tipik olarak bir sistem, yeni bir log dosyası oluşturur ve diskte yeterli alan olduğu sürece bu dosyaya eklemeye devam eder. Log rotasyonu işlemi ile sisteme, yeni bir log dosyasına başlaması ve mevcut dosyayı arşivlemesi öğretilir. Bu biçimde, dosyanın başında en eski olaylar ve sonunda ise en güncel olaylar kronolojik sırayla yer alır.

Birçok sistem tarafından benimsenen en yaygın biçimlerden birisi syslog biçimidir. Syslog aracılığıyla gönderilen loglar, gönderilmeden önce diskte bulunur ve çok basit bir biçime sahiptir. Aşağıda syslog biçiminde bir log mesajı gösterilmektedir:

Apr 11 09:45:13 can-pc kernel[0]: macx_swapon SUCCESS
Apr 11 09:48:24 can-pc kernel[0]: 21.489565: Use hw queue 8 for CAB traffic
Apr 11 10:05:23 can-pc kernel[0]: Previous Shutdown Cause: 5
Apr 11 10:11:16 can-pc kernel[0]: nstat_lookup_entry failed: 2
Apr 11 10:45:05 can-pc kernel[0]: IOSurface: buffer allocation size is zero

Düz metin dosyalarındaki log verisinin uzun vadeli saklanmasının yararı, bu biçimdeki veriyi okumak ve incelemek için kullanılan araçların çok olmasıdır. Her platformda bu biçime kolayca erişen ve okuyabilen birçok araç bulunmaktadır. Bu ise verileri 5 ya da 10 yıl sonra okumak ve incelemek gerekli olduğunda ve olayları işleyebilecek ve ilişkilendirebilecek araçlara ihtiyaç duyulduğunda, önemli bir özellik haline gelmektedir.

3.2.1.2 İndekslenmiş Düz Metin Dosyaları

Kurumların düz metin dosyalarında yaşadığı zorluklardan birisi, yönettikleri platformlardaki anlamlı eğilimleri bulmak için düz metin dosyasından hızlı bir şekilde sorgulama ve sıralama yaparak anahtar öğeleri alabilmek istemeleridir. Log dosyaları, terabayt ve petabayt verilere hızla dönüştüğü için geleneksel bir *grep, awk* ve metin tabanlı arama araçlarını kullanmak sabır istemekte ve zor bir süreç haline gelmektedir.

İndekslenmiş düz metin dosyaları, log dosyalarındaki verileri organize etmenin bir yoludur, böylece logların anahtar öğeleri daha hızlı sorgulanabilir. Bunlar, kurumların rapor üretmek ve tutma süresi dolduktan sonra log verilerini imha etmek için ihtiyaç duyduğu yapıyı da sağlar. İndekslenmiş düz dosyalar, verilerin hızlıca eklenmesi ve okunabilir bir biçimde kalması gibi düz metin dosyalarının birçok avantajına da sahiptir. Düz metin dosyalarının log tutma politikasına göre indekslendiği örnek dizin yapısı aşağıda verilmektedir:

/logs/debug/2016/Sep
/logs/debug/2016/Oct
/logs/debug/2016/Nov
/logs/debug/2016/Dec

Yukarıdaki dizin yapısında logların, alındığı yıl ve aya dayalı olarak düzenlendiği görülmektedir. Aralık 2016'daki hata ayıklama logları incelenmek istenirse, "/logs/debug/2016/Dec" dizinine gidilmesi gerekir.

3.2.2 İkili Dosyalar

İkili log dosyaları, okumak ve işlemek için özel araçlara ihtiyaç duyulan makine tarafından okunabilir log dosyalarıdır. İkili log dosyalarının yaygın bazı örnekleri, Microsoft Internet Information Server (IIS) logları ve Windows Olay Günlüğü'dür. Özel uygulamalar içeren birçok ortamda log dosyaları, Genişletilmiş İkilik Kodlu Ondalık Değişim Kodu (EBCDIC) gibi ikili veya makineye özgü biçimlerde kodlanabilir ve aynı zamanda bu kodları çözmek ve okumak için donanım platformlarında araçlar gerekebilir.

İkili log dosyalarının uzun vadeli saklanması, kuruma çeşitli zorluklar getirir. İkili log dosyalarını kendi biçiminde saklamadan önce göz önüne alınması gereken öğeler şunlardır:

- Gelecekte 5 ya da 10 yıllık log dosyalarını okumak için araçların bulunup bulunmadığıdır. Eski IIS 6.0 web sunucusu loglarını okumak ve adli işlemleri gerçekleştirmek için bir Windows NT sunucusunu yaklaşık 10 yıl bulundurmak mümkün değildir.
- İkili log dosyaları, disk alanının kullanımı bakımından verimli olma eğilimindedir. Bununla birlikte, iyi sıkıştırılamazlar. İkili dosyalar sıkıştırıldığında orijinal boyutlarının yüzde 90'ı kadar yer kaplar. Buna karşılık, metin tabanlı dosyalar sıkıştırıldığında orijinal boyutlarının yalnızca yüzde 10'unu kadar yer kaplar. Metin tabanlı loglamayla karşılaştırıldığında ikili dosyalar, saklama alanı ihtiyaçlarını artırabilir.

3.2.3 Sıkıştırılmış Dosyalar

Log üreten sistemlerin çoğu, loglar belirli bir boyuta ulaştığında veya günlük, haftalık, aylık olarak yapılandırılmış bir zaman aralığına ulaşıldığında, genellikle yeni bir log dosyasına başlar. (Kent ve Souppaya, 2006) Önceki log dosyası genel olarak yeniden adlandırılır ve sistemin disklerinde sıkıştırılmamış bir biçimde arşivlenir, böylece hâlâ kolayca erişilebilir ve sorgulanabilir. Bir log dosyası yaşlandıkça, bu log dosyası günlük raporlar ve log inceleme görevleri için daha az alakalı hale gelir; ancak uyumluluk tutma süresinin sağlanması ve adli bilişim işlemlerinde hâlâ kritik önem taşır. Çoğu durumda, log dosyalarına hızlı bir şekilde erişilmek istenecektir; ancak ilave loglama ve diğer sistem operasyon gereksinimleri için yerden tasarruf etmek

gereklidir. Sistemdeki log dosyalarının sıkıştırılması, bu gereksinimleri karşılayan ve disk alanından tasarruf ettiren bir mekanizmadır.

Unix/Linux sistemlerinde, tutulan log dosyalarının sayısını yönetmeye ve ayrıca logları sıkıştırmaya yardımcı olan kullanışlı bir *logrotate* programı vardır. Aşağıda syslog mesajlarının rotasyonu için *logrotate*'in örnek bir yapılandırma dosyası gösterilmektedir. (Troan ve Brown, 2002)

```
compress
/var/log/messages {
   rotate 5
   weekly
   postrotate
   /sbin/killall -HUP syslogd
   endscript
}
```

Unix/Linux sistemleri, birçok standart araca eşdeğer sıkıştırma araç setine sahiptir. *Zgrep* ve *zcat* araçları, tıpkı *grep* ve *cat* gibi sıkıştırılmamış dosyalar üzerinde çalışabilecekleri gibi, sıkıştırılmış dosyalardan veri okuyabilir ve bunlardan veri alabilir. Bununla birlikte, sıkıştırılmış dosyalarla çalışamayan pek çok araç vardır. Bu araç setlerinden de faydalanabilmek için, sıkıştırılmış dosyanın çözülmesi gerekir ve analiz ihtiyacını karşılamak için bir alan ayrılmalıdır. Bir sıkıştırma biçimi seçerken eskimeyi önlemek için uzun yıllar kullanılmış ve birden fazla platformda bulunan bir sıkıştırma biçimi seçmek iyidir. Unix/Linux'da *tar* ve *zip* biçimleri uzun bir kullanım geçmişine sahiptir ve *zip* biçiminde sıkıştırma dosyaları Windows'ta da yaygın olarak kullanılmaktadır. İkili dosyalardaki gibi, gelecekte log verilerine erişmek ve sıkıştırmayı çözmek için kullanılan araç setlerinin eskimesini sınırlamak gerekir.

3.3 Veritabanında Log Saklama

Buraya kadar ele alınan saklama tekniklerinden çoğu, log inceleme için sistemlere ve özel araç setlerine doğrudan erişilmesini gerektirir. Bu, hızlı ve verimlidir; ancak çoğu durumda özet raporlar oluşturma, filtreleme ve sunucular arasındaki log verilerini ilişkilendirme becerisini ciddi şekilde sınırlamaktadır. Log bilgilerini veritabanına yazmak, log bilgilerine ihtiyaç duyan paydaşlara hızlıca aranabilen ve sorgulanabilen bir biçimde erişim sağlamakta ve log inceleme sürecinde arayüz araçlarının kurulumunu ve kullanılmasını kolaylaştırmaktadır.

Log verilerini veritabanında saklamanın avantajları şunlardır:

1. Log kayıtlarını hızlı bir şekilde aramak ve almak için standart SQL sorguları kullanılabilir.
2. Veritabanlarını sorgulamak için birçok standart araç bulunmaktadır. Bu araçlar, log verilerinin özel bilgi ve erişim gerektiren her bir sistemde platforma özgü araçları kullanmadan ortak bir araç seti aracılığıyla veri almak için sorgulama yapılmasına izin verir.
3. Veritabanlarındaki verilerle çalışmak amacıyla pek çok programlama dili, log verilerinin gerçek zamanlı olarak görüntülenmesi ve analizi için arayüz araçlarının geliştirilmesini sağlar.
4. Veritabanı sistemleri, verilere erişmek için güçlü bir kullanıcı erişimi ve izin sistemleri içerir.
5. Veritabanı sistemleri, halihazırda kurumun yedekleme ve kurtarma planının bir parçasıdır.

Log verilerini veritabanında saklamanın dezavantajları ise şunlardır:

1. Log mesajlarını bir veritabanına yazarken önemli bir maliyet oluşur. Veritabanına veri yazmak, ağ gecikmesi, veritabanında SQL'in ayrıştırılması, indeks güncelleştirmeleri ve bilgilerin diske işlenmesi nedenleriyle diskteki bir metin dosyasına yazmaya göre yavaş olacaktır.
2. Hızlı arama ve alma işlemlerini gerçekleştirmek için gereken indeks dosyalarının sayısı ve verileri veritabanında sıkıştırmak için kullanılabilecek sınırlı seçenekler nedeniyle log saklamak için gereken disk alanı daha yüksek olacaktır.
3. Veritabanı sistemleri, kurumda birçok amaçla kullanıldığından dolayı bir veritabanının devre dışı kalması, bakım veya yükseltme durumlarında loglama sistemleri veya diğer iç sistemler veri kaybı riskine maruz kalabilir.
4. Tutma politikasına dayanarak log girdilerinin gerekli olmadığı durumlarda verilerin yok edilmesi problemli olabilir. Düzgün planlanmamış veya bölümlendirilmemiş log verilerinin silinmesi çok uzun sürebilir. Log verileri genellikle çok büyüktür ve log verilerinin imhası, veritabanı sistemine milyonlarca satırın kaldırılmasını ve kaldırılan veriler için tüm indekslerin güncellemesini söyleyebilir.

3.3.1 Saklama Hedefleri

Bahsedilen birtakım dezavantajlardan kaçınmanın anahtarı, verileri bir veritabanına taşırken saklama hedefleri geliştirmektir. Saklama gereksinimlerini kurumun log tutma politikasıyla eşleştirebilmek için bir plan tanımlanmalıdır. Önceden hazırlanmış iyi bir plan gelecekteki bakım arızalarını, veri geri yüklemelerini ve yeniden yapılandırma ile gelecekteki operasyon kaynak ihtiyaçlarını azaltacaktır.

3.3.1.1 Neler Saklanacak

Veritabanı log inceleme ve analiz için merkezi depo ve çevrimiçi araç olarak kullanıldığında, tüm log girdilerinin ve log alanların veritabanında saklanılması istenecektir. Rsyslog ve syslog-ng gibi bazı syslog türleri, önceden oluşturulmuş veritabanı şemaları ve yapılandırma seçenekleri ile birlikte gelir ve syslog'u doğrudan veritabanı saklama alanına aktararak syslog mesajının tüm alanlarını saklar. Yaygın olarak kullanılmayan uygulamaların loglarını veritabanında saklamak için yardımcı programlar yazmak ve veritabanı şeması geliştirmek gerekebilir. Bütün alanları tutmak için gösterilen çabalar, analiz yapmak için kritik bir sistem ihlali gözden geçirilirken kritik bilgilerin eksik olması gibi bir durumu önler.

Birçok kurumda, birincil log saklama birimi olarak veritabanının kullanılmasıyla log güvenliği veya altyapısı açısından çok fazla dezavantaj bulunmaktadır. Hibrit depolama yaklaşımı, orijinal log verileri kaynak sistemde veya merkezi bir syslog sunucusunda tutulduğu durumlarda en iyi sonucu verebilir; ancak analiz ve raporlama sistemlerini kolaylaştırmak için veritabanında aşağıdakilerden biri veya daha fazlası saklanmalıdır.

- **Başlık Bilgisi:** Genellikle bir olayın zaman damgasını ve olayda yer alan IP adreslerini içerir. Fazla veya eksik raporlama yapan sunucuları belirlemek veya sistemler arasındaki olay eğilimlerinin bağlantısını tespit etmek için bu bilgilerin tek başına saklanması faydalıdır.
- **Gövde:** Genellikle olayın mesajıdır. Bu bilgilerin bir veritabanı sisteminde saklanması, gerçek zamanlı bir uyarı sisteminin oluşturulmasında yararlıdır. Örneğin; aynı başarısız oturum açma mesajının yüksek frekansta görülmesi hızlı bir şekilde sorgulanabilir ve raporlanabilir.

- **Analiz ve özet sonuçlar:** Eğilimleri belirlemek ve sonuçlarını özetlemek için her bir sistemde özel komut dosyaları ve araçlar kullanılabilir. Bu analizin merkezi bir havuzda depolanması, kurum genelinde olay analizinin raporlanmasını kolaylaştıracak ve veritabanı depolama ve ölçeklenebilirlik gereksinimlerinin düşük olduğu bir kurum için merkezi denetimin geliştirilmesini ve özet raporları kolaylaştıracaktır. (Chuvakin vd. 2013)

3.3.1.2 Hızlı Erişim

Nelerin saklanacağını belirlendikten sonra, ilgili verilere hızlı bir şekilde erişilebilmesi amacıyla veritabanını optimize etmek için bazı analiz ve incelemelerin yapılması gereklidir. Günlük incelemeler, raporlama ve uyarı mekanizmaları için ortak sorgularda kullanılacak veritabanı sütunlarını belirlemek kritik bir noktadır. Syslog verilerini saklarken önerilen bir yaklaşım, aşağıda belirtilen alanlar gibi yaygın olarak kullanılan syslog alanlarına veritabanı indeksleri oluşturmaktır.

- **Öncelik:** Mesajın önem derecesi
- **Tarih ve saat:** Olayın ne zaman oluştuğu
- **Üretildiği sunucu:** Olayı üreten sistem
- **Mesaj:** Oluşan olayla ilgili ayrıntılar

Log verileriyle birlikte saklama alanı da büyür. Sorgular için indekslenmiş ve optimize edilmiş bir veritabanında bile, trilyondan daha fazla satırı aramak yavaş ve hantal olabilir. Birçok veritabanı sistemi bölümlemeyi desteklemektedir. Bölümleme, mantıksal olarak tek bir veritabanı tablosunun daha küçük parçalara ayrılmasını sağlar. Log verileriyle birlikte, bir veritabanı tablosunun tarih ve saat bazında bölümlendirilmesi mantıklı bir yaklaşımdır ve aşağıdaki avantajları sağlayacaktır.

- Daha küçük bir fiziksel dosya olarak eklemek veri ekleme hızını iyileştirilir ve her ekleme sırasında daha küçük indeks dosyaları güncellenir.
- Veritabanı sistemi tarafından incelenen ve filtrelenen küçük veri parçaları, bazı sorgular için sorgu performansını iyileştirilebilir.
- Toplu silme işlemi, bir log veri setinin tutma süresi dolduğunda ilgili veri bölümü kaldırılarak gerçekleştirilebilir. Bu işlem, veritabanında gerçekleştirilen tek tek silme işlemlerine göre log verilerinin imha edilmesini önemli ölçüde iyileştirir.

- Bazı sistemler, nadiren kullanılan bölümlerin daha yavaş ve ucuz depolama seçeneklerine geçirilmesine izin verir.
- Ayrık bölümler veritabanı bakımı için çevrimdışına alınabilir. Bu ise, yeni log verilerinin eklenmesini etkilemeden en iyi performans için veritabanının bakımını sağlamaya izin verir. (Chuvakin vd. 2013)

3.3.1.3 Raporlama

Kurumların, iç paydaşlar ve dış denetçilerle birlikte log verilerinin denetimini ve incelenmesini sağlamak amacıyla log verilerinin raporlarını oluşturmaları gerekir. Raporların hızlı ve isteğe bağlı olarak oluşturulabilmesi için genellikle ilave rapor tabloları gerekir. Bu tablolar kurumla ilgili önemli öğelerin özet sayılarını içermelidir. Arayüz araçlarının veriye erişimini hızlandırmak için bu tablolar, arka plan işlemleri vasıtasıyla hesaplanabilir. Bu bilgileri oluşturmak çok zaman alabilir ve raporlama yapısında gerçekleştirilecek değişiklikler maliyetli olabilir. Özet raporlama verileri üretilmesi düşünülen alanlar aşağıda yer almaktadır.

- **Analiz sonuçları:** İncelenmesi veya üzerinde hareket edilmesi gereken ilginç bir şeyi gösteren log girdilerinin birleşimidir.
- **Önem durumuna göre sunucu başına olay sayısı:** Saldırı desenleri bulma veya sorun alanlarını tespit etme konusunda yararlı olabilir.
- **Zaman esaslı özet sayımlar:** Birçok kurumun diğer kuruluşlarla paylaşmak için günlük, haftalık veya aylık rapora sahip olması gereklidir ve bu zaman dilimlerine dayalı özetleri toplaması, bu raporlamayı otomatikleştirir ve hızlandırır.
- **Ağa veya cihaz türüne göre raporlama:** Ağın belirli bölümleri farklı raporlama ihtiyaçları gerektirir veya farklı uyumluluk çerçevelerine dâhil olur. PCI DSS, ağın ödeme kartı işleme bölümünün ilave raporlama ve denetim gereksinimlerinin bulunduğu yaygın kullanım durumudur.

3.4 Hadoop Log Saklama

Log verisi ve etkinliklerde yaşanan artışların aşırı depolama ve sistem kapasitesi gerektirmesi nedeniyle geleneksel veritabanıyla ilgili birçok zorluk sistemin ölçeklenebilirliğiyle ilgilidir. Geleneksel veritabanı sistemleri, çok sayıda eşzamanlı kullanıcıyı ve isteği desteklemek ve veri saklama ihtiyaçlarını karşılamak için SAN

depolamasına sahip hızlı üst düzey bir donanım kullanılarak oluşturulur. Bunun aksine Hadoop sistemleri genel olarak, Linux çalıştıran ve Intel donanımı kullanan ticari bir PC ve RAID içermeyen bir makine kümesinin her düğümünde birkaç terabaytlık yerel depolama alanı kullanılarak oluşturulur. Bir Hadoop kümesi birkaç bağımlı düğümden ve en az bir ana düğümden oluşur. Kümeye başka bir düğüm ekleyerek ilave alan ve kapasite eklenebilir. Günümüzde Yahoo ve Facebook gibi şirketler, uygulamalarını petabyte boyutunda bilgi ile destekleyerek son kullanıcı durum güncellemelerini ve arama sorgularını hızlı bir şekilde işlemek amacıyla Hadoop'u kullanmaktadır.

Log verilerini Hadoop'ta saklamanın avantajları şunlardır:

1. Her sistemde platforma özel sorgu araçlarını kullanmak yerine log verilerinin çabucak alınmasını ve aranmasını sağlar.
2. Sonuçların hızlı bir şekilde bulunması, işlenmesi ve alınması için arama isteklerini küme düğümlerine dağıtırken veri boyutu arttıkça ölçeklenir.
3. Java ile oluşturulmuştur ve log verilerinin gerçek zamanlı olarak görüntülenmesi ve analizi için araçlar geliştirilebilir.
4. Verileri Hadoop kümesindeki düğümler boyunca yapılandırılmış bir düz dosya kümesi olarak Hadoop Dağıtık Dosya Sistemi'nde (HDFS) saklar. Bu, Hadoop'un geleneksel veritabanı sistemlerine göre daha hızlı veri ekleme değerlerine ulaşmasına olanak tanır.
5. Hata toleranslıdır ve küme düğümlerinde çok sayıda veri kopyası oluşturur. Bu nedenle tek bir düğüm başarısız olursa veri, kümedeki diğer düğümlerden alınmaya devam edebilir.

Log verilerini Hadoop'ta saklamanın dezavantajları ise şunlardır:

1. Hâlihazırdaki loglama araçlarının birçoğu tarafından Hadoop için şu anda sınırlı destek bulunmaktadır. Rsyslog, Hadoop kümesine syslog mesajları yazabilme özelliğini eklemiştir, ancak diğer log kaynaklarının log verilerini Hadoop'a yazabilmesi için bir araç geliştirilmesi gerekir.
2. Verileri doğrudan sorgulamak ve raporlamak için kullanılan araçlar oldukça sınırlıdır. Hâlihazırda az seçenek mevcut olduğundan kurum, bu ihtiyacı karşılamak için gerçek zamanlı analiz ve inceleme için özel arayüz sistemleri geliştirmelidir.

3.5 Log Verisine Erişme ve Arşivleme

Logların fiziksel olarak depolanması için birden fazla seçenek bulunmaktadır. Seçenekler gözden geçirilirken her aracın alma ve erişim hızına dikkat edilmesi gerekir. Genel olarak, yüksek erişim hızı seçenekleri çok maliyetlidir. Burada log yönetiminde logların çevrimiçi, yakın zamanlı ve çevrimdışı depolanması ve bunların göreceli maliyetleri ile erişim hızlarından bahsedilecektir.

3.5.1 Çevrimiçi

Çevrimiçi log bilgileri, doğrudan erişilebilen ve alınabilen bilgilerdir. Doğrudan erişim için tahsis edilen donanımın açık ve hazır olması nedeniyle bu en pahalı seçenektir. Çevrimiçi depolamada bir sunucuya, bir veritabanı sistemine veya depolama alan ağı (SAN) sistemlerine fiziksel olarak bağlı diskler bulunmaktadır.

3.5.2 Yakın zamanlı

Yakın zamanlı depolama, çevrimiçi ve çevrimdışı depolama arasında yer alır. Yakın zamanlı sistemler genellikle insan müdahalesine ihtiyaç duymaz ve veriler optik depolama kutusu veya robotik kaset sisteminin bir parçası olarak ele alınır. Bu tür depolama alanlarının erişim süreleri genellikle yüksektir ve sisteme bağlı olarak birkaç saniye ile birkaç dakika arasında değişebilir. Yakın zamanlı depolama sistemleri için maliyetler geniş çapta değişir; ancak genel olarak çevrimiçi depolama seçeneklerinin maliyetinin yarısı kadardır. Yakın zamanlı depolama, kapasiteyi artırmak için ek kasetler veya optik diskler ekleyerek de ölçeklenebilir.

3.5.3 Çevrimdışı

Çevrimdışı depolama en yavaş ve en ucuz seçenektir. Çevrimdışı sistemler genellikle bir optik disk veya kaset almak için insan müdahalesi gerektirir ve verileri erişim için çevrimiçi veya yakın zamanlı depolama sistemlerine geri yükler. Çevrimdışı depolama, ek optik disklerin veya kasetlerin satın alınmasıyla ölçeklendirilebilir ve genellikle yakın zamanlı depolamadan daha ucuzdur. Yakın zamanlı ve çevrimdışı depolamadaki sorun, depolama ortamının raf ömrünün olmasıdır. Kaset veya CD/DVD'nin genel kabul gören raf ömrü kabaca 2 ila 5 yıl arasındadır. Medyanın ömründen daha uzun bir tutma politikası varsa ortam ömrünün sonuna yaklaştıkça verileri farklı bir medyaya yeniden kaydetmek gerekecektir.

4. LOG ANALİZİ

Buraya kadar, log verisinin ne olduğu, log verisinin biçimleri, merkezi loglama, log tutma ve birkaç farklı konularda temel bilgiler verildi. Bu bölümde ise log analizinden bahsedilecektir.

4.1 Analize Giriş

Teknikler ve yaklaşımlar hakkında konuşmaya başlayabilmek için, log verilerini analiz etmeye nereden başlamak gerektiği ve nelerin gerekli olduğu noktasında bazı temelleri koymak gerekir. Burada üç önemli konu ele alınacaktır: hedefler, planlama ve hazırlık. Log analiz hedefleri, log analizi aracılığıyla gerçekleştirmek istenilen hedeflerdir. Planlama, log analizi sürecinde bir sonraki adımdır ve log analizine yaklaşım planlanır. Son adım olan hazırlık ise log analizi etkinliğini ele almak için log verilerini ve ortamı hazırlamakla ilgilidir.

4.1.1 Hedefler

Analizin hedefleri, özel ihtiyaçlara göre değişebilir. Bir bankacılık kuruluşuyla bir devlet kurumu farklı hedeflere sahip olacaktır. Bununla birlikte, çoğunlukla herkes için geçerli olan üst düzey hedefler vardır. Birincil hedef, hâlihazırda olmuş olan şeylerin farkına varmak ve onlar hakkında uyarı oluşturabilmektir. İkincil hedef, bilinen kötü şeyleri tanımaktan başka, bilinmeyen, daha önce görülmeyen ya da bilinen şeylerin dışındakileri algılamaktır. Saldırganlar tekniklerini geliştirdikçe, normal araçlar ile norm dışı davranışları tespit etmek zorlaşmaktadır. Örneğin; log verilerine bilinen şeyler için bakıldığında ortamda dolaşan bir saldırgan fark edilmeyebilir. Bu tür bir analiz, logların manuel olarak incelenmesi ile kolay yapılamaz, daha karmaşık teknikler gerektirir.

4.1.2 Planlama

Log analiz sistemleri için gereksinimler, hedefler kadar kritiktir. Bir analiz sistemi planlanmadan önce temel kavramlar iyi anlaşılmalıdır.

4.1.2.1 Doğruluk

Doğruluk; alınan, işlenen, arşivlenen vb. log verilerinin herhangi bir şekilde kusurlardan veya yanıltıcı bilgi içermeyeceğinden emin olmaktır. Başka bir deyişle, alınan veriler log sisteminin amaçladıkları olmalıdır. Buradaki bir sorun yanlış

pozitiflerdir. Saldırı Tespit Sistemleri (IDS) yanlış pozitiflerin bir kaynağıdır. Örneğin; bir IDS kötü amaçlı olmayan ya da hedef sistemdeki gerçek bir zafiyeti temsil etmeyen trafiği kötü amaçlı olarak rapor edebilir. Bunun bir çözümü, olası kötü niyetli davranışa çapraz referans yapmak için zafiyeti veritabanına danışmaktır. IDS, bireylerin kabul edilebilir bir ağ kullanımını oluşturmak için kullanıcı ve grup profillerinin kullanıldığı bir politika şeması da uygulayabilir. Yanlış pozitifleri azaltmak, dağınıklığın minimuma indirilmesine yardımcı olur.

Doğruluk konusunda önemli bir konu, zaman damgası kavramıdır. Bir zaman damgası, olayların gerçekleştiği tarih ve saattir. Hemen hemen her güvenlik uygulaması zaman damgalarını kullanır. Bu uygulamaların çoğunda, log mesajlarıyla birlikte bir zaman damgası gönderilir. Sorun, farklı üreticilerin farklı biçimleri kullanmasıdır. Bazıları UNIX zamanını kullanırken, bazıları tarih ve saat için standart biçimlendirilmiş bir dize kullanmakta ya da daha kötüsü kendi tarihlerini oluşturmaktadır. Zaman damgalarına ilişkin ana standart ISO 8601 standardıdır. ISO 8601 zaman damgasının bir örneği şu şekildedir:

2017-02-19T13:15:26+00:00

Biçimi YYYY-MM-DDTHH:MM:SS.SSS+/− H şeklindedir. Tarih ve saati ayırabilmek için zaman damgasında bir "T" harfi vardır. +/-, GMT için bir uzaklıktır. Ayrıca RFC 3339, internette kullanılmak üzere tarih ve saat biçimlerini tanımlar. (Date and Time on the Internet: Timestamps, 2002)

Çeşitli sistemlerden gelen logların zaman damgalarına ilişkin örnekler aşağıda verilmektedir. Görüldüğü gibi logların zaman damgası biçimleri gerçek bir sorundur.

- 15/May/2014:02:25:10 -0300
- 12-03-201103:21:25
- 12/24/2012 18:45:06
- 2010-03-09 06:12:41
- Mon Apr 11 16:12:19 2016
- 11/6/2013,7:14:39 PM
- Sep 19 11:14:35

4.1.2.2 Bütünlük

Bütünlük, log mesajlarının kaynağına ve içeriğine güvenmeyle alakalıdır. Ticari log mesajlarından gelen log verilerine güveniliyorsa, üreticinin inisiyatifine kalınmış demektir. Hâlihazırda bazı üreticiler, istemcileri ve sunucuları doğrulamakla kalmayıp verilerin şifrelenmesinde de SSL (Secure Sockets Layer) kullanmaktadır. Diğer üreticiler ise eski syslog'a ya da SNMP'ye güvenmektedirler. SNMP'nin en yeni sürümü (sürüm 3) güvenlik ve şifreleme için tolerans sağlar; ancak pek çok üretici henüz bunu desteklememektedir.

Bir üreticinin log verilerine güvenmek gerekiyorsa, STunnel gibi bir şey kullanılabilir. STunnel, bir SSL tüneli üzerinden her türlü veriyi iletmeyi ve almayı sağlayan genel bir uygulamadır. Güvenli iletişime ihtiyaç duyulduğunda, verilerin kendisi üzerinde kontrol sahibi olunmadığında veya belirli bir SSL API'si bulma imkânı olmadığında kullanışlıdır.

Başka bir çözüm, verileri açık bir şekilde göndermektir. Bunun için tehlikelerden arındırılmış olduğu bilinen özel ağ bağlantıları kullanılmalıdır. Kurumda hâlihazırda birçok amaç için kullanılan ağ bağlantıları olabileceğinden bu maliyetli bir öneridir ve açıkçası hassas veri iletimi için bu bağlantılar kullanılamaz.

Bütünlükle ilgili son bilgi, dijital imza ile ilgilidir. Log verileri dijital olarak imzalanırsa, güvenilirliğinden emin olunabilir. Bazı düzenleyici kuruluşlar, log verilerinin bir suç soruşturmasında kanıt olarak kullanıldığı sistemlerde bu seviyede bir bütünlük isteyebilir.

4.1.2.3 Güven

Güven, ilgilenmeye değer bir olay olduğuna ne kadar emin olunmasıyla ilgilidir. Bazı ürünler belirli bir olaya öncelik veya önem derecesi koyarak bunu dener. Bu kavramsal görüş, yüksek, orta veya düşük şeklinde olabilir veya 0'dan 10'a kadar giden bir ölçek olabilir. Örneğin; birçok Cisco cihazı 0 ile 7 arasında bir ölçek kullanır.

Gerçekte güven, basit bir öncelik planından biraz daha derine inerek mevcut ağ ve altyapı hakkında her şeyi bilmekten geçer. SIEM sistemleri bu konuda faydalıdır. SIEM'ler genellikle bir ağdaki tüm güvenlik cihazlarının üzerine konumlanan kurumsal uygulamalardır. Güvenlik duvarları, IDS'ler, zaafiyet tarayıcıları,

veritabanları ve uygulamalar da dâhil bütün sistemler SIEM'i besler. SIEM, log verilerini analiz etmek için istatistiksel algoritmalar kullanabilir veya bir operatörün log verilerinde belirli desenler arayacak özel kurallar oluşturmasına izin verebilir. Özel kurallar oluşturmanın yararı, heterojen log verilerine dayalı olarak yakalanması muhtemel bir güvenlik tehdidinin soyutlanmasına olanak tanımasıdır. Güvenin merkezi, birçok alandan girdi almak ve tüm girdilerin kümesinden daha doğru bir gerçek çıkarmaktır.

4.1.2.4 Koruma

Koruma, log verisinin herhangi bir şekilde değiştirilmemesi düşüncesidir. Kötü amaçlı eylemleri araştırmak için bir log verisi kullanılması planlanıyorsa bu şarttır. Log sistemindeki işleme, analiz ve arşivleme sırasında bu bilgilerin hiçbir şekilde değiştirilmemesi kritik öneme sahiptir.

4.1.2.5 Sanitizasyon

Log sistemleri ile uğraşırken log verilerinin gizliliği bir endişe kaynağı olabilir. Örneğin; log verileri, bir ağ bağlantısı üzerinden iletilmesi istenilmeyen kullanıcı hesap bilgilerini veya IP adreslerini içerebilir. Ayrıca loglar üçüncü taraf bir alıcıya iletilirken bazı şeylerin gizli tutulması istenebilir. Log sanitizasyonu için iki temel teknik vardır. Birincisi, sanitizasyona tabi tutulması istenen girdileri belirleyerek değiştirmek ya da kaldırmaktır. Örneğin; 10.65.125.13 IP adresi xxx.xxx.xxx.xxx olabilir. Bu, sanitize edilmiş log girdisini görüntüleyen bir kişiye, log dosyasında bir IP adresi olduğunu bildirir. Diğer teknik kaldırılan log verilerinin belirli bir noktada yeniden oluşturulabileceği bir şemaya dayanmaktadır. Örneğin; sanitize edilmiş bilgiler çıkarılarak güvenli bir dosyaya yerleştirilebilir.

4.1.2.6 Normalleştirme

Normalleştirme, log ve analiz sistemlerine ait olan diğer bir konudur. Üretici verilerinin güvenlik ve ağ analistleri tarafından işlenebilecek, rapor edilebilecek şekilde biçimlendirilmesi gerekir. Pek çok güvenlik sistemi üreticisi, syslog veya SNMP kullanmak yerine kendi loglama biçimini seçer. Verileri, IP adresleri ve port bilgileri gibi parçalarına ayırmak için log mesajları işlenmek istendiğinde bu gerçek bir sorundur. Normalleştirme, birleştirme olarak da adlandırılır, aslında buradaki amaç birçok farklı üreticinin log veri biçimini iyi bilinen birisine eşlemektir.

Verileri saklamak için mantıksal bir düzen tanımlamak, normalleştirmenin bir parçasıdır. Uygulamanın veya kurumun gereksinimlerini karşılayan bir veri şeması oluşturularak üreticilerin log verileri ne kadar ayrıntılı veya kullanışsız olursa olsun istenilen biçime kolayca eşlenebilir. Verilerin ön biçimi bilindiğinde veri madenciliği ve tarihsel arama gibi şeylerin yapılması çok daha kolay hale gelir.

4.1.2.7 Zamanla İlgili Zorluklar

Planlama bölümündeki son konu zaman senkronizasyonudur. Zaman, olaydan sonra gerçekleştirilecek bir soruşturmanın çok kritik bir bileşenidir. Loglar, bir şeylerin olup olmadığını tam olarak gösterebilir. Zaman ve log verileriyle ilgili bazı zorluklar şunlardır:

- **Yanlış zaman damgası:** Ölü bir pil ya da başka bir donanım arızası, gerçekle hiçbir şekilde ilgisi olmayan zaman damgaları görmenin yaygın bir nedenidir.
- **Herhangi bir yerde zamanın hep aynı olması:** Birçok log dosyasında, logların kendileri bir zaman dilimi içermez. Bu noktada loglama sisteminin bulunduğu zaman dilimine dikkat edilmesi gerekir.
- **Zaman kayması:** Ağ Zaman Protokolü'nün (NTP) saat kayması problemi saniye cinsinden zaman sapmalarına neden olur. Bu da olayların sırasını değiştirebilir. Tüm sistemlerde UTC zamanını kullanmak bu sorunu çözebilir.
- **Syslog gizemi:** Arabellek syslog sunucularının (syslog-ng veya logları saklayıp iletebilen ticari sistemler gibi) benimsenilmesi nedeniyle gelecekte sık karşılaşılacak bir durumdur. Böyle bir durumda, bir syslog mesajının gecikmiş log iletiminin zamanını değil de olayın zamanını içerdiğinden emin olunması gerekir.
- **İki zaman damgası:** Sistemlerin bazılarında birden fazla zaman damgası olabilir ve bunlardan birisi olayın hangi saatte olduğu hakkında yüksek bir kesinlik derecesine sahip olabilir.
- **Loglama gecikmesi:** Loglama işlemleri veya programların yürütülmesi konusunda bir gerçektir. Bazı sistemler, program başlatıldığında veya sonlandırıldığında değil de işlem yürütüldüğünde loglama yapar.

- **Zaman dilimi problemi:** Saatin 4:35 olması, 4:35 AM veya 4:35 PM (16:35) anlamına gelebilir. Bazı eski sistemlerde bu hata yapılmaktadır.

Bahsedilen zorlukların azaltılması için uyulması gereken bazı kurallar vardır:

- NTP açısından, garantili bir zaman sunucusu ile senkronize olan bir zaman sunucusu çalıştırılmalıdır (*time.nist.gov* gibi).
- NTP sunucusuyla tüm loglama sistemleri arasında zaman senkronizasyonu uygulanmalıdır.
- Loglar alınırken ve merkezileştirilirken saat dilimi kontrol edilmelidir (GMT, PDT veya EST +/-N biçimi kullanıldığından emin olunmalıdır).
- Log gecikmesini hesaba katmak için loglanan sistemler bilinmelidir.
- Daha yüksek düzeyde güvence için güvenilir zaman protokolleri incelenmelidir (ANSI X9.95: 2005 gibi). (Chuvakin vd. 2013)

4.1.3 Hazırlık

Veriler sadece analiz için değil, saklamak ve üzerinden rapor almak amacıyla da kullanılacağından ortamın ve log verilerinin hazırlandığından emin olmak gerekir.

4.1.3.1 Log Mesajlarını Ayırma

Log mesajları, bilgisayar veya hizmetlere göre ayrı ayrı tutulabiliyorsa bu analizin daha kolay olmasını sağlar. Bununla birlikte aşağıdaki görevler de daha kolay yapılabilir.

- Benzersiz desenler dizisi geliştirmek: Mesajlar hizmete göre ayrılırsa daha kolay yapılabilir.
- Her desendeki değişkenleri tanımlamak: Tür ve aralık.
- Yanlış pozitifleri ve bozulmuş verileri kontrol etmek.
- Desenleri saklamak: Bir düz metin dosyası, veritabanı vb. olabilir.

4.1.3.2 Ayrıştırma

Ayrıştırma, ham bir log mesajını alma ve buradan özellik bilgisi ayıklama ile ilgilidir. Bu özellikler, log mesajının kaynağı, logun ne zaman oluştuğunu gösteren zaman damgası, kaynak ve hedef IP bilgileri, kullanıcı bilgileri vb. içerir. Bu, log verilerinin daha iyi anlaşılmasını sağlar. Çünkü detaylı analiz, raporlama ve benzeri işlemler için log mesajının parçalı bölümleriyle çalışılması gerekebilir.

4.1.3.3 Veri İndirgemesi

Veri indirgeme teknikleri, log verilerini kaynağından ara işlem sunucusuna etkin bir şekilde iletmekle birlikte verileri verimli bir şekilde saklamak için de önemlidir; ancak bu tekniklerden bazıları kaynak bilgisayar sisteminin CPU performansını, alıcı sistemi veya her ikisini birden etkilemesi gibi bir maliyetle birlikte gelir. Bu tekniklerden elde edilen kazanımlar, sisteme olan etkisinden daha önemli olabilmektedir. Ağ üzerinden ne kadar veri gönderilebileceğine dair sınırlamalar varsa, bu teknikler fayda sağlar.

4.1.3.3.1 Veri Sıkıştırma

Veri sıkıştırma, bir bayt dizesini alıp daha küçük bir bayt kümesine sıkıştırmakla ilgilidir. Sıkıştırılmış dize daha sonra alıcı sistem veya uygulama tarafından açılır. Olumsuz yönü, verilerin sıkıştırılması ve diğer tarafta açılması gerektiği zaman hem gönderen hem de alıcı üzerinde performans etkisi oluşturmasıdır.

Log analizine sıkıştırmanın nasıl uygulanabileceği konusunda iki yol vardır. Birincisi, normalleştirilmiş olayları analiz etmek veya arşivlemek için log toplayıcıdan işleme sunucusuna gönderildiği zaman sıkıştırmaktır. Diğer yol ise veritabanında sıkıştırmaktır.

4.1.3.3.2 Veri Tekilleştirme

Log verisi toplanırken yinelenen olay verileri sıklıkla alınır. Teknik açıdan bakıldığında, yinelenen olay verisi kavramı hiçbir zaman güvenlik log verileri için geçerli değildir. Bazı özellikleri farklı olan olayların türleri aynı olabilir. Örneğin; bir saldırganın ağdaki bilgisayarlara port taraması yaptığı varsayılırsa, bu tarama girişimlerinin her biri için IDS muhtemelen ayrı olaylar oluşturacaktır. Olay türü aynı olabilir; ancak saldırganın kaynak IP adresi ve portu özellikle sahtecilik (spoofing) kullanılırsa her olay için farklı olabilir.

Güvenlik olayının logları iki ana kategoriye ayrılır: bilgilendirici olanlar ve operasyonel olanlar. Bilgilendirici log verileri, sistem başlatma veya kapanma gibi şeylerdir. Operasyonel log verileri, IDS'ten, güvenlik duvarından ve benzer sistemlerden bildirilen etkinliklerdir. Örneğin; başarısız SSH giriş denemeleri, operasyonel ilgi alanı içerisindedir.

Hedef, bir olay için benzersiz bir tanımlayıcı oluşturmak ve görünüşte farklı olayların mantıksal olarak tek bir olayda bir araya getirilmesine olanak tanımak ve söz konusu olayın kaç kez farklı olduğunu bir sayıyla ortaya koymaktır. Bu sayı, tüm olayları temsil eden tek bir olayla yüzlerce ve binlerce olayın gerçekleşmesine izin verir. Bu çok büyük miktarda yerden kazandırır.

Log verilerinde, tekilleştirme konusunda yardımcı olabilecek birkaç bilgi parçası şunlardır:

- Log Olayının Kaynağı
- Kaynak IP Adresi
- Kaynak Port
- Hedef IP Adresi
- Hedef Port

Log Olayının Kaynağı: Log olayının kaynağı, etkinliği oluşturan cihaz, yazılım vb. kaynaklardır. Pek çok güvenlik ürünü yönetim sunucusu mimarisine doğru ilerlemektedir. Bunun anlamı, tek tek bir log sunucusuna rapor veren her cihaz yerine, her cihaz bir veya daha fazla cihaz adına olayları log sunucusuna ileten bir yönetim sistemine rapor vermesidir. Bu, üreticilerin üçüncü taraf alıcılara log olaylarını iletmesi için daha verimli ve ölçeklenebilir bir yoldur. Olumsuz yönü, SNMP veya syslog kullanılıyorsa, log olayının fiziksel kaynağı yönetim sunucusudur, yani syslog mesajında yönetim sunucusunun IP adresi görünecektir. Olayı gerçekte üreten kaynağı tanımlamak için yönetim sunucusu, olayın kaynağını log sunucusuna gönderdiği log mesajının içine yerleştirecektir. Olay log biçimlerinin standardizasyonu olmadığı için kaynak alanının biçimi üreticiye özgüdür. Bu kaynak alan, sensörün IP adresi, sunucu adı veya benzersiz bir tanımlayıcı olabilir. Tekilleştirme için log olayının kaynağı kullanılacağı zaman bu durumun farkında olunması gerekir.

Kaynak IP Adresi: Kaynak IP adresi, saldırının ortaya çıktığı yerden gelmektedir. Pratikte bu, tekilleştirme garantisi için yeterli değildir. Bunun nedeni, esas olarak saldırganların bir veya daha fazla kaynak IP adresi kullanarak kimliğini gizleyebilmesidir.

Kaynak Port: Kaynak port, kaynak sistemde rastgele oluşturulmuş porttur. Yine sahtekârlıktan dolayı bu, tekilleştirme için yeterli değildir.

Hedef IP Adresi: Hedef IP adresi hedef sistemin IP'sidir. Bu, genellikle bir saldırı sırasında statik kalacak olan bir alandır ve tekilleştirme çalışmalarında yardımcı olacak iyi bir adaydır.

Hedef Port: Bir port tarama saldırısı gerçekleşmesi durumunda, IDS aynı hedef IP adresi için birçok farklı portu rapor eder. Birçok saldırı için, bir hedef IP adresi ve bir hedef port olacaktır. Hedef port, tekilleştirme işlemine dâhil edilmek için az çok iyi bir adaydır. (Chuvakin vd. 2013)

4.1.3.3.3 Sonradan Veri İletimi

Temel ilke, analiz edilmek istenmeyen ancak verileri arşivlenecek olayları iletmek için günün belirli bir saatini belirlemektir. Örneğin; çok sıkı güvenlik duvarı kuralları uygulanıyorsa, güvenlik duvarı kabulleri, izinleri vb. görmek veya analiz etmek önemsenmeyebilir, ancak yine de tüm bu olayları arşivlemek gerekir. Mesai saatleri dışında ağ trafiğinin minimumda olduğu zaman, bu güvenlik duvarı olayları gönderilebilir. Ayrıca, bu olayları sunucuda analiz etmemek, yükü azaltarak CPU döngülerini ele alınacak daha ağır bir nitelikteki olaylar için ayırabilir.

Bu tekniği gerçekleştirmek için gereken adımlar oldukça basittir. Birincisi, daha sonra iletmek istenilen log mesajlarını algılamak ve ayırmak için bir yol bulmak gerekir. Bu, syslog-ng'deki yerleşik özellikler veya sistemin desteklediği herhangi bir şey ile yapılabilir. Bu mesajların log toplama noktasında tutulması gerekir; ancak daha yüksek önceliğe sahip olayların geliş gidişini etkilemeyecekleri bir yerde saklanır. İkincisi, olayların ne zaman iletileceğini tanımlamak için bir tarih ve saatin bildirilmesi gerekecektir. Son olarak zamanı geldiğinde, analiz veya arşivleme için olaylar ana sunucuya iletilmelidir. Sunucuya, bu olayların analiz edilemeyeceğini bildirmenin bir yoluna da ihtiyaç olacaktır; bunlar daha sonraki tarih ve saatte iletiliyorsa, bu olayları işlemek mantıklı değildir. Özel log analiz aracı bu kavramı desteklemiyorsa, scp, rsync veya başka bir araç kullanılarak oluşturmak zorunda kalınabilir.

Dikkat edilmesi gereken bir nokta, zaman aralıklarının belirlenmesine izin veriliyorsa, ayrılan tüm olay verilerini iletmek için belirlenen saat aralığının yeterli

olup olmadığıdır. Burada bir karar verilmesi gerekir: olayları göndermeyi bırakmak ya da tüm olaylar gönderilene kadar zamandan bağımsız olarak hareket etmek. Bu tamamen özel gereksinimlere bağlıdır.

4.2 Basit Analiz Teknikleri

Logları manuel olarak incelemek, zor bir süreçtir. Bununla birlikte, otomatikleştirilmiş araçlar yoksa logları manuel olarak incelemek gerekir. Log analizi yapmanın nedenlerinden bazıları şunlardır:

- Uyumluluk ve Düzenleyici Kuruluşlar
- Ağda Gerçekleşen Durumlardan Haberdar Olmak
- Altyapı Yatırım Getirisi
- Güvenliği Ölçmek
- Olaya Müdahale

4.2.1 Basit Log Görüntüleyiciler

Burada, Unix ve Windows'daki bazı basit log görüntüleyicilerden ve bunların nasıl yardımcı olduklarından bahsedilecektir.

4.2.1.1 Gerçek Zamanlı İnceleme

Çoğu Unix ve Linux dağıtımında, log analizine yardımcı olabilecek çeşitli araçlar bulunur. Bazıları, gerçek zamanlı log incelemelerine yardımcı olurken bazıları geçmiş logları incelemeye yardımcı olur.

Logları incelemeye en basit komuttan başlanılacak olursa:

```
# tail –f /var/log/auth.log
```

Bu komut, Linux'ta yer alan *auth.log* dosyasındaki son satırları ve komut başlatıldıktan sonra ortaya çıkan yeni satırları gösterecektir. Çoğu durumda fazla kullanışlı olmasa da en basit gerçek zamanlı log görüntüleyicidir. Bununla birlikte, yeni bir program testi veya bir arka plan programı çökmesi sırasında ortaya çıkan sistem log kayıtlarına göz atmak için oldukça kullanışlıdır.

Yukarıdaki komuta, metin eşlemeye yardımcı olan *grep* komutu eklenerek istenen kayıtların filtrelenmesi sağlanabilir:

```
# tail –f /var/log/auth.log | grep sshd
```

Bu komut yalnızca *sshd* tarafından üretilen mesajları gösterecektir.

Feb 23 14:44:15 xyz sshd[258]: Accepted publickey for root from 60.55.10.8 port 125 ssh2
Feb 23 14:45:12 xyz sshd[258]: refused connect from 60.45.128.36
Feb 23 14:46:43 xyz sshd[258]: Failed password for root from 60.15.32.58 port 2345 ssh2

Yukarıdaki gerçek zamanlı log görüntülemeyi daha da geliştirmek için, verilere bakmanın yanı sıra bir dosyaya göndermeye yardımcı olacak *tee* komutu eklenebilir:

 # tail –f /var/log/auth.log | grep sshd | tee son-sshd-loglari.txt

Bu komut, log satırlarının *"son-sshd-loglari.txt"* dosyasına da kaydedileceğini gösterir.

less komutu, bekleme modunda aynı işi gerçekleştirmek için kullanılabilir:

 # less /var/log/auth.log

Windows ortamında, yeni gelen log kayıtlarını izlemek için Windows'un içerisinde gelen Olay Görüntüleyicisi çalıştırılabilir.

4.2.1.2 Tarihsel Log İnceleme

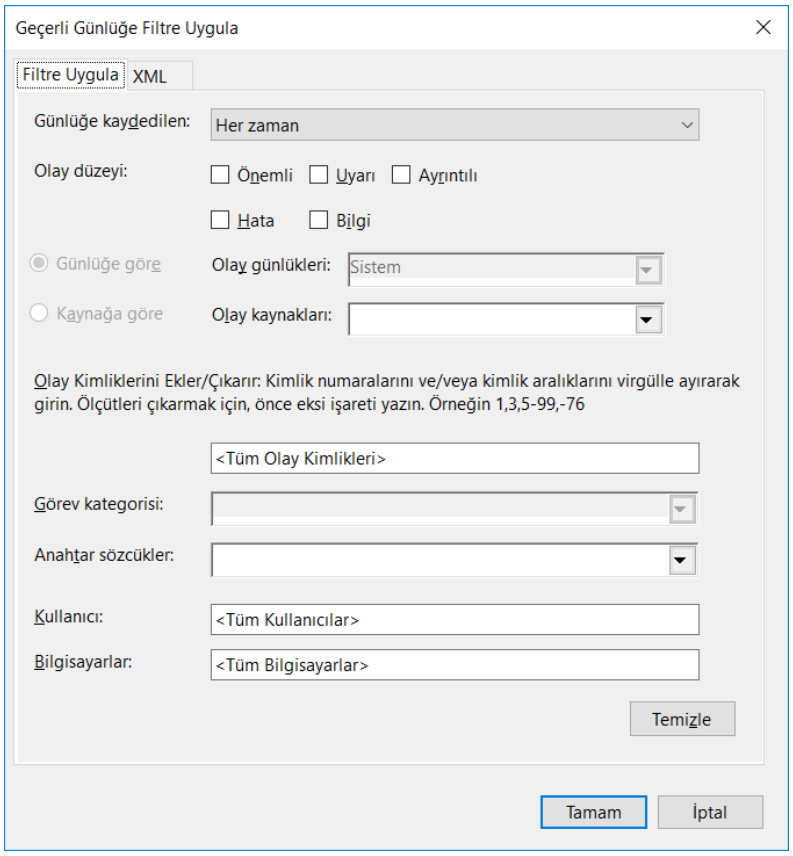
Seçilen platformdaki herhangi bir metin editörü, düz metin loglarına bakmak için kullanılabilir. Burada dikkat edilmesi gereken nokta, bazı log dosyalarının çok büyük olmasıdır. Bu nedenle, her metin editörü onları verimli bir şekilde işleyemez. Windows'ta yer alan not defterine 2 GB'lık bir dosya yüklemek sorun çıkartabilir. Diğer editörlerle çalışılırsa, işlem son derece yavaş olacaktır.

Unix'te *less* ve *more* kullanmak, genellikle daha uygundur. Büyük dosyaları işleyebilir ve arama yapabilirler. Unix/Linux sisteminde *man* komutuyla, ilgili komut hakkında bilgi alınabilir. Örneğin; *man less* veya *man more* gibi. Aşağıdaki gibi bir *info* komutu da daha fazla ayrıntı sağlamaktadır.

$ info less

Windows'ta yalnızca paket olarak gelen Olay Görüntüleyicisi aracı bulunmaktadır. Burada, standart üç kategorideki Windows logları (Uygulama, Sistemler ve Güvenlik) incelenebilmektedir. Yeni olaylar geldiğinde "Yenile" düğmesi tıklatılarak görüntülenebilir. Filtrelemeye ve sıralama yapmaya izin vermektedir. Şekil 4.1'de Olay Görüntüleyicisi'nin filtreleme ekranı gösterilmektedir.

Şekil 4.1 Windows Olay Görüntüleyicisi Filtreleme Ekranı



Windows Olay Görüntüleyicisi'nin özelliklerinden bazıları şunlardır:

- Her bir olayla XML olarak çalışma.
- Olay aboneliklerini kullanarak uzak makinelerdeki olaylara abone olma.
- Filtreleri özel görünümler olarak kaydetme.
- Uygulama logunda özel olayları loglama.
- Bir olaya yanıt olarak bir görevi yürütme. (Scott, 2009)

4.2.1.3 Basit Log İşlemleri

Unix'te loglara *tail* ve *less* kullanarak gerçek zamanlı olarak bakılabilir ve birçok dosya görüntüleme komutu (*less*, *more*, *cat* gibi) kullanılabilir.

Aşağıda belirtilen amaçlar doğrultusunda basit araçlar kullanılabilir:

1. Log filtreleme: Sadece belirli şeyleri görmek için.
2. Log yeniden biçimlendirme: Görülmesi istenilen şekilde değiştirmek için.
3. Log özetleme: Özetlenmiş şekilde görmek için.

Yukarıda belirtildiği gibi, *tail* ile birlikte *grep* basit filtreleme sağlar. *Grep,* büyük log dosyalarında daha gelişmiş filtreleme konusunda yardımcı olabilir. *Grep* ile yaygın log filtreleme işlemlerine örnekler şunlardır:

 # grep –ev 'cron|systemd' /var/log/auth.log

Cron ve *systemd* içerenler hariç tüm mesajlar gösterilir.

 # grep –f desenler /var/log/auth.log

"desenler" dosyasındakilerle eşleşen tüm mesajlar gösterilir.

Bir log dosyasının ilk ve son bölümlerine bakmak için *head* ve *tail* komutları *grep* ile birlikte kullanılırlar. Örneğin:

 # tail – 500 /var/log/auth.log | grep rror

Bir mesaj dosyasının son 500 satırında "rror" dizesi olan kayıtları arar. *Grep* büyük küçük harfe duyarlı olması sebebiyle "Error" ve "error" olarak her ikisini de yakalamayı amaçlar. Benzer şekilde;

 # head – 500 /var/log/auth.log | grep rror

Bir mesaj dosyasının ilk 500 satırında "rror" dizesi olan kayıtları arar.

Awk ve *sort* komutu da eklenirse loglarda daha fazlası yapılabilir. Örneğin; aşağıda verilen komutla Linux'ta oluşturulan standart log mesajlarında dördüncü bölümde yer alan, logu oluşturan cihaz veya sistemlerin adları sıralanmış olarak görüntülenebilir:

 # cat /var/log/auth.log | awk '{print $4}' | sort

Yukarıdaki örnek, logları sıralamayı ve basit olarak özetlemeyi sağlar. A*wk,* log dosyalarını işlemek için karmaşık komut dosyalarının yazılmasına izin verir.

Windows'ta Olay Görüntüleyicisi, logları filtrelemeye, sıralamaya ve dosyalara çıkarmaya izin verir. Tipik bir Windows sisteminde filtreleme aşağıdakilerle sınırlıdır:

- Olay türü
- Olay kategorisi
- Olay kimliği
- Kaynak bilgisayar

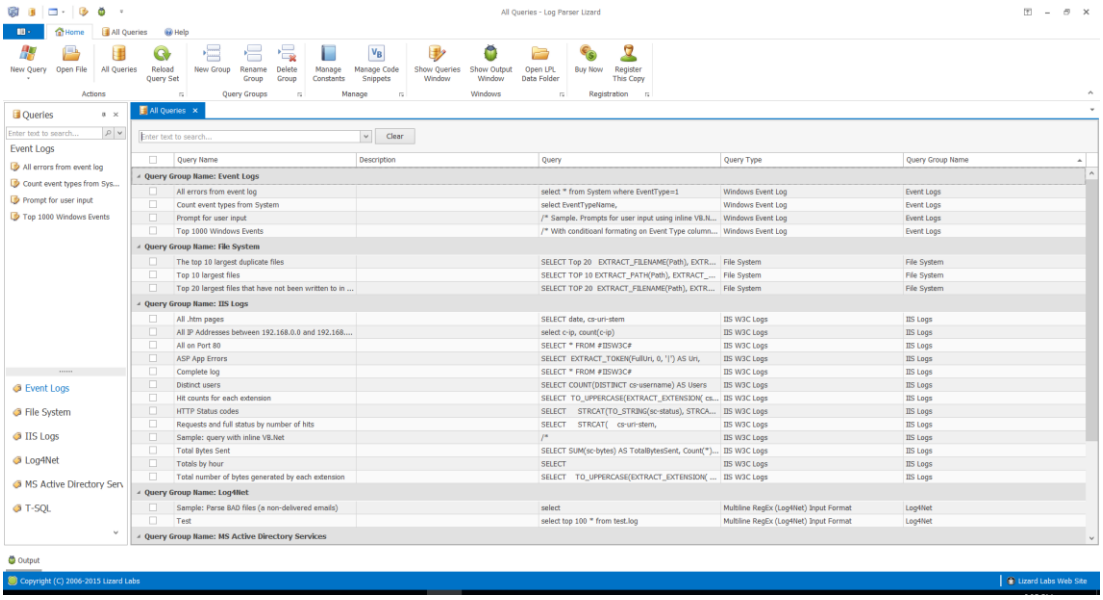
- Kaynak uygulama veya sistem bileşeninin adı
- Kullanıcı adı
- Tarih ve saat aralığı

Windows loglarının biraz daha gelişmiş analizinde *Log Parser* ile *Log Parser Lizard* birlikte kullanılabilir. *Log Parser*, Windows Olay Günlüğüne SQL benzeri arabirim sağlayan bir Microsoft Windows eklentisidir. Örneğin; Windows Olay Günlüğü'nden "Sistem" kategorisindeki tüm satırları alacak bir sorgu aşağıda verilmektedir:

*SELECT * FROM System*

Log Parser, tek başına herhangi bir analiz için kullanılması zor olan bir komut satırı aracıdır. *Log Parser Lizard*, *Log Parser*'da güzel bir arayüz oluşturan bir grafik arabirimidir. İki aracın kurulumu çok kolay ve basittir. Şekil 4.2'de Log Parser Lizard'ın arayüzü gösterilmektedir.

Şekil 4.2 Log Parser Lizard Arayüzü



4.2.2 Manuel Log İnceleme Sınırlamaları

Manuel log incelemesi ile basit araçlar ve komutların kullanımı, log analizinde yer almasına rağmen log analizinin ana mekanizması olarak düşünülmemesini gerektiren bazı sınırlamalar vardır. Burada yer alan sınırlamalar genel olarak şunlardır:

- Manuel log incelemesi giderek artan log dosyası boyutlarıyla ölçeklenememektedir. Büyük ölçekli kurumsal ortamlarda, genel olarak saniyede yüz binlerce log kaydı üretilir. Manuel log analizi, çok düşük bir loglama hızında bile etkisiz ve verimsiz olmaktadır.
- Manuel log incelemesi basit loglar için kolaydır; dosya açıklayıcı olduğunda manuel olarak yorumlanabilir, ancak daha belirsiz ve dokümante edilmemiş bir log kaynağı olması durumunda yaklaşım başarısızlıkla sonuçlanır. Her satırın yorumlanması için saatlerce süren araştırmalar gerekir.
- Basit araçlar ve elle inceleme, bilgi işlem ortamında ne olup bittiğinin bütününü asla veremeyecektir, ancak daha gelişmiş araçlar log verilerinden böyle bir sonuç üretebilirler.
- Düzensiz ve belirsiz biçimlere ilave olarak, birçok durumda analiz, birden fazla kaynaktan gelen logları birlikte ele almayı gerektirir. Bu tür bir etkinlik manuel olarak yapılabilir, ancak bir log dosyasına bakmakla bile karşılaştırıldığında gereken süreler büyük bir farkla artar. Bu tür bir ilişkilendirme, otomatikleştirilmiş araçlara bırakılmalıdır.

4.2.3 Analiz Sonuçlarının Eyleme Dönüştürülmesi

Analizin pratikteki gerçek amacı, tamamlandıktan sonra harekete geçmektedir. Bu kısımda log analizine dayalı olarak gerçekleştirilen bir dizi eylem kısaca gözden geçirilecek ve logların olaya müdahaleyle ilgisinden bahsedilecektir.

Analiz edilen logların çeşitli eylemlerle bağlantısının kurulması gerekir. Belirli bir satırda veya bir log dosyasında bir işaret görüldüğünde ne yapılması gerektiği gibi alt seviye ve log analizinin bir sonucu olarak hangi iyileştirmelerin yapılması gerektiği gibi üst seviye olmak üzere iki seviyedeki eylemler ele alınmalıdır.

4.2.3.1 Kritik Loglar Üzerindeki Eylemler

Bazı mesajlar için alınması gereken eylemler açık olarak belirlenmelidir. Bu mesajları içeren durum gerçekleştiğinde olaya müdahale planına hemen geçilmelidir. Genellikle çok fazla düşünülmeye ihtiyaç duyulmayan veya daha önceden düşünülmüş acil bir eylemi tetikleyen log mesajlarının önceden belirlenmiş olması gerekir. Tablo 4.1'de kritik log mesajlarında alınması gereken belirli eylemler gösterilmektedir.

Tablo 4.1 Kritik Log Mesajları Üzerinde Alınacak Eylemler

Kritik Log	Gereken Eylem
Sistem operasyonlarını etkileyen arıza	Sistem operasyonu kaybolursa, yedeklenen sistemi geri yükleyin.
Başarılı olan saldırı	Taviz verilen makine için olaya müdahale ve kurtarma işlemini başlatın.
Başarma şansının yüksek olduğu saldırı	Taviz verilen makine için olaya müdahale ve kurtarma işlemini başlatın.
Sistem kapasitesine veya azami değere ulaşılması	Ek kapasite sağlayın veya tüm sistemi kaybetmeyi göze alın.
Güvenlik ve erişilebilirlik sorunlarına neden olabilen sistem değişikliği	Değişiklikler yetkisiz olarak yapıldıysa sorunları önlemek için geri alın.
Sistem çökmesi	Yedeklenen sistemi geri yükleyin.
Çok sayıda başarısız oturum açma işlemi	Saldırgan şifreyi tahmin etmeyi başardıysa, bir taviz belirtisi olup olmadığını kontrol edin.
Donanım arızası	Sistem operasyonu kaybolursa, yedeklenen sistemi geri yükleyin.
Güvenlikle ilgili yapılandırma değişikliği	Değişiklikler yetkisiz olarak yapıldıysa geri alın.
Yetkisiz bağlantı tespit edilmesi	Sisteme kimlerin neden girdiğini araştırın.

(Chuvakin vd. 2013)

4.2.3.2 Kritik Olmayan Logların Özetleri Üzerindeki Eylemler

Güvenlik duvarı, yönlendirici veya bilgisayar mesajları, IDS'ten gelen kritik mesajlar gibi acil bir çağrı yapmamaktadır. Bir bağlantı bir güvenlik duvarı tarafından reddedildiğinde, genellikle logları izleyen kullanıcı tarafından anlık olarak eyleme alınmaz. Bununla birlikte, genellikle bu mesajların bir kombinasyonu dolaylı bir eylem başlatacaktır.

"Çoğu log dosyası işe yaramaz ve çoğu kurum log dosyalarını silmeyi ve verimli bir şey için disk alanından tasarruf etmeyi tercih eder." (Grafinkel, 2005) Günümüzde yasal uyumluluğun var olmasıyla beraber loglar uyumluluk nedeniyle saklanırken, bahsedilen iddia halen geçerlidir; çünkü ne kurumlar ne de uyumluluk denetçisi, loglara bakmayıp bulduklarına göre davranırsa loglar yine faydasız olmaktadır. Verizon'un yayınladığı bir Veri İhlal Raporunda, ihlal olayların çok yüksek bir yüzdesinin log mesajlarında delilleri bulunduğu; ancak hiç kimse loglara dikkat etmediği için keşfedilemediği yer almaktadır. (Verizon, 2011) Bu, log analizinin gerçekten başarılı olabileceğinin açık bir kanıtıdır.

Yalnızca log dosyalarını incelemek ve analiz etmek değil, aynı zamanda onlardan maksimum değeri elde etmek için de harekete geçilmesi gereklidir. Eylem dışı log mesajlarından eyleme götüren birçok yol vardır. En yaygın olanlar şunlardır:

1. Özet veya eğilime göre hareket etmek: Eylem dışı mesajların çoğu, bir eğilime veya özetten çıkan bazı yeni bilgilere bağlı olarak eyleme dönüşür.
2. İlişkili bir olay veya olaylar grubu üzerinde hareket etmek: Çoğunlukla, kritik olmayan olayların bilinen bir bileşimi, sıraları ve zamanlamaları nedeniyle kritik bir desene dönüşür.
3. Keşfedilen bir desen ya da kritik olmayan olayların alışılmadık dizisi üzerinde hareket etmek: Gelişmiş log analiz araçları kullanılarak, normal ve zararsız log kayıtlarından bazen yeni şüpheli bir desen çıkarılabilir.

Özetler ve eğilimler, log analizinin ortak bir sonucudur. Uzun bir log dosyası, "En Yüksek 10 Saldırı" ya da "En Şüpheli IP Adresleri" gibi çok sayıda yararlı şekilde özetlenebilir. Genellikle, böyle özet görünümler bir eylem gerektirecektir. Örneğin; "En Yüksek Bant Genişliği Kullananlar" raporunda şirketteki ilk üç kullanıcının mevcut bant genişliğinin yüzde 90'ından yararlandığı açık bir şekilde görülebilir. Bu bant genişliği özellikle P2P'de dosyaları paylaşmak veya işle ilgili olmayan materyalleri indirmek için kullanılıyorsa, disiplin eylemine neden olabilir. Benzer şekilde, bir yönlendiricinin CPU kullanım logunun uzun bir süre gözlemlenmesi, alışılmadık derecede yüksek etkinlik dönemlerini ortaya çıkarabilir, bu da muhtemelen ele geçirilen bir sistemle saldırganın iletişiminin keşfedilmesini sağlayabilir.

Korelasyon, çoğunlukla önemsiz olay gruplarının işaretlenmesini sağlayacaktır. Başarısız oturum açma işlemi ve başka bir yerde başlatılan bir uygulama, sistem ayrıcalıklarının içeriden istismar edilmesi veya bir sistem tavizi anlamına gelebilir. Örneğin; bir yöneticinin sisteme 01:00'da erişmesi fark edilirse, kötü bir şey ifade etmeyebilir, sadece biraz şüphe uyandırabilir. Daha sonra yönetici birkaç uygulamaya erişmeye çalışırsa ve bir veritabanına bağlanarak veri yüklemeye başlarsa şüpheler artabilir; ancak yine de kötü bir şey anlamına gelmez. Bununla birlikte, aynı sistemden geniş bir veri yükü indirmeye yönelik bir eylem gerçekleştirilirse, büyük olasılıkla bir saldırının gerçekleştiği ve harekete geçme zamanının geldiği bilinmektedir. Bu nedenle, böyle bir etkinliği izlemek için bir korelasyon kuralı oluşturmak, önemsiz görünen olaylarla ilgili bir eylem sağlayabilir.

Desen keşfi, loglardaki yeni bilgileri keşfetmenin ve sonuç olarak sıkıcı ve rutin logları eyleme dönüştürmenin etkili ve otomatize edilmiş bir yolunu sunar.

Örneğin; bir log dosyasının, farklı kaynak IP adreslerinden yinelenen bir bağlantı deseni içerdiği keşfedilebilir. Bu desen, bir kurum ağı üzerinde gizlice denenen yeni bir istismar aracı olarak yorumlanabilir. Bu durum daha sonra güvenlik ekibi tarafından bir dizi eyleme neden olabilir.

Tablo 4.2'de eylem dışı loglarla eylemlerin nasıl ilişkilendirildiğine dair daha fazla örnek gösterilmektedir.

Tablo 4.2 Eylem Dışı Loglarla İlişkilendirilen Eylemler

Hesaba Katılabilir Log	**Eylem Yolu**	**Gereken Eylem**
Sistem durumu mesajları	İlişki	Bir sistem durumu mesajı, yetkisiz sistem kullanımını gösterebilir ve bu durum bir soruşturma ve daha sonra disiplin işlemine neden olabilir.
Saldırı girişimleri	İlişkili özet	Bir saldırı deseni, ileri düzey bir saldırganın güvenlik savunmalarını kırmaya çalışmak istediğini gösterebilir.
Düşük etkili saldırılar	Eğilim	Düşük etkili saldırıda görülen artış, daha zararlı saldırılarda yaşanan artış ile çakışabilir ve bu durum üzerinde eyleme neden olabilir.
Sistemin bir parametreye oranla yüksek bir değere ulaşması	Eğilim	Bu tür mesajların keskin bir şekilde artması ya bir saldırı ya da genel sistem kararsızlığı anlamına gelir; her ikisi de performansı iyileştirmek için bir eylem gerektirir.
Çeşitli sistem değişiklikleri	İlişki	Sistem değişiklik mesajının bir deseni, yetkisiz sistem yeniden yapılandırmasını gösterebilir ve bu durum bir soruşturma ve daha sonra disiplin işlemine neden olabilir.
Sistem başlatma/kapatma	Özet	Belli bir zaman aralığında çok sayıda sistem kapanması, sistemin kararsız hale geldiğini ve düzeltilmesi gerektiğini gösterebilir.
Başarılı giriş	İlişki	Başarısız girişlerden sonra gelen başarılı bir giriş, bir kaba kuvvet şifre saldırısının başarılı olduğunu gösterebilir.
Donanım durumu mesajı	Özet	Çeşitli sistemler üzerinde donanım durum mesajlarını özetlemek, en fazla probleme sahip olan sistemlerin belirlenmesine yardımcı olabilir, böylece optimizasyon ve performans elde edilebilir.
Bağlantı kuruldu/sona erdi	Keşif	Log madenciliği, yeni bir hack aracına işaret eden ve güvenlik savunmalarını güçlendirmeye ihtiyaç duyan bir bağlantı deseninin keşfedilmesine yardımcı olabilir.

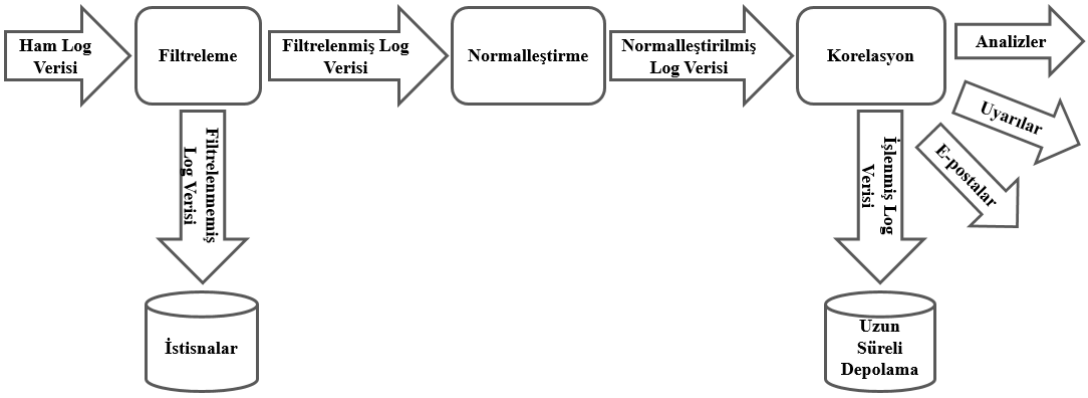
(Chuvakin vd. 2013)

4.3 Filtreleme, Normalleştirme ve Korelasyon

Loglama ve log yönetimi konusunda sistem yöneticileri, zamanının büyük bir kısmını olası sorunlar için logları incelemeye harcamaktadır. Geçmişte, bu görev büyük oranda manuel yapılıyordu. Günümüzde log analiz sistemleri, bu yükü hafifletmeye yardımcı olmak için otomatikleştirilmiş mekanizmalar sağlamaktadır.

Bu kısımda üç temel kavram yer almaktadır; filtreleme, normalleştirme ve korelasyon. Filtreleme, ham log verilerini alarak aralarından tutulması istenilenleri belirleme eylemidir. Normalleştirme, filtrelenen log verisinin zenginleştirilmesini sağlar. Normalleştirilmiş veri, korelasyonun bir girdisidir. Korelasyon, normalleştirilmiş veri parçasını eşleştirme eylemidir. Şekil 4.3'te bu işlemin mantıksal akışı gösterilmektedir.

Şekil 4.3 Filtreleme, Normalleştirme ve Korelasyon İçin Basit Akış



Sürecin her adımı kısaca açıklanacak olursa:

1. **Ham Log Verisi:** İşlemin ilk girdisidir.
2. **Filtreleme:** Filtreleme aşamasında, önem verilen ve önemsenmeyen log mesajları aranır. Önemsenmeyenler, sistem üzerindeki yükü azaltmak için atılabilir. Bunlar, Şekil 4.3'te, istisnalar deposuna giden bir okla gösterilmektedir. İstisnalar deposu, az ilginç olan log mesajlarını daha sonra gözden geçirmek için kullanılabilir.
3. **Normalleştirme:** Bu adımda ham log verileri alınıp, çeşitli öğeleri (kaynak ve hedef IP gibi) ortak bir biçimde eşleştirilmektedir. Bu, korelasyon adımı için önemlidir. Ham log mesajı normalleştirildiğinde, sonuç bir olaydır.

Normalleştirme sürecindeki bir diğer adım ise kategorize etmektir. Bu, bir log mesajının daha anlamlı bir bilgi parçasına dönüşmesi anlamına gelir.

4. **Korelasyon:** Korelasyon genellikle önemsiz her bir olay grubunun işaretlenmesini sağlar. Başarısız bir oturum açma ve başka bir yerde başlatılan bir uygulama, sistem ayrıcalıklarının içeriden kötüye kullanımı anlamına gelebilir. Kural tabanlı ve istatistiksel olmak üzere korelasyonun iki temel formu vardır.
5. **Eylem:** Bir eylem, genellikle bir korelasyon oluştuktan sonra yapılan bir faaliyettir. Şekil 4.3'te, eylemin birkaç çeşidi gösterilmektedir:
   a. Analizler: Bir log izleme arabirimi varsa, hemen ilgilenilmesi gereken yüksek öncelikli olaylar buraya gönderilebilir.
   b. Uyarılar: Genellikle bir analiste bir olayı gönderen karma bir yapıdır. Bir uyarı, yüksek düzeyde bir şeyin gerçekleştiğini gösteren bir grup olay olabilir.
   c. E-postalar: Nöbette olan personeli uyarmak için bir araç olarak kullanılabilir.
   d. Uzun süreli depolama: Log verilerinin ve normalleştirilmiş olayların saklandığı yerdir. Raporlama, denetim, uzun vadeli analiz vb. için bir ön şarttır. (Chuvakin vd. 2013)

4.3.1 Filtreleme

Filtreleme, ilgilenilen log verilerinin tutulmasını veya atılmasını sağlar. Üreticiler log mesaj biçimlerini birbirleri arasında standartlaştırmazlar. Yine de her bir üretici, temel mesaj aktarımı olarak syslog'u kullanabilir. Çoğu log analiz sistemi, filtrelemeyi gerçekleştirmek için mekanizmalar sağlar; ancak filtrelemekten daha çok fazla veri bulundurma hatasını yapmamak önemlidir.

4.3.2 Normalleştirme

Filtrelemeden bir sonraki adım normalleştirme işlemidir. Bu aşamada hangi log verilerinin saklanmak istendiği bilinmektedir. Normalleştirme, bilinen log mesajlarını alarak bileşenlerine (zaman damgaları, IP adresleri vb.) ayrıştırıp bunları ortak bir biçime çevirmek demektir. Bu ortak biçim tarihsel olarak, ilişkisel veritabanı sistemi (RDBMS) veya başka bir alt seviye biçimin (diskte ikili vb.) kullanımı vasıtasıyla

gelmektedir. Bununla birlikte büyük veri ve NO SQL hareketiyle giderek artan sayıda üretici, ölçeklenememesi sebebiyle veritabanlarından uzaklaşmaktadır.

Ham log mesajını normalleştirme adımları şunlardır:

1. Kullanılan ürünler için dokümantasyonun edinilmesi.
2. Ham log verisinin nasıl göründüğü ve her alanın ne olduğu hakkındaki tanımlamalar için dokümantasyonun okunması.
3. Verileri normalleştirmek için uygun ayrıştırma ifadesinin bulunması. Çoğu log analiz sistemi, verileri ayrıştırmak için bir düzenli ifade (regular expression) uygulaması kullanmaktadır.
4. Ayrıştırma mantığının örnek ham log verileri üzerinde test edilmesi.
5. Ayrıştırma mantığının uygulanması.

Normalleştirilmiş olaylar için kullanılan saklama mekanizmasına bakılmaksızın, bazı alanlar diğerlerinden daha yaygın ve kullanışlıdır. Bu alanlar şunlardır:

- Kaynak ve Hedef IP adresleri: Korelasyon işlemi sırasında çok faydalıdır.
- Kaynak ve Hedef Portlar: Hangi hizmetlere erişmeye çalışıldığını veya erişildiğini anlamak için kullanılır.
- Taksonomi: Bir log mesajının anlamını kategorize etmek ve kodlamak için bir yoldur. Örneğin; tüm cihaz üreticileri bir çeşit giriş mesajı üretir. Bunlar tipik olarak giriş başarıları, başarısızlıkları, denemeleri vb. olarak eşleştirilir. Başarılı bir giriş için örnek bir taksonomi; giris.basarili olabilir. Bir taksonominin kritik olmasının nedeni, log mesajlarını üreten belirli bir üreticiyi önemsemeksizin mesajları birbirine gruplamaya izin vermesidir.
- Zaman Damgaları: Log dünyasında genellikle iki tür zaman damgası kullanılır; log mesajının cihazda oluşturulduğu zaman ve loglama sisteminin log mesajını aldığı zaman.
- Kullanıcı Bilgileri: Sağlandığı takdirde, herhangi bir kullanıcı bilgisini (kullanıcı adı, komut, dizin konumu vb.) yakalamak genellikle iyi olur.
- Öncelik: Bazı log mesajları, log mesajında bulunan bir çeşit öncelikle gelir. Bu, log mesajının önceliği noktasında üreticinin bir değerlendirmesidir; ancak, bu konuyla ilgili düşüncelerle eşleşmeyebilir. Normalleştirmenin bir

parçası olarak, belirli bir log mesajının ortamı nasıl etkilediğini anlamak gerekir. Öncelik için tipik değerler düşük, orta ve yüksektir.

- Ham Log: Normalleştirme sürecinin bir parçası olarak, ham log verilerinin tutulması istenebilir. Bu, normalleştirilmiş olayın geçerliliğini sağlamak için kullanılır. Bir başka kullanım durumu ise log tutmadır. Ham log verilerini belirli bir süre boyunca saklamak için bir gereksinim olabilir. Bunun için iki çözüm vardır: ham logları normalleştirilmiş olayın bir parçası olarak saklamak veya diskte saklayarak normalleştirilmiş olaydan ham log mesajına geri dönmek için bir araç sağlamaktır.

4.3.3 Korelasyon

Korelasyon, bir sözlükte şu şekilde tanımlanmaktadır: "İlişkili olma durumu veya ilişkisi; özel olarak: olgu ya da olaylar arasında ya da değişime eğilimli, tek başına şansa dayalı olarak beklenmeyecek şekilde birlikte bulunma ya da birlikte görülen matematiksel ya da istatistiksel değişkenler arasında var olan bir ilişki". (Merriam-Webster, correlation)

Bu tanım log veri analiziyle ilişkili olan korelasyon amacını tam olarak yakalayamaz. Log analizi için korelasyon, basitçe tek bir olaya bakarak ne olduğunu anlamaya çalışmanın aksine, tek bir bilgi parçasının içinde benzerliği olan veya olmayan birkaç olayı bir araya getirerek çok daha büyük bir eylemin gerçekleştiğini ifade etmektir. Bu, güvenlik, ağ yöneticileri ve analistler için ilginç olan durumları modelleyen bazı dillerde kuralların oluşturulmasıyla gerçekleştirilir. Örneğin; güvenlik duvarlarından ve IDS'lerden log verileri alınırsa, aşağıdaki kuralla güvenlik duvarı ilkesi ihlalini izleyen arama girişimleri yakalanır:

Eğer sistem X.olayTuru = porttarama olduğu X olayını görürse
bunu takiben
Y.kaynakIp=X.kaynakIp ve Y.hedefIp=X.hedefIp ve Y.olayTuru = reddet olduğu bir Y olayından sonra
biseylerYap

Sözde kodla ifade edilen bu kural, birbirinden ayrılmış iki olayın birbirine nasıl bağlanabileceğini detaylandırır. Kural, iki farklı ağ güvenlik sisteminden gelen iki farklı olaya bakar. Birincisi (X), IDS tarafından tespit edilen bir port tarama olayıdır. İkincisi (Y), bazı bilgisayarların hariç tutulduğu güvenlik duvarı politikasına sahip olan sunucuya bir girişimde bulunulduğunda güvenlik duvarının reddetme olayıdır.

Kuralda yer alan, biseylerYap eylemininin (uyarı, e-posta vb.) ortaya çıkması için porttarama olayından sonra belirli koşullara sahip reddet olayının gerçekleşmesi gerekir. Reddet ve porttarama ifadeleri, X ve Y olay türleri için basit kategorizasyondur (taksonomi).

Durumsal kural motorları, yöneticinin ve analistin, rastgele dizileri veya desenleri algılayabilen kurallar oluşturmalarını sağlar. Bir kişinin bütün syslog dosyalarına bakarak bu olayları ilişkilendirmeye çalışması imkânsızdır. Kural motoru, bir veya daha fazla kuralla eşleşen olay dizileri için veritabanını basitçe araştırabilir. Bu çok maliyetli bir yoldur; çünkü yapılabilmesi için büyük miktarda SQL sorgusu ve bellekte olay verisinin kullanılması gerekir. Çok fazla G/Ç işlemi ile birlikte veritabanını kullanmaya çalışan diğer uygulamalar da olumsuz etkilenir.

Daha iyi bir çözüm, olay verisini uzun vadeli depolamaya gitmeden önce ele almaktır. Veriler toplayıcıya geldiğinde veya analiz sunucusunda, bir mesaj sırasıyla beslenerek kural motoru tarafından tüketilir ve işlenir. Kural motoru olay verilerini tüketirken, bu olaylara kurallar uygulayabilmek için olayları hafızasında tutması gerekir. Tutulan bu olaylar belirli bir süre sonra hafızada yaşlandırılmazsa, RAM'in fazla miktarda tüketilmesine sebep olur. Bu nedenle, verilerin geçersiz hale gelmesiyle birlikte olayları yaşlandırmak gereklidir. Bunu gerçekleştirmenin bir yolu, motordaki her kural için bir yaşam süresi (TTL) belirlemektir. Daha özel olarak, kuraldaki her adım için de TTL ayarlanabilir. Örneğin; yukarıdaki kuralda, olay türü port tarama olan bir olay elde edilinceye kadar olaylar izlenmeye başlanmaz. Bu olay gerçekleştiğinde TTL işlemeye başlar. Belirlenen TTL değerine ulaşıldığında kuralın bir sonraki aşamasının gereksinimlerini karşılayan bir etkinlik görülmediği takdirde kural sıfırlanabilir ve bu kural için izlenen tüm etkinlikler atılabilir.

5. LOG VERİ MADENCİLİĞİ

Log analiz tekniklerinin büyük bir çoğunluğu, bir analistin loglarda ne aradığına dair belirli bir şeyi bilmesini gerektirir. Bu durumda, log analiz sürecinin tüm aşamalarında analiz için mevcut olan belirli log türleri ve güvenlik konusunda uzmanlık gerekir. Buna ilave olarak, görevi başlatmak çok fazla sabır gerektirir; çünkü anormal çizgilerin bulunması için uzun bir süre loglara göz atılması gerekebilir.

Bu bölümde, bütün analiz aşamalarında uzmanlık gerektirmeyen, güvenlik için log dosyalarında ilginç desenleri keşfetme yöntemleri anlatılacaktır. Ele alınacak teknikler birçok açıdan veri madenciliği ile benzer olduğundan, konunun anlaşılabilmesi amacıyla kısaca veri madenciliğinden de bahsedilecektir.

5.1 Veri Madenciliği

Veri madenciliği hakkında bir sözlük tanımı şu şekildedir: "Veri madenciliği, büyük miktarda veride ilginç ve kullanışlı desenleri ve ilişkileri keşfetme sürecinde veritabanı ve bilgisayar biliminde bilgi keşfi olarak da adlandırılır. Veri setleri olarak bilinen büyük dijital koleksiyonları analiz etmek için veritabanı yönetimiyle istatistik ve yapay zekâ (sinir ağları ve makine öğrenimi gibi) alanındaki araçları bir araya getirir." (Encyclopedia Britannica, Data Mining)

Bu konuda diğer tanımlar da şu şekildedir: "Veriden, önceden bilinmeyen faydalı bilginin kolay olmayan içerik çıkarımıdır" (Bray ve Cid, 2008) ve "Büyük veri kümelerinden veya veritabanlarından yararlı bilgileri çıkarma bilimidir." (Bejtlich, 2004)

Bu tanımlar arasından bir çıkarım yapılırsa:

- Veri madenciliği, büyük bir veri havuzu ile ilgilenir.
- Bu tür veriler, bir bilgisayar sisteminde makine okuyabilir biçimde bulunan yapılandırılmış verilerdir (ilişkisel veritabanı sistemi gibi).
- Eldeki veriler ilginç bir sonuca götürebilir de götürmeyebilir de.
- Veriyi aramayı veya başka türlü analiz etmeyi gerektiren, ulaşmaya çalışılan bazı faydalı sonuçlar vardır.

Veri madenciliğini yukarıdaki maddeleri kullanarak planlamak, log analiziyle arasındaki benzerlikleri ortaya koyar. Log veri havuzu büyüktür ve bu logları üreten

bilgisayar sistemlerinin söylemeye çalıştıklarını anlamak gerekir. Sistemlerin söylemeye çalıştıkları bazı durumlarda gün yüzüne çıkar, bazı durumlarda ise anlamsız ya da alakasız bir bilgi denizi içinde kaybolur. Log analizi ve veri madenciliği arasındaki benzerlik, veri madenciliğinin eldeki zorluklar için yararlı olabileceği konusunda bir fikir oluşturur.

Veri madenciliği dolandırıcılık analizinden risk yönetimine kadar pek çok alanda kullanım imkânı bulmaktadır. Veri madenciliği alanı, çok çeşitli bilimsel yöntemleri, yazılım uygulamalarını ve kullanım senaryolarını kapsar. Burada güvenlik log analiziyle ilgili veri madenciliği kısımları ele alınacaktır.

Genel olarak, yaygın iki veri madenciliği yöntemi vardır: tahmine dayalı (predictive) ve tanıma dayalı (descriptive). Tanıma dayalı veri madenciliği, veri setlerinin bilgilendirici bir şekilde tanımlanmasını sağlar, böylelikle neler olduğunu açıklamaya çalışır. Diğer taraftan, tahmine dayalı veri madenciliği, mevcut olanlara dayalı olarak henüz mevcut olmayan verilerle ilgili tahminler yapmaya olanak tanır.

Veri madenciliği işlemi genellikle aşağıdaki adımlara sahip olacak şekilde tanımlanmaktadır:

1. **Veri ile ilgili alanda uzmanlık edinme:** Veri madenciliğinde diğer alanlarda olduğundan daha fazla ne yaptığını bilmek önemlidir. Kullanıcı, çıkarılmış veriyi anlamadığı sürece veri madenciliği araçları sonuç üretmeyecektir.
2. **Hedefi tanımlama:** Keşif teknikleri bilinmesine rağmen, neden hâlâ verilerin toplanması, hazırlanması ve çıkarılması istendiğinin ve sonuçlarla ne yapılacağının bilinmesi gerekir.
3. **Toplamayı planlama ve daha sonra verileri toplama:** Gelecek veri madenciliği çabalarının temelini oluşturur. Veriler kontrolün ötesinde büyümek için kötü bir eğilime sahiptir.
4. **Veri önişleme ve temizleme:** Veri madenciliği işleminin etkili olması veya gerçekleşmesi için gereklidir. Veri madenciliği uzmanları bu adımın tüm madencilik çabalarının yüzde 60'ını oluşturduğunu tahmin etmektedir. Bu aşamada, eksik bilgi parçalarının yanı sıra yinelenen veri noktalarıyla

ilgilenilmesi, gürültülü verilerin düzeltilmesi, veri kümesindeki diğer tutarsızlıkların belirlenmesi ve düzeltilmesi gerekir.

5. **Veri azaltma ve dönüşüm:** Verilerin ek boyutlarını düşürmeyi, veri kümesinin bölümlerini kaldırarak verilerin daha kolay yönetilebilir olmasını amaçlayan algoritmalar uygulamayı içerir.
6. **Uygulanacak yöntemi seçme:** Seçilen yöntem, yukarıdaki birkaç adımda tanımlanan hedefler tarafından yönlendirilir.
7. **Seçilen yöntemi uygulayan belirli bir algoritma seçme:** Araştırmacılar ve araç üreticileri, desenleri aramak için mevcut algoritmaların performansı üzerinde çalışmanın yanı sıra yeni algoritmalar da geliştirmektedir.
8. **Madencilik yazılımını çalıştırma:** Sonuçları alma adımıdır. Sonuçlar görselleştirildiğinde bir resim görmeyi veya verilerin bir tablo ya da başka metinsel gösterimini ifade edebilir.
9. **Sonuçların gerçekte ne anlama geldiğini bulma:** Basit metin raporlarından süslü görselleştirmelere kadar, önceki tüm adımların faydalı olup olmadığının anlaşıldığı adımdır. (Chuvakin vd. 2013)

Yukarıdaki adımlar normal log analiz sürecine çok yakındır. Bu nedenle, bir sonraki kısımda, log analizinde kullanılan teknikler gibi veri madenciliğinin nasıl kullanılabileceğini gösteren örnekler verilecektir.

5.2 Log Madenciliği

Log madenciliği teknik olarak bir tür analiz olsa da sıradan log analizinin tükendiği yerde başlamasıyla farklılaşmaktadır. Burada, veri madenciliği tekniklerinin log analiz etme amacı için nasıl uygulanabileceği gösterilecektir.

Log verisi için veri madenciliği uygulayarak genellikle aşağıdaki kazanımlar elde edilmeye çalışılmaktadır.

- Daha iyi cevaplar ve tahmin gücü sağlamak için log analizinin kalitesini artırmak.
- Pahalı uzmanlık gerektirmeden gelişmiş ve etkili yöntemleri hazırlamak.

Güvenlik Ağ İzleme (NSM), loglar ve paket yakalamaları gibi ağ trafiği bilgilerini kapsar. NSM yaklaşımının ana ilkesi, çok yetenekli analistlerin, loglar,

alarmlar ve paket yığınları (dump) gibi gelen verilere dair iç görü kazanmak için iyi optimize edilmiş bir analiz sürecini izlemesi ve ilgili araçları çalıştırmasıdır. Bununla ilgili ufak bir sorun, bu tür analistlerin eldeki görevi yerine getirememeleri veya basitçe elverişsiz olmalarıdır. Sonuç olarak yöntem, eğitimli, akıllı ve motive edilmiş analisti her zaman konsolun önünde kullanmaya dayandığından ölmeye mahkûmdur. Dolayısıyla bu yaklaşım, büyük olasılıkla iyi finanse edilen ve güvenlik düşüncesine sahip kurumlar alanında kalacaktır. (Bejtlich, 2004)

Veri madenciliği ve diğer gelişmiş otomatik analitiği kullanarak analiz yükü, son derece yetenekli ve bulunması zor analistlerden uzak, yazılım ve otomatik sistemlere taşınacaktır. Böyle bir analizin dâhil olması, belirli madencilik yöntemleri tasarlandığında, yalnızca sürecin ilk aşamalarında kritik olacaktır. Bu durumda uzmanlar, madencilik işleminin özelliklerini tanımlayabilir ve daha az yetenekli işlemlerin algoritmayı çalıştırmasına ve verimlilikte bir azalma olmadan sonuçlara göre hareket etmesine izin verebilir.

Log madenciliği terimi, veri madenciliği tekniklerinin gelişmiş log analizine uygulanması anlamına gelir. Amaç, insan çabasını azaltmak ve log analizinde otomatik sistemin rolünü arttırmaktır. Log analizi aşağıdaki gibi zorluklarla karşı karşıya kalır.

- **Çok fazla veri:** Log analiz sistemlerine ve analiste çok fazla veri gelmektedir ve böylece cevapları alma imkânı mümkün olmamaktadır. Bu sebeple gelen veri seliyle uğraşmak için araçlara ihtiyaç vardır.
- **Yeterli veri olmaması:** Çeşitli nedenlerden dolayı kritik veri parçalarının eksik olması, log analizini daha zor hale getirmektedir.
- **Yanlış alarmlar:** Loglar, gerçekliği ölçülebilir bir şekilde yansıtmayan mesajlarla doludur (IDS'te yanlış pozitifler bu kategoriye girmektedir).
- **Çoğaltılmış veriler:** Farklı loglar aynı olayı ifade edebilir. Bu durum, farklı log kaynakları arasında zaman senkronizasyonun eksikliği ile daha da karmaşıklaşır.
- **Veri almanın zorluğu:** Bir cihaz, tescilli biçimde oldukça yararlı log kayıtları oluşturabilirse de syslog veya Windows Olay Günlüğü gibi standart log biçimlerinin eksik olması nedeniyle merkezi log toplayıcıyla

iş birliği yapamaz. Benzer şekilde, detaylı sunucu denetim kayıtlarının alınması bir sorun olabilir. (Chuvakin vd. 2013)

Zorluklarla başa çıkmak için birçok teknik geliştirilmektedir. Bütün bu teknikler olmasına rağmen log madenciliğinin yapılmasının başlıca nedenleri şunlardır:

- İnsanın yaptığına benzer şekilde desen tanımayı mümkün hale getirerek deneyimli analistlere olan ihtiyacı azaltır ve sadece analizin erken aşamalarında bu tür uzmanlığı gerektirir.
- Geleneksel yöntemlerle etkili bir şekilde analiz edilemeyen nadir verilerle uğraşır. Çok büyük veriler göz önüne alındığında log madenciliği gibi tamamen otomatikleştirilmiş yaklaşımlar daha uygun olacaktır.
- Radarın altına gizlenen şeyleri algılar. Log madenciliği, saldırı izlerini tespit etme verimliliğini artırır.
- Makinelerle sonuç üreterek sadece insanın yapabileceği görevler de otomatikleştirilebilir. Bu şekilde insanlar neler olduğunu anlamak için uğraşmak yerine elde edilen sonuçlar üzerinde hareket edecektir.
- Önceden oluşmuş olan şeylerle başa çıkmanın yollarını bulmak yerine, sorunlar öngörülmeye çalışılır. Veri madenciliği, bu tahmin için garantili bir yol sağlamasa da diğer yöntemlerden daha iyi bir yol sunar.

Günümüzdeki eğilim, bu tür işlerden insanları çekmeye yönelik olmasına rağmen, algoritmaların her zaman uzmanlar tarafından tanımlanması ve ayarlanması gerekecektir. Bununla birlikte, madencilik teknikleri operasyon personelinin beceri gereksinimlerini azaltacaktır.

5.3 Log Madenciliği Gereksinimleri

Veri madenciliğini log verilerine uygulamadan önce nelerin gerekli olduğuna bakılması gerekir. Log madenciliği için birçok gereksinimin herhangi bir log analizinin ihtiyacı ile benzer olduğuna dikkat edilmesi gerekir. Bununla birlikte, log verilerinin madencilik için uygun hale getirilmesi veya opsiyonelden zorunlu gerekliliklere dönüştürülmesi gibi bazı ilave faktörler vardır. Bunlar:

1. **Veri merkezileştirme:** Sadece bir yere bakmak, filtreleme ve özetleme gibi düzenli log analizleri için iyidir ancak madencilik algoritmaları tek bir analistten çok daha fazla veri sıkıştırabildiğinden log madenciliği için kritik hale gelmektedir.
2. **Normalleştirme:** Merkezileştirilen veri kaynaklarına bakmak için tek tip bir bilgi biçimi gerekir. Normalleştirme, loglardaki ortak alanlara bakarak gerçekleştirilir. Bunlar sıklıkla şunları içerir:
    a. Zaman
    b. Kaynak
    c. Hedef
    d. Protokol
    e. Port
    f. Kullanıcı adı
    g. Olay/saldırı türü
    h. Alınıp verilen baytlar
3. **İlişkisel depolama:** Burada ilişkisel veri depolaması çok önemlidir; ancak basit analiz ve filtreleme gerçekleştirilecekse bu madde dâhil edilmeyebilir.

Böylelikle normalleştirilmiş ve merkezileştirilmiş veriler, log veri madenciliği algoritmalarına tabi tutulabilir.

5.4 Neler Çıkartılabilir

Veri madenciliği yöntemleri, ne arandığı konusunda fikir sahibi olunmadığı durumlarda çok yararlıdır. Aksi takdirde, ihtiyaç duyulan şeyler filtrelenebilir veya aranabilir. Burada amaç, ilginç bir şeyler bulmaktır.

Bir sistem yöneticisinin veya bir güvenlik analistinin ilginç bulduğu ve log madenciliğinde keşfedilmesi beklenen örneklerden bazıları şunlardır:

- **Zararlı yazılım yaymakta olan enfekte sistem:** Birçok durumda belirgin olmakla birlikte, enfekte olmuş ve daha sonra enfeksiyonu kurumsal çapta ve hatta internet genelinde yayan sistemler her güvenlik yöneticisinin öncelik listesinde üst sırada yer almaktadır.

- **Tehlikeli sistem:** Saldırganların veya onların kullandıkları zararlı yazılımların ağdaki bir sistemi ele geçirdiğini bilmek her güvenlik uzmanının ilgisini çekebilir.
- **Başarılı bir saldırı:** Saldırgan sisteme ulaşmayı başarırsa, bunu bilmek ilgi çekicidir. Bir önceki maddeyle alakalı iken genelde saldırganın deneme girişiminden tam taviz ve sistem kullanımına kadar geliştirdiği saldırının erken aşamalarını tanımlar.
- **İçerden istismar ve fikri mülkiyet hırsızlığı:** Her şeyi çalan hacker ve solucanlarla karşılaştırıldığında içeriden ağ istismarı daha ilkeldir. Bununla birlikte, içerdekiler çok daha fazla zarara uğratma potansiyeline sahiptir. İçerdekilerin istismarlarını tespit etmek, ortalama zararlı yazılımdan daha zordur. İçerideki saldırılara dair açık kanıtlar, yöneticilerin ve analistlerin ilgisini kesinlikle çekecektir.
- **Gizli kanal/gizli arka kapı iletişimi:** Gizli kanallar, bu tür bir çalışma yapılmadığı sürece ağda düzenli olarak kullanılmaz. Bu nedenle, bunu bilmek ağ güvenlik yöneticilerinin ilgisini çekebilir.
- **Sorgulamada artış:** Fazla hassas olmayan kurumların ağları, internet sorgulama etkinliğini gürültü olarak nitelendirirken, bu tür etkinliklerin loglara yansıyan belirli artışlarının saldırıların öncüsü olarak bilinmesi ilgi çekici olabilir.
- **Sistem çökmesi:** Bir hizmetin servis dışı kalması her zaman kendini belli etse de (herhangi bir hizmete erişilemediğinde fark edilir), bir sistem yöneticisi tüm sistemlerin çalışma süresini izlemeyebilir.

İlginç olma ölçütlerini bir bilgisayar için tanımlamak çok zordur ancak log madenciliği ile bu mümkündür. Yukarıdaki ilginçlik ölçütleriyle eşleşmeyen, ilginç olmayan, beklenmedik ve eyleme geçilebilir yaygın örnekler şunlardır:

- **Sorgular** (beklenmedik değil): Ağdaki sorgular ve taramalar sürekli gerçekleşir ve insanlar bunları sürekli çoğaltır. Bunlardan haberdar olunmalı; ancak bunları aramak için kaynaklar harcanmamalıdır. Aynı zamanda, bu tür sorguların sayısının (günlük, haftalık, aylık vs.) değişiminin ilginç olma şansı yüksektir.

- **Yaygın başarısız saldırı** (beklenmedik değil): Güvenlik altyapısı sağlamsa başarısız saldırıların görülmesi beklenebilir. Sorgulara benzer şekilde bunlar da bilinmeli, ancak bunları aramak için kaynaklar harcanmamalıdır.
- **Normal mesaj** (beklenmedik değil): Loglar, bazı rutin süreçlerin ve normal olayların tamamlandığını gösteren, denetim ve diğer amaçlar için kaydedilen mesajlarla doludur. Değişiklikler burada da önemlidir; normal mesajların gelmemesi veya daha az/çok sıklıkla gelmesi ilginç olabilir.
- **Engellenmiş saldırı** (eyleme geçilemez): Başarısız bir saldırıya benzer şekilde, güvenlik önlemleri bir saldırıyı engelliyorsa, herhangi bir işlem yapılması gerekmez. Bu saldırı bir keşifin sırada olduğunu gösterse bile hâlâ ilginçlik kriteriyle uyuşmamaktadır.
- **Sistem durumu güncellemeleri** (eyleme geçilemez): Normal bir olaya benzer şekilde, büyük olasılıkla hiçbir eylem gerektirmez. Aynı zamanda, alışılmadık zamanlarda gerçekleşen sistem durumu güncellemeleri ilgi çekici olabilir. (Chuvakin vd. 2013)

6. RAPORLAMA VE ÖZETLEME

Herhangi bir süre için log verisine bakmak zorunda kalan herkes, ham loglara bakmanın işe yaramayacağını bilir. Bununla birlikte, log analizi görevlerinin çoğunluğu belirli bir süre boyunca veya belirli bir sistem grubunda log verilerinin özetlerine ve raporlarına bakmakla ilgilidir. Analiz edilecek veri, özetleme ve birleştirme yapılmasını gerektirir. Çoğu insanın logları incelerken ilk deneyimi, log verileri üzerindeki raporlar olmaktadır. Ticari log analizi ve SIEM araçlarının kullanıcıları bile orijinal log verilerine baktıklarından daha çok raporlara bakmaktadır.

Özetleme, bakılması gereken veri miktarını düşürmekte; ancak veriler özetlendiğinde yararlı bilgileri de kaybedilmektedir. Ayrıca, kullanmak için seçilen özetlerin türüne bağlı olarak da ilgili veriler kaybedilebilir. Örneğin; ilk 10 kullanıcıya bakıldığında alt 10 kullanıcıların da ilginç olabileceği unutulmamalıdır.

Raporlar oldukça fazla olduğundan hangi raporun seçileceği konusunda zorluklar vardır. Bunun üzerine pek çok kurum, gelişmiş tehditler (zararlı yazılım gibi) yanında, birçok mevzuata uyum çerçevesi (PCI DSS, HIPAA, FISMA ve diğerleri) ile mücadele etmektedir. Ayrıca içerdeki kötü niyetliler, bir kuruma zarar vermek veya dolandırmak için ek fırsatlar elde etmektedirler. Aynı zamanda, işletmeler ve kamu kurumları için bilgi teknolojisinin önemi muazzam bir şekilde büyüdü ve daha da büyüyecektir. Raporlar bu noktada çok fazla önem kazanmaktadır.

Log analizi için raporlamanın geniş kullanımıyla birlikte çok sayıda olası raporlar arasından en iyi raporların bir listesini veya en azından endüstride en geniş uygulanabilirliği olanları oluşturmaya çalışmak önemlidir.

Bu bölümde, oluşturulan böyle bir liste incelenecektir. Raporlar, çoğu kurum için uygulanabilir olarak altı ana kategori şeklinde düzenlenebilir. Bu kategoriler:

1. Kimlik Doğrulama ve Yetkilendirme Raporları
2. Sistem ve Veri Değişikliği Raporları
3. Ağ Etkinlik Raporları
4. Kaynak Erişim Raporları
5. Zararlı Yazılım Etkinlik Raporları
6. Kritik Hata ve Arıza Raporları

6.1 Kimlik Doğrulama ve Yetkilendirme Raporları

Bu raporlar, çeşitli kullanıcı ayrıcalık düzeylerinde (kimlik doğrulama) çeşitli sistemlere erişmek için yapılan başarılı ve başarısız girişimlerin yanı sıra ayrıcalıklı kullanıcı etkinliklerini ve ayrıcalıklı yetenekleri kullanma (yetkilendirme) girişimlerini tanımlar.

Kimlik doğrulama, günümüz sistemlerine erişimi kontrol altına almak için ana geçit ve araçtır. Basit şifrelerden şifreleme mekanizmalarına kadar kimlik doğrulama etkinliğini kurum genelinde gözden geçirmek önemli güvenlik etkinliklerinden biridir.

Bu kategorideki önemli raporlar şunlardır:

- Kullanıcı, sistem, iş birimine göre yapılan tüm başarılı ve başarısız girişler: Çeşitli sistemler, erişim yöntemleri (yerel, uzak) ve kullanıcılar üzerinden başarılı ve başarısız girişleri gösteren tek bir rapor veya birden fazla rapor olabilir. Bu rapor, yalnızca başarısız girişleri değil başarılı girişleri de loglamayı gerektirir.
- Devre dışı bırakılan, mevcut olmayan, varsayılan, askıya alınmış hesaplara ve hizmetlere giriş denemeleri (başarılar, başarısızlıklar): Bu rapor grubu, erişilmemesi gereken hesaplara ve hizmetlere erişim denemelerini kapsamaktadır. Hem başarısızlıklar hem de başarılar güvenlik uzmanlarının ilgisini çekmektedir.
- Çalışma saatleri dışındaki tüm oturumlar: Yukarıdaki rapora benzer şekilde, özellikle erişim denemesi başarılı olursa, bu tür etkinlikler çoğunlukla ilgi çekici olur. Bu gibi olayların sistem yöneticilerinin 7/24 çalıştığı ortamlarda araştırılması gerekir.
- Denediği sistemlerin sayısıyla birlikte kimliği doğrulanamayan kullanıcılar: Bu rapor, tek bir makinenin birçok sistem üzerindeki aynı veya farklı hesabı kontrol ettiği yerlerde hesap taramalarını tespit eder.
- VPN kimlik doğrulaması ve diğer uzaktan erişim girişleri (başarı, başarısızlık): Tüm giriş denemeleri doğru koşullar altında ilgi çekici olabilirken VPN veya diğer uzaktan bağlantı yöntemleri gibi uzaktan oturum açma denemelerine bir ilgi vardır ve dikkatle izlenmesi gerekir.

- Ayrıcalıklı hesap erişimi (başarı, başarısızlık): Ayrıcalıklı bir kullanıcı normal bir kullanıcıya göre çok daha fazla zarar verebildiğinden *root* veya yönetici girişleri, farklı çalıştır kullanımı ve diğer platformlar ve sistemler için ilgili eşdeğerlikler hesaba katılmalıdır.
- Aynı hesap tarafından birden fazla oturum açma başarısızlığını takip eden başarılı bağlantı: Bu raporun üretilmesi için kural tabanlı korelasyona (SIEM tarzı) ihtiyaç duyulurken, birden fazla başarısızlığın hemen ardından başarılı bir bağlantı gerçekleşmesi ilgi çekici olmaktadır. Çünkü böyle bir olay genellikle tahmin etme girişimlerini gösterir. (Chuvakin vd. 2013)

Tablo 6.1'de sistemlere giriş denemelerine ilişkin örnek bir rapor gösterilmektedir.

Tablo 6.1 Giriş Denemelerini Gösteren Örnek Rapor

Sistem	Hesap Adı	Kaynak IP	Durum	Metot	Sayı
Sistem1	root	60.15.183.46	Başarısız	SSH	17
Sistem2	user1	60.47.112.78	Başarılı	Local	5
Sistem3	admin	60.78.122.35	Başarısız	Local	3

6.2 Sistem ve Veri Değişikliği Raporları

Bu raporlar, çeşitli sistem ve güvenlikle ilgili kritik değişiklikleri tanımlamaktadır; yapılandırma dosyaları, hesaplar, düzenlenmiş ve hassas veriler ve sistem veya uygulamaların diğer bileşenleri.

Bilgi sistemlerinde yetkisiz değişiklikler çok maliyetli çökmelere, veri kayıplarına ve güvenlik olaylarına yol açar. Bunun da üstünde, saldırganlar sistemlerini gelecekteki erişimlerini sağlamak için sık sık değiştirir. Değişiklikleri izlemek için özenli davranmak, tüm bilgi teknolojileri operasyonunu da geliştirecektir.

Bu kategorideki önemli raporlar şunlardır:

- Kullanıcılara ve gruplara eklemeler/değişiklikler/silmeler: Saldırganlar sık sık yeni eylem ekler ve bazen erişildikten sonra bunları siler. Bu etkinlikler yetkili bir şekilde yapılmalıdır.

- Yönetici ve ayrıcalıklı grup hesaplarına eklemeler: Özellikle, yönetici hesaplarında ve diğer ayrıcalıklı kullanıcılarda yapılan değişiklikler, izlenen hesap değişiklikleri listesinin en başında olmalıdır.
- Kullanıcılar ve yöneticiler tarafından şifre değişiklikleri ve sıfırlamalar: Şifre değişiklikleri genellikle yeni hesap oluşturmalar kadar önemlidir. Bunlar hem kullanıcılar hem de yöneticiler tarafından gerçekleştirilebilir. Buna ek olarak bu rapor, yetkili şifre değişiklikleri politika programına göre gerçekleştirilirse emin olmak için kullanılabilir.
- Ağ hizmetlerine eklemeler/değişiklikler/silmeler: Ağ bağlantısına izin veren yeni hizmetler ağa ilave saldırılar başlatabilir. Saldırganlar tarafından bunlar sık sık gerçekleştirilirler.
- Sistem dosyalarındaki değişiklikler, yapılandırmalar: Sistem dosyalarında yapılan değişikliklerin dikkatli bir şekilde izlenmesi gerekir.
- Diğer önemli dosyalardaki değişiklikler: Çeşitli sistemlerde, ikili yürütülebilir dosyalara ve yapılandırma dosyalarına ilave olarak önemli dosyaların geniş listeleri olabilir. Bunlara da erişimin izlenmesi gerekir.
- Dosya erişim izinlerinde yapılan değişiklikler: Riskli değişikliğin sinsi bir çeşidi dosya izinlerinde yapılan değişikliklerdir. Hesaplanmazsa, bu tür değişiklikler hassas verilerin ele geçirilmesine neden olmaktadır.
- Hassas dosyalardaki değişiklikler: Bununla birlikte hassas belgelerin indirilmesi veya kopyalanması.
- Sistem, uygulama ve kullanıcı tarafından uygulama yüklemeleri ve güncellemeleri: Bütün uygulama yüklemeleri ve güncellemeleri, tüm sistemlerde loglanması gerekir. Olaya müdahalede bu loglar oldukça yararlı olacaktır. (Chuvakin vd. 2013)

Tablo 6.2'de bir Linux sistemindeki hesap ve grup eklemelerine ilişkin örnek bir rapor gösterilmektedir.

Tablo 6.2 Hesap ve Grup Eklemelerini Gösteren Örnek Rapor

Tarih	Sistem	Hesap Adı	Operasyon	Nesne	Durum
3/12/17 10:33AM GMT	sistem1	root	Hesap Eklendi	user1	Başarılı
3/15/17 14:12AM GMT	sistem2	root	Hesap Eklendi	root1	Başarısız
3/17/17 11:53AM GMT	sistem3	user1	Grup Eklendi	admin	Başarılı

6.3 Ağ Etkinlik Raporları

Bu raporlar, çeşitli şüpheli sistemleri ve potansiyel olarak tehlikeli ağ etkinliklerinin yanı sıra ortak düzenlemeler için izlenmesi gereken etkinlikleri tanımlar.

Ağ, tehditlerin bilgi varlıklarına ulaşmasının temel yoludur. Açıkçası ağ, günümüz kurumlarından bilgi varlıklarını çalmanın da temel yoludur.

Bu kategorideki önemli raporlar şunlardır:

- Sisteme, bağlantı sayısına, kullanıcıya, bant genişliğine, benzersiz hedef sayısına göre iç ve DMZ sistemlerinden giden tüm bağlantılar: Ortamdaki giden bağlantılarla ilgili bilgileri ayıklamanın birden çok yolu vardır, ancak temel esas aynıdır. İç ağdan dışarıya kimin bağlantı kurduğunu izlemek, saldırıları, tavizleri, zararlı yazılımları ve ağ erişimini kötüye kullanan kullanıcıları tespit etme yoludur.
- Çalışma saatleri dışındaki iç ve DMZ sistemlerinden giden tüm bağlantılar: Güvenlik duvarı ve web proxy logları kullanılarak, yukarıdaki raporun daha hedefe yönelik bir sürümünden faydalanılabilir ve yalnızca alışılmadık zamanlarda giden erişim izlenebilir.
- En büyük dosya aktarımları (gelen, giden) veya aktarılan baytlara göre en büyük oturumlar: İki rapordan herhangi biri, kurumların bariz veri hırsızlığı ve bant genişliğinin kötüye kullanımını izlemelerine olanak tanır.
- Harici sitelere yüklenen web dosyaları: Proxy loglarına dayalı olarak, hangi dosyaların harici sitelere yüklendiği veya webmail'e eklenmekte olduğu izlenebilir.
- İçerik türü ve protokole göre tüm dosya indirmeleri: Web'den hangi dosyaların ortama girdiğini izlemek de önemlidir, protokoller ve yöntemler arasında dosyaları izleme yoluyla yapılabilir.
- Birçok farklı protokolü/portu kullanan iç sistemler: İç sistemlerden gelen zararlı yazılım etkinliklerini her zaman bilmenin güvenilir bir yolu yokken, aniden yeni birçok port ve protokol üzerinden konuşulmaya başlanması kötü niyetli etkinliğin belirtisidir.

- Birden fazla NIDS, NIPS veya WAF alarmlarının kaynağı olan en yüksek iç sistemler: En kullanışlı raporlardan biridir. Birçok farklı türden oluşturulan iç bilgi varlıklarını izlemektir.
- Kullanıcı adına, toplam oturum baytına, oturum sayısına, iç kaynak kullanımına göre VPN ağ etkinliği: Yukarıdaki bölümde VPN girişlerini izlemek gerektiği vurgulandı, ancak VPN erişimi ve trafik anormalliklerini tespit etmek için VPN kullanımı da izlenmelidir.
- İç sistemler tarafından kullanılan P2P: P2P yazılımı, istenmeden kötü amaçlı veri hırsızlığına ve kayıplara neden olabilir.
- Kablosuz ağ etkinliği: Kablosuz ağ cihazları birçok farklı olay kaydedebilir ancak bunları VPN'ler ve diğer uzaktan erişim ağı mekanizmaları olarak ele almak ve erişimi izlemek yararlıdır.
- Gün boyunca üretilen log hacminin eğilimi: Ağ etkinliği raporunun tam olarak bir örneği değilken, ağda üretilen log hacminin gözden geçirilmesi, log verilerinin tüm havuzunda büyük resmi görmek için yararlıdır. (Chuvakin vd. 2013)

Tablo 6.3'te VPN hesap erişimlerine ve etkinliklerine ilişkin örnek bir rapor gösterilmektedir.

Tablo 6.3 VPN Hesap Erişimi ve Etkinliklerini Gösteren Örnek Rapor

Tarih	VPN	Kullanıcı Adı	Sistem	Eylem	Durum	Sayı
3/12/17	vpnx	user1	user1desktop	Giriş	Başarısız	5
3/15/17	vpny	user2	user2desktop	Giriş	Başarılı	3
3/17/17	vpnz	root	desktop1	Giriş	Başarısız	17

6.4 Kaynak Erişim Raporları

Bu raporlar, kurumdaki çeşitli sistem, uygulama ve veritabanı kaynak erişim desenlerini tanımlar ve etkinlik denetimi, eğilimi ve olay tespiti için kullanılabilir.

Kaynak erişiminin izlenmesi içeriden kötüye kullanımın ve dolandırıcılıkların ortaya çıkarılmasında kullanılabilir. Saldırganın hangi kaynaklara eriştiğini, bozduğunu veya değiştirdiğini belirlemek için olayla ilgili yanıt sırasında değerlidirler. Buna ilave olarak kaynak erişimi, kapasite planlaması gibi güvenlik dışındaki amaçlar için de kullanılabilir.

Bu kategorideki önemli raporlar şunlardır:

- Çalışma saatlerinin dışında kritik sistemlerdeki kaynaklara erişim: Alışılmadık zamanlarda kritik sistem üzerindeki erişimi ve etkinlikleri izlemek için kullanılabilir.
- Proxy tarafından yasaklanmış sitelere, zararlı yazılım kaynaklarına vb. erişimi engellenen en yüksek iç kullanıcılar: Bu çok yönlü web erişimi raporu, taviz verilmiş sistemlerin izlenmesinden veri sızıntısının izlenmesine ve üretkenliğin arttırılmasına kadar pek çok amaç için kullanılabilir.
- Dosya, ağ paylaşımı veya kaynak erişimi: Bu rapor yalnızca denetlenen belirli kaynaklar için çalıştırıldığında yararlı olabilir. Dosya erişimi (Windows) ve sistem çağrıları (Unix) loglamayı etkinleştirmek her zaman veri fazlalığına yol açar.
- En yüksek veritabanı kullanıcıları: Güvenlik etkinliğini izlemek için yararlıdır ancak veritabanına bilinen uygulama erişiminin hariç tutulması gerekir. İdeal olarak, kullanıcıların veya geliştiricilerin bir üretim (production) veritabanına doğrudan erişimi olmaması gerekir.
- Sorgu türlerinin özeti: Bilinen uygulama sorguları hariç tutulduğunda bu rapor, anormal veritabanı erişimini gösteren anomali tespit aracına dönüşür.
- Tüm ayrıcalıklı veritabanı kullanıcı erişimi: Sunucularda ve uygulamalarda olduğu gibi, tüm ayrıcalıklı kullanıcı etkinlikleri kaydedilmeli ve düzenli olarak analiz edilmelidir.
- *Insert* ve *Delete* veritabanı komutlarını çalıştıran tüm kullanıcılar: Uygulama ve kullanıcı erişimini izlemenin yanı sıra, verileri imha edebilecek zararlı komutları ayrı ayrı izlemek mantıklıdır. Bilinen uygulama sorguları hariç tutulduğunda yararlı bir pratiktir.
- Veritabanında *Create, Grant* ve şema değişiklikleri komutlarını çalıştıran tüm kullanıcılar: Uygulama ve kullanıcı erişimini izlemenin yanı sıra, verileri imha edebilecek ve veritabanı örneğinin kendisini değiştirebilecek zararlı komutları ayrı ayrı izlemek mantıklıdır.

- Veritabanı yedeklemelerinin özeti: Yedekler, bir veritabanından büyük miktarda veri çekmek ve böylece veri hırsızlığı yapmak için kolay bir yol sunar. Bu rapor eski veritabanı yedeklerini incelemeye ve yetkisiz yedek alanları yakalamaya olanak tanır.
- Ekleri dışarıya gönderen en yüksek iç e-posta adresleri: Birçok e-posta erişim raporunda bu göze çarpar. İçerdekilerin kötüye kullanımını ve veri hırsızlığını tespit etmek için yararlı bir araçtır.
- E-posta ile gönderilen tüm ek içerik türleri, boyutları ve adları: Yukarıdaki rapora benzer şekilde, bilgi sızıntısını izlemek ve aynı zamanda hassas bilgilerle e-posta gönderen kullanıcıları tespit etmek için kullanılabilir.
- Bilinen e-posta sunucuları hariç e-posta gönderen tüm iç sistemler: Ortamdaki spam gönderen botlar ile spam bulaşmış sistemleri bulmanın temel yoludur.
- Log erişim özeti: Loglama ve daha sonra loglara erişimi gözden geçirme, düzenlemelerle belirlenir. Bu temel rapor görüntülenen log verilerini hariç tutmaya izin vermelidir. (Chuvakin vd. 2013)

Tablo 6.4'te birden çok sunucudaki tüm dosyalara erişim hakkında örnek bir rapor gösterilmektedir.

Tablo 6.4 Birden Çok Sunucuya Dosya Erişimini Gösteren Örnek Rapor

Tarih	Sunucu	Kullanıcı Adı	Dosya Adı	Erişim Türü	Durum	Sayı
3/12/17	Windows1	user1	abc.docx	Yazma	Başarısız	7
3/15/17	Windows2	user1	qwe.xlsx	Okuma	Başarılı	1
3/17/17	Unix1	user1	xyz.pdf	Okuma	Başarısız	19

6.5 Zararlı Yazılım Etkinlik Raporları

Bu raporlar, çeşitli zararlı yazılım etkinliklerini ve zararlı yazılımla ilgili olayları özetler.

Zararlı yazılımlar günümüz kurumları için önemli tehditlerden biri olmaya devam etmektedir. Son birkaç yıldır zararlı yazılımları durdurma konusunda antivirüs araçlarının verimliliğinin düştüğü göz önüne alındığında, zararlı yazılımlarla mücadele için loglar gibi diğer bilgi kaynakları da kullanılmalıdır.

Bu kategorideki önemli raporlar şunlardır:

- Sonuçlarla birlikte zararlı yazılım algılama eğilimleri: Zararlı yazılımı tespit etmeye yönelik bir eğilim veya özet içeren temel bir raporda sistemin ve sonucun da gösterilmesi iyi bir başlangıç noktasıdır.
- Antivirüs araçlarından algılanan olaylar: Tüm antivirüs araçları zararlı yazılım tespit edildi ancak temizlenemedi gibi durumları loglar. Bunlar çok sayıda kurumun büyük hasarlardan kaçınmasına yardımcı olur.
- Tüm antivirüs koruması arızaları: Günümüzdeki zararlı yazılımlar, antivirüs araçlarıyla mücadele için iyi donanıma sahip olmaları nedeniyle tüm çökmeler, güncelleme arızaları vs. loglanmalı ve gözden geçirilmelidir.
- Bilinen zararlı yazılım IP adreslerine yapılan iç bağlantılar: Bu rapor, logları ve IP adresinin genel bir kara listesini kullanarak oluşturulabilir. Böyle basit bir yaklaşım, kurumdaki zararlı yazılım kullananlara ait değerli verilerin kaybedilmesine engel olabilir.
- En az yaygın olan zararlı yazılım türleri: Alışılmadık bir duruma yararlı bir bakış açısı sunar, böylece kurumdaki kötü amaçlı yazılımlara zarar verebilir. (Chuvakin vd. 2013)

Tablo 6.5'te bir ağdaki bir haftalık log verilerinin virüs türlerini gösteren örnek bir rapor verilmektedir.

Tablo 6.5 Bir Ağdaki Virüs Türlerini Gösteren Örnek Rapor

Zararlı türü	Durum	Etkilenen Sistem Sayısı
TrojanA	Tespit Edildi	2
VirüsB	Karantinaya Alındı	1
VirüsC	Tespit Edildi	3

6.6 Kritik Hata ve Arıza Raporları

Bu raporlar, güvenlik açısından önem arz eden çeşitli önemli hataları ve arıza belirtilerini özetler.

Hata ve arıza log mesajları genellikle IDS ve IPS sistemleri gibi güvenlik özellikli cihazlar tarafından yakalanmayan gelişmiş tehditler de dâhil olmak üzere

güvenlik tehditlerinin erken belirtileridir. Ağda zararlı yeni bir tehdit unsuru belirdiğinde alışılmadık hata mesajlarına dikkat edilmesi gerekir.

Bu kategorideki önemli raporlar şunlardır:

- Sistem, uygulama ve iş birimine göre kritik hatalar: Güvenlik açısından önemsiz olmakla birlikte, loglarda ortaya çıkan çeşitli hata mesajları, genellikle kötü niyetli etkinliklerin çok erken bir belirtisini gösterdikleri için araştırılmalıdır.
- Sistem ve uygulamanın çökmesi, kapanması ve yeniden başlaması: Başarısız saldırılar veya diğer nedenlerden dolayı uygulamalar çöktüğünde kurumun işleyişi muhtemelen etkilenir. Bu olaylar sadece ulaşılabilirlik üzerinde etkili değil, aynı zamanda muhtemel erken dolaylı saldırı belirtileri olarak araştırılmalıdır.
- Yedek arızaları: İş sürekliliğini ve mevzuata uygunluğu etkileyen kritik olaylardır. Buna ilave olarak yetkisiz yedeklemeler, saldırganın veri çalmaya teşebbüsüyle tetiklenebilir.
- Bellek, disk, CPU ve diğer sistem kaynakları için kapasite tükenme olayları: Genellikle saldırgandan veya iş sistemlerinin yetkisiz kullanımından kaynaklanır. Yüksek kaynak kullanımı, DoS veya kaba kuvvet saldırıları nedeniyle de olabilir. (Chuvakin vd. 2013)

Tablo 6.6'da Unix/Linux sunucularının bir havuzunda dolu disk ve yüksek CPU kullanımı mesajları hakkında örnek bir rapor gösterilmektedir.

Tablo 6.6 Dolu Disk ve Yüksek CPU Kullanımını Gösteren Örnek Rapor

Sunucu	Olay Türü	Tarih
Server1	Dolu Disk	3/12/17
Server2	%100 CPU Yükü	3/15/17
Server3	Dolu Disk	3/17/17

7. LOGLAMA KURALLARI VE HATALARI

Bilgi sistemlerinden gelen loglar yıllardır toplanmakta ve analiz edilmektedir. Burada bazı genel gerçekler ortaya çıkmaktadır. Bu bölümde, loglar hakkında ortaya çıkan evrensel gerçeklerden bahsedilecek ve bunlar loglama kuralları olarak nitelendirilen bazı varsayım derecelerine göre sınıflandırılacaktır.

7.1 Loglama Kuralları

Buradaki loglama kuralları, toplama, analiz ve karar vermeye kadar log verileri ile ilgili tüm yelpazeyi kapsamaktadır. Loglama ve IDS konularında belirtilmiş olan kurallar sırasıyla şunlardır:

1. Asla kullanmayı düşündüğünüzden fazla veri toplamayın.
2. İlginç bir şeyin kaç kez gerçekleştiği de ilginç bir şeydir.
3. İlk kural ile ihtilafa düştüğünüz yerler dışında her şeyi toplayın.
4. Gerçek zamanlı sistem yöneticiniz yoksa IDS'inizin ne kadar gerçek zamanlı olduğu önemli değildir. (Ranum, 2004)

Bu kurallar günümüzde hâlâ geçerliliğini korumaktadır; ancak biraz daha genişletilmesi gerekir. (Chuvakin vd. 2013)

7.1.1 Toplama Kuralı

Kural, yukarıdaki ilk maddeyle ilişkilidir: “Kullanmayı planlamadığınız log verilerini asla toplamayın.” Bu kural, basitçe loglanan ve saklanan her mesaj için bir nedenin bulunduğunu belirtmektedir. Bu kural, yalnızca toplama için değil, log verisinin oluşturulması için de geçerlidir: “Kullanmayı planlamadığınız şeyi asla loglamayın.”

7.1.2 Tutma Kuralı

Günümüzde çok fazla veri toplanabilirken bunlar nadiren kullanılmaktadır: “Log verilerini kullanılabileceği düşünüldüğü ya da düzenlemelerle öngörüldüğü sürece tutun.”

Tutma rutiniyle log mesajı silindikten hemen sonra ihtiyaç duyulduğu durumlar düşünüldüğü kadar az değildir. Bu nedenle verilerin yararlı olduğu sürece saklanması gerekir. Öte yandan, uygulama çökmesi sonucunda oluşan bir hata ayıklama mesajının yıllarca kullanılabileceği de ihtimal dâhilinde değildir.

7.1.3 İzleme Kuralı

İlk iki kural, daha fazla loglamanın ve daha uzun süre log tutmanın akıllıca olmayacağından bahsederken izleme kuralı ile ters düşmektedirler: "Olabildiğince herşeyi loglayın; ancak sadece müdahale edilmesi gereken hususlara dikkat edin."

Bu kural evrensel gerçek olarak, sistemin petabaytlık veriyi, trilyonlarca log mesajını kolayca depolayabildiği zaman hayata geçirilebilir; ancak, güvenlik uzmanı günde bir düzine sorunu anca ele alabilir. Bu da loglama ile izleme arasındaki büyük farkı ortaya koyar.

Kurumun takip etmesi gereken kural "Her şeyi loglayın, bazılarını saklayın, ihtiyaç duyulan şeyleri izleyin" olmalıdır.

7.1.4 Erişebilirlik Kuralı

Güvenlik izleme sisteminin iş sistemleri kadar erişilebilir olması gerektiği mantıksaldır ama daha fazlası için değil: "Loglama veya izleme sisteminizi, iş sistemlerinizden daha fazla erişilebilir hale getirmek için para ödemeyin."

Log verilerinin toplanıp analiz edildiğinden emin olmak önemlidir ve gelecekteki araştırmalar ve hatta mahkeme davaları için kullanılabilir. Bununla birlikte iş dünyasının liderleri, iş sistemlerinin erişilebilir olmasının daha da önemli olduğunu söyleyebilir.

7.1.5 Güvenlik Kuralı

Erişilebilirlik kuralına benzer şekildedir: "Log verilerinizi korumak için kritik iş verilerinizi korumaya ödediğinizden daha fazla para ödemeyin."

Loglar gerçekten değerlidir ancak çoğu kurum, loglardan daha değerli olacak başka bir veri setine sahiptir. Bu nedenle az sayıda kurum aslında log verilerini şifrelemeyi tercih etmektedir. Şifrelemenin ortaya çıkardığı operasyonel zorlukların yanı sıra birçok durumda bu kontroller gereksiz olacaktır. Bazıları, logun kurcalanmasını azaltmak için log arşivlerini özet (hash) fonksiyonuna tabi tutar. Diğerleri, sıkı erişim kontrollerine güvenmeyi tercih eder.

7.1.6 Sabitlerdeki Değişiklik Kuralı

Loglama, log yönetimi ve log analizi alanındaki sabit bir şey değişir: "Log kaynakları, log türleri ve log mesajları değişir."

Kurumların log inceleme politikalarını, prosedürlerini, operasyonel görevlerini ve sistem yapılandırmalarını sürekli olarak gözden geçirmesi gerekir. Log toplama işlemindeki her bir adımı dokümante etmek ve bu dokümanı güncel tutmak, logun üretildiği noktadan saklandığı yere kadar aradaki tüm noktaları göstermek yararlıdır.

7.2 Loglama Hataları

Kurallarla ilgisi olmayan ancak bazıları daha uzun yıllar önce ortaya çıkmış, günümüzde halen geçerli olan birçok hata vardır. Bunlardan bazıları daha modern ve günümüz bilgi teknolojisi ortamlarına özgüdür.

Bu kısım, kurumların bilgi sistemleri loglarına ve bilgisayarlar ile ağ cihazları tarafından üretilen diğer kayıtlara yaklaşırken yaptıkları tipik hataları kapsamaktadır. Bu hatalar, bilincin artmasına rağmen birçok kurumda ne yazık ki yaygın olarak yapılmaktadır.

Teknolojinin yaygınlaşması ve bilgisayarların iş hayatında ve kişisel hayatta daha önemli rol oynamaya başlamasıyla birlikte logların rolü de artmaktadır. Böylece, bu hataları yapmanın maliyeti de artmaya devam etmektedir. Uzun yıllar önce buradaki listeden bir hata yapılsaydı, yalnızca sorun giderme veya diğer sistemlerle ilgili işlemler engellenmiş olurdu. Günümüzde ise yapılan bu hatalar çok daha büyük sonuçlara neden olmaktadır.

Güvenlik duvarları ve saldırı önleme sistemlerinden (IPS) veritabanları ve kurumsal uygulamalara, kablosuz erişim noktalarından sanallaştırma platformlarına kadar, loglar giderek artan bir hızla dağılmaktadır. Bu tür log kaynaklarının mevcut olabileceği yerler, sistemlerden bulut uygulamalarına ve oradan da mobil cihazlara kadar genişlemektedir. Hem güvenlik hem de diğer BT bileşenlerinin sayıları artmakla kalmayıp, çoğu zaman daha fazla loglama özelliğiyle birlikte gelir. Bu eğilime örnek olarak günümüzde Linux sistemleri ve web sunucuları artan loglama seviyesi ile birlikte gelmektedir. Windows sistemleri de yıllar öncesine göre günümüzde çok daha fazla olay kaydetmekte ve her olay içinde de daha fazla ayrıntı yer almaktadır.

Hem eski hem de yeni, geleneksel ve bulut tabanlı bütün sistemlerin sürekli olarak dikkat çekmek için yoğun miktarda log, denetim izi, kayıt ve uyarı ürettiği bilinmektedir. Bu nedenle, birçok şirket ve devlet kurumları, log toplama, merkezileştirme, analiz süreçleri ve araçları kurmaya çalışmaktadır.

Bulut bilişimin ön planında yer alan şirketler, loglarda boğulanlardır. Facebook ve Google'dan diğer web şirketlerine kadar loglar artık gigabayt değil, terabayt ve hatta petabayt cinsinden ölçülmektedir. Loglanan verilerin (güvenlik, hata ayıklama ve operasyon) birleşik hacminin terabayt olarak ölçüldüğü kurumlar giderek daha yaygın hale gelmektedir.

Log toplama ve analiz altyapısını planlarken ve uygularken, kurumlar genellikle böyle bir sistemin vaat ettiğini tam anlamadıklarını keşfederler ve bazen verimliliğin kazanılmadığını sonuç olarak kaybolduğunu fark ederler.

Burada bahsedilecek olan hatalar, PCI DSS gibi birçok düzenlemelerin olduğu bu çağda bile yaygın olarak yapılmaktadır. Bu hatalar, loglardan yararlanma olasılığını da oldukça azaltmaktadır. (Chuvakin vd. 2013)

7.2.1 Her Şeyi Loglamama

Yapılan ilk hata basitçe, her şeyi loglamamakdır. Bu hatadan daha kötüsü ise, loglamamak ve çok geç olmadan onu öğrenmemektir. Gelişmiş kurumlar bile bu hataya düşmektedirler. Örneğin; web sunucusunun loglama özelliğinin etkin olup olmadığı bilinmelidir; Apache ve Microsoft IIS gibi popüler web sunucularında bu varsayılan bir seçenektir. Sunucunun işletim sistemi log mesajlarının oluşup oluşmadığına göz atılmalıdır; bunlar varsayılan olarak */var/log/messages* gibi bir dizine kaydedilmektedir. Ayrıca veritabanı loglarının bulunup bulunmadığına bakılmalıdır; MS SQL ve Oracle'da varsayılan seçenek, herhangi bir veri erişim denetimi loglaması yapılmamasıdır. Veritabanında orta seviyede bir denetim izi oluşturma sürecini başlatmak için sistemin derinliklerine inmek gerekir.

Bu hatayı önlemek için yerleştirilen yazılımın ve donanımın belirli loglama seviyesinin etkinleştirildiğinden emin olunmalıdır. Loglama ayarlarına sahip olmayan sistemlerle çalışılması durumunda bu tür sistem kullanımını bırakmak veya üreticiyle görüşmek ya da etkinlikleri kaydetmek için ilave teknolojiler eklemek gerekir.

7.2.2 Log Verilerine Bakmama

İkinci hata loglara bakmamaktır. Logların var olduğundan ve daha sonra bunların toplanmasından ve saklanmasından emin olmakla birlikte, ortamda ne olup bittiğinin bilinmesi, bunlara müdahale edilmesi ve sonrasında neler olacağının önceden tahmin edilmesi gerekir. Dolayısıyla teknoloji yerleştirildikten ve loglar toplandıktan sonra, gerekirse eyleme giden bir izleme ve gözden geçirme süreci olması gerekir. Buna ilave olarak, logları gözden geçiren veya izleyen personel, logların ne anlama geldiğini ve herhangi bir işlem gerektiğinde ne yapılması gerektiğini belirleyebilmek için yeterli bilgiye sahip olmalıdır.

Bazı kurumların doğru yönde fakat yarım adım attığını da ele almak gerekir. Bu kurumlar, sadece önemli bir olaydan sonra logları inceler ve sürekli izleme ve log incelemesinden kaçınırlar. Bunun nedeni de çoğunlukla kaynak yetersizliği olarak aktarılır. Bu onlara log analizin reaktif faydasını sağlar; ancak önemli olan proaktif olanı (kötü şeylerin yaşanacağını bilmek gibi) gerçekleştirmektir. Aslında loglara proaktif bir şekilde bakmak, kurumların mevcut ağ, güvenlik ve sistem altyapılarının değerini daha iyi anlamasına yardımcı olur.

Bazı kurumların, belirli bir düzenleyici baskı nedeniyle log dosyalarına ve denetim izlerine bakması gerekir. Örneğin; HIPAA yönetmeliği, sağlık kurumlarını denetim kaydı ve analiz programı oluşturmaya zorlar. PCI DSS ise hem log toplama hem de log izleme ve periyodik gözden geçirme için hükümler içerir, toplanan logların tek başına durmaması gerçeğini vurgular.

Risk değerlendirmesine ve erişim yeteneklerine göre gözden geçirme önceliklerinin sıralandığı bir liste aşağıda verilmektedir. Böyle bir yaklaşım, kurumun loglara nasıl bakabileceği konusunda fikir sahibi olmasını sağlar.

1. DMZ NIDS
2. DMZ güvenlik duvarı
3. DMZ sunucuları ile uygulamaları
4. Kritik iç sunucular
5. Diğer sunucular
6. Seçilen kritik uygulamalar
7. Diğer uygulamalar

7.2.3 Çok Kısa Süreli Saklama

Üçüncü yaygın hata, logları çok kısa bir süre saklamaktır. Bu, güvenlik veya BT operasyon ekibinin, izleme, soruşturma veya sorun giderme için gerekli tüm loglara sahip olduklarını düşünmeleri ve olaydan sonra kısa görüşlü tutma politikalarından dolayı tüm logların kaybolduğunu fark etmeleri sonucuna neden olur. Bu, genellikle suç ya da suiistimalin işlenmesinden uzun bir süre sonra olay keşfedildiğinde ortaya çıkan bir durumdur. Depolama donanımından bir miktar tasarruf etmek için yapılan bu hata, düzenleyici para cezaları yüzünden tasarruf edilmesi planlanandan çok daha fazlasını kaybettirebilir.

Düşük maliyet önemliyse çözüm, saklama düzeyini iki kısma ayırmaktır: Kısa vadeli saklama için çevrimiçi depolama alanı (daha maliyetli) ve uzun vadeli saklama için çevrimdışı depolama alanı (daha ucuz). Daha iyi bir üç katmanlı yaklaşım da yaygındır ve bir öncekindeki sınırlamanın bazılarını giderir. Bu durumda, kısa süreli çevrimiçi depolama birimleri, logların hâlâ erişilebilir ve aranabilir olduğu yakın zamanlı depolama birimiyle tamamlanmaktadır. En eski ve en az ilgili log kayıtları, ucuza depolanabildiği kaset veya DVD gibi üçüncü bir katmana yüklenir; ancak burada gerekli loglara seçerek erişmek için herhangi bir yol yoktur.

Logları çok uzun süre saklamak da bir hatadır. Belli bir süre sonra loglar imha edilmelidir. Bunun hata olmasının üç ana faktörü vardır:

1. Avrupa'da yaygın olan gizlilik düzenlemeleri, genellikle hem veri toplama hem de tutma oranını sınırlar. Bazı ülkelerde log saklamak yasadışıdır ve bu nedenle loglama politikası planlandığında bunun da dikkate alınması gerekir.
2. Dava yönetimi riski. Hukuki ekip, bir dava durumunda yasal olarak keşfedilen çok fazla bilgiyi toplamanın iyi bir şey olmadığını hatırlatacaktır. En bilinen örnek, bazı etkinliklerin loglarına sahip olmanın hak ihlali olarak görülmesidir. Böyle şeyler kurumu yasal olarak tehlikeye atabilir.
3. Sistem kaynağı kullanımı sıklıkla depolanan log verilerini sınırlar. Güvenlik duvarları işlekse birkaç yıl logları tutmak imkânsız veya masraflı olabilir.

Log saklama stratejisi oluşturmak ve hatalardan kaçınmak için örnek bir saklama stratejisi şu şekildedir:

1. Log kaynağı türü: Güvenlik duvarı, IDS, sunucu, masaüstü vb.
2. Ağın konumu: DMZ, bölge, il müdürlüğü vb.
3. Depolama katmanı: Çevrimiçi (bir log yönetim sistemi), yakın zamanlı (ham log için büyük sabit disk depolaması), çevrimdışı (kaset veya CD).

Bu, bir tür log kaynağı (güvenlik duvarı gibi) seçmek, ağda nereye yerleştirildiğini (DMZ gibi) düşünmek ve ardından nasıl depolanacağını (hızlı çevrimiçi depolama birimi gibi) planlamak anlamına gelir. Örneğin; DMZ IDS logları üç ay boyunca çevrimiçi olarak depolanırken, aynı konumdaki güvenlik duvarı logları yüksek hacminden dolayı bir ay boyunca depolanır. Aynı loglar daha sonra çevrimdışı depoda üç yıl süreyle saklanır.

Toplanan cihazların sırası da önemlidir; çünkü bu, log yakalama ve tutma uygulamasının önceliğini tanımlayacaktır.

7.2.4 Toplamadan Önce Önceliklendirme

Dördüncü hata, log verisinin önceliği ile ilgilidir. Log analiz çabalarını daha iyi organize etmek için önceliğe ihtiyaç duyulurken, günümüzde yaygın olarak görülen hata, toplanmadan öncesi logların önceliklendirilmesidir. Aslında, bazı en iyi uygulama dokümanları bile yalnızca önemli şeyleri toplamayı önerirler. Faydalı bir biçimi olmamasından dolayı kılavuz dokümanları en önemli şeylerin ne olduğu konusunda kısa kalmaktadır. Güvenlik tutumunda haberdar olunan her şey, göze batan boşluklara yol açabilir veya mevzuata uyum çabalarını sarsabilir. Bir log türünün diğer log türünden daha önemli olduğu gibi iddialar tartışılabilir ve bu da her şeyi toplamak gerektiğine dair bir kavrayışa neden olur.

Logu görmeden önce hangi logun daha önemli olduğu konusunda yapılan seçimin doğru olup olmadığının cevabını vermeye çalışmak, bu sorunu çözümlenemez hale getirir. Her şeyi kaydetmek, bir analistin her bir log kaydını gözden geçireceği anlamına gelmez. Bu, ihtiyaç duyulduğunda tümü olmasa da çoğu log kaydının mevcut olduğu anlamına gelir.

Bu hatadan kaçınmak için şu basitleştirme adımları kullanılabilir:

1. Her şeyi loglayın
2. Her şeyin çoğunu saklayın
3. Yeterli analiz yapın
4. Bir altküme üzerinde özetleyin ve raporlayın
5. Bazılarını izleyin
6. Eylem gerektiren az kayıt üzerinde hareket edin

Basitçe söylemek gerekirse, öncesinde değil de loglar toplandıktan ve saklandıktan sonra bilgi indirgemesi gerçekleşir. Bu, logları mevzuat veya güvenlikle ilgili olarak gelecekteki herhangi bir durum için kullanmaya olanak tanır.

7.2.5 Uygulama Loglarını Yok Sayma

Beşinci hata, uygulamaların loglarını yok saymaktır. Sadece ağ cihazlarına ve sunuculara odaklanarak, log yığınının üst kısmındaki uygulama loglarını hesaba katmamaktır.

Microsoft, SAP, IBM gibi büyük şirketlerin uygulamalarından, küçük ölçekte kurum içi geliştirilenlere kadar uzanan kurumsal uygulamalar alanı, birçok kurum için kritik süreçleri ele alır. Logların erişilebilirliği ve kalitesi, eksik olandan son derece detaylı olanlara kadar uygulama genelinde farklılık göstermektedir. Ortak loglama standartlarının ve yazılım geliştiricileri için loglama kılavuzunun eksikliği, uygulama logları ile birçok zorluğa neden olur.

Uygulamalar buluta yerleştirildiğinde veya kurum servis olarak yazılım (SaaS) bulut modelinden yararlandığında etkinlikleri ve saldırıları izlemek için tek yol uygulama logları olmaktadır. Bu nedenle uygulama loglarının öneminin arttığı gerçeği göz önünde bulundurularak bunların er ya da geç analize dâhil edilmesine ihtiyaç duyulacaktır.

7.2.6 Sadece Bilinen Kötü Girdileri İzleme

Altıncı hata, loglarda sadece kötü olduğu bilinen girdileri incelemektir. En gelişmiş kurumlar bile bu hataya düşmektedir. Bu hata bir log analizi projesinin değerini önemli ölçüde düşürebilir. Açık kaynağın büyük bir çoğunluğu ve bazı ticari araçlar, kötü log satırlarını, saldırı imzalarını ve kritik olayları filtrelemek ve aramak üzerine oluşturulmuştur.

Bununla birlikte, log verilerinin değerini tam olarak anlamak için, onu bir sonraki seviyeye, log madenciliğine taşımak gerekir. Log madenciliği, bulunması gerekenler konusunda peşin hüküm vermeden log dosyalarında ilgilenilen şeyleri keşfetmektir. Olası tüm kötü niyetli davranışların önceden bilinmesi zordur. Bazen, sadece bilinen tüm iyi şeyleri listelemek ve ardından geri kalanı bulmak daha kolaydır. Bir çözüm gibi görünse de bu tür bir görev zahmetli olmakla kalmayıp aynı zamanda da boşunadır. Çünkü bir sistem veya ağda meydana gelebilecek tüm iyi şeyleri listelemek, tüm kötü şeyleri listelemekten çok daha zordur.

Sadece tüm olanakların listelenmesiyle saldırı izlerini ayıklamak verimli değildir. Daha akıllı bir yaklaşıma ihtiyaç vardır. Bazı veri madenciliği ve görselleştirme yöntemleri, log verileri üzerinde büyük bir başarı ile çalışır. Bunlar, kurumların kötü ve iyi bilinenin ötesinde log verilerinde gerçek anormallikleri aramasına izin vermektedirler.

8. GELİŞTİRİCİLERE YÖNELİK LOGLAMA

Genellikle geliştiriciler log mesajlarında ya yeterli bilgi sağlamazlar ya da çok fazla bilgi sağlarlar. Buradaki püf noktası, güvenlik ve performans konuları ile loglarda yeterli bilgiyi sağlamanın nasıl dengede olacağını anlamaktır. Ayrıca uygulama tarafından üretilen log mesajları, log toplama ve analiz araçları ile toplanabilir. Bunun anlamı, logların ayrıştırma becerisinin de göz önünde bulunması gerektiğidir. Bu bölümde sunulan kavramlar herhangi bir dile ve ortama uygulanabilir.

8.1 Roller ve Sorumluluklar

Geliştiriciler için loglama teknikleriyle ilgili ayrıntılara girmeden önce, yazılım geliştirme ve programlamayla ilişkili olarak ortak rolleri ve sorumlulukları anlamak önemlidir. Bu kısımda bahsedilen roller en yaygın olanlardır; kurumun gerçek rolleri farklı olabilir. Tablo 8.1'de yaygın olan roller ve sorumluluklar gösterilmektedir.

Tablo 8.1 Roller ve Sorumluluklar

Rol	Sorumluluk
İş/Ürün Sahibi	Bir ürün, özellik seti, uygulama vb. için sorumlu olan kişi veya grup.
Güvenlik Ekibi	İçte geliştirilen yazılımlar için loglama standardı geliştirmek; iş sahiplerini sürece dâhil etmek.
BT/Geliştirme Yöneticileri	Loglama standardını olması gereken özellik olarak tatbik etmek.
BT Destek Ekibi	Üretim (production) ortamındaki uygulamalara destek vermek; uygulamalarda hata olduğunda log dosyalarını incelemek.
Yazılım Mimarları	Loglama değerini almak ve denetim ile hata ayıklama loglamasını ayırmak.
Yazılım Geliştiricileri	Kütüphaneleri, API'leri ve loglama özelliklerini kullanarak loglama standartlarını takip etmek.

(Chuvakin, Application Logging: Worst Practices)

Görevlerin ayrılması, üretim ortamına koruma sağlamak için ihtiyaç duyulan bir güvenlik ilkesidir. Bununla birlikte, bazı durumlarda BT destek personeli olan yazılım geliştiricisinin aynı zamanda yazılım mimarı olduğunu da belirtmek gerekir. Bu genellikle daha küçük kurumlarda olur; ideal bir durum değil ama gerçekleşmektedir.

Görevlerin ayrılmasının kritik bir başka nedeni de mevzuata uygunluğu sağlamaktır. Çeşitli düzenleyici kurumlar, yazılım geliştiricilerin üretim sistemlerinde yazma erişimine sahip olamayacağını belirtmektedir. Bazı durumlarda geliştiriciler üretim sistemlerine erişemezler. Bir geliştiricinin, üretim ortamındaki bir sorunun giderilmesine yardımcı olmasının birkaç yolu vardır:

1. Geliştirici, üretim destek personeli ile birlikte çalışır ve operatöre rehberlik eder.
2. Ekran kontrolleri, geliştiriciye üretim sisteminin geçici görünümünün ve denetiminin verildiği yerde destek personelinin terminalinin bakış açısıyla kullanılabilir.
3. Bir güvenlik istisnası verilebilir. Düzgün belgelenirse, genellikle buna izin verilir. Ana fikir, geliştiriciye kısa süreliğine (30 dakika gibi) geçici olarak sisteme erişim izni verilmesidir.

8.2 Geliştiriciler İçin Loglama

Geliştiricilerin loglama yaparken neden dikkatli olmaları gerektiği konusunda ortak nedenler şunlardır:

- Geliştiriciler, log mesajlarının esas kaynağıdır. Uygulamaların değerli ve faydalı olan iyi loglar oluşturduğundan emin olunması gerekir. İyi oluşturulmuş ve zamanında hazırlanan log mesajları, hata ayıklamaya, daha fazla teşhis toplamaya ve geliştiricinin uygulamanın gerçekte ne yaptığını anlamasına yardımcı olabilir.
- Bir log mesajında doğru bilgi sağlandığında, kötü amaçlı davranış tespitinde yardımcı olabilir.

Bunlar, loglama konusunda zorlayıcı nedenlerden sadece birkaçıdır. Geliştirici açısından loglamanın arkasındaki sürecin üst seviye adımları şu şekildedir:

1. Geliştirici uygulamayı yazar, koda hangi log mesajlarının ekleneceğine karar verir.
2. BT destek ekibiyle neyin loglanacağının listesi gözden geçirilir. BT destek personelini dâhil etmenin birinci nedeni girdilerini almaktır. İkinci bir nedeni, ihtiyaçlarını izleme bakış açısıyla anlayabilmektir.
3. Geliştirici, loglama eylemini kolaylaştıran fonksiyon veya metot çağrıları yapan kodu yazar.
    a. Genellikle, logların nereye yazıldıklarını gösteren bir yapılandırma dosyası bulunur. Bu, yerel disk, uzak disk, syslog vb. içerir.
    b. Log rotasyonu şemaları da dikkate alınır.
4. Loglar daha sonra hata ayıklama, analiz vb. için kullanılır.

8.2.1 Neler Loglanmalı

Logları yararlı kılan şeyleri anlamak iyi olacaktır. Bir log mesajı, olanları ve neden olduğunu anlatmalıdır:

- Ne oldu
- Nerede oldu
- Ne zaman oldu
- Neden oldu
- Nasıl oldu
- Kim katıldı

Özel ihtiyaçlar farklı olabilir; ancak temel log türlerini bilmek gerekir. Loglanması gereken log mesaj türleri için temel yönergeler aşağıda verilmektedir:

- **Kimlik Doğrulama, Yetkilendirme, Erişim:** Başarılı ve başarısız doğrulama veya yetkilendirme kararları; sistem erişimi, veri erişimi ve uygulama bileşeni erişimi; dağıtık bir ortamda bir uygulama bileşeni arasında başka bir uygulamayı da içeren uzaktan erişim.
- **Değişiklik Olayları:** Sistem veya uygulama değişiklikleri; veri değişiklikleri; uygulama ve bileşen kurulumu ve değişiklikleri.
- **Kötülükler/Tehditler:** Geçersiz girişler ve olası diğer uygulama kötüye kullanımı; uygulamayı etkilediği bilinen diğer güvenlik sorunları.
- **Kaynak Sorunları:** Tükenmiş kaynaklar ve kapasite aşımı; bağlantı sorunları ve problemleri; ulaşılmış sınırlar.
- **Karma Erişilebilirlik Sorunları:** Sistemlerin, uygulamaların ve uygulama bileşenlerinin başlatılması ve kapatılması; arızalar ve hatalar, özellikle uygulamanın erişilebilirliğini etkileyen hatalar; erişilebilirliği etkileyen yedekleme başarıları ve başarısızlıkları. (Chuvakin vd. 2013)

8.2.2 Loglama API’si

Burada herhangi bir loglama API'si hakkında ayrıntılı bilgi vermek kapsam dışındadır. Tablo 8.2’de geliştirilen uygulamalarda yaygın olarak kullanılan loglama API'lerinin bir listesi gösterilmektedir.

Tablo 8.2 Dillere Göre Yaygın Loglama Kütüphaneleri

Programlama Dili	Kütüphane
.NET	Log4Net
	NLog
Java	Log4j
	Logback
C/C++	Log4C
Unix/Shell	Logger

Kısaca bahsedilmesi gerekirse; Log4Net, .NET için popüler bir loglama API'sidir. Loglama davranışı, yapılandırma dosyaları ile kontrol edilebilir. Aşağıda, Log4Net yapılandırma dosyasının bir örneği gösterilmektedir.

```
<configuration>
  <configSections>
    <section name="log4net" type="log4net.Config.Log4NetConfigurationSectionHandler, log4net"/>
  </configSections>
  <log4net>
    <root>
      <level value="DEBUG" />
      <appender-ref ref="LogAppender" />
    </root>
    <appender name="LogAppender" type="log4net.Appender.ConsoleAppender">
      <layout type="log4net.Layout.PatternLayout">
        <conversionPattern value=%date{dd.MM.yyyy HH:mm:ss.FFF} %level [%thread] %logger - %message%n" />
      </layout>
    </appender>
  </log4net>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.2" />
  </startup>
</configuration>
```

Bu örnek, standart çıkışa (*Console*) loglayacaktır ve en basit örnektir. Yukarıdaki yapılandırmayı kullanan bir uygulamadan üretilen log mesajının biçimi aşağıdaki gibi görünecektir:

```
15.04.2017 12:20:09.924 [1] INFO OrnekUygulama.Program - Giriş Yapıldı!
```

Yapılandırmadaki şu satır biçimlendirmeyi yönlendirir:

```
%date{dd.MM.yyyy HH:mm:ss.FFF} %level [%thread] %logger - %message%n"
```

Burada yer alan her parametrenin açıklaması sırayla şunlardır:

- %date{dd.MM.yyyy HH:mm:ss.FFF}:%date loglanan olayın tarihini verir. Sağ ve sol süslü parantez, Log4Net'in burada belirtilen tarih biçimlendirmesini kabul etmesini söyler.
- %level: Olay için belirtilen seviyeyi gösterir.
- [%thread]: Sağ ve sol köşeli parantezler, log mesajında aynı şekilde görülecektir. %thread, iş parçacığının adını veya loglama mesajının çağrıldığı metodu yazacaktır.
- %logger: Uygulamanın kendisidir.
- %message%n: %message, loglama çağrısına verilen (geliştici tarafından) ham log mesajıdır. %n, platforma bağlı satır ayırıcı karakterdir.

8.2.3 Log Rotasyonu

Log rotasyonu hakkında endişelenmek genelde geliştiricinin sorumluluğu değildir; ancak ne olduğunu iyi anlamak gerekir. İki temel log rotasyon mekanizması vardır:

- Log rotasyon komut dosyaları, logların rotasyonu yapıldığında ve bazı harici depolama konumlarına loglar arşivlendiğinde yönetmek amacıyla kullanılır.
- Uygulamanın kendisi, özel hazırlanmış uygulama kodu tarafından veya üçüncü taraf loglama kütüphanesinin dâhili özelliklerini kullanarak log rotasyon görevlerini gerçekleştirir.

Log dosyası rotasyonu, etkin log dosyasının bir kopyasının arşive taşındığı ve uygulamanın yazmaya başlaması için yeni boş bir dosyanın oluşturulduğu bir şemadır. Bu teknik, kısa süreli analiz veya yerinde arşivleme ve depolama için bir log dosyasının birden çok kopyası tutulduğu zaman önemlidir. Uygulama tarafında, log rotasyon işleminin gerçekleştiğini bilinmez.

İsteğe bağlı birkaç log rotasyon şeması vardır:

- Zamana dayalı: Bir log dosyasına, belirli bir zamana dayalı olarak rotasyon işlemi yapılır. Örneğin; saatlik, günlük vb.
- Boyuta dayalı: Bir log dosyası, önceden tanımlanmış bazı boyutlara ulaştığında rotasyon işlemi yapılır. Örneğin; 5 MB, 50 MB vb.

- Zamana ve boyuta dayalı: Bu şema, zamana ve boyuta dayalı şemaları birleştirir. Log dosyası zamana dayalı olarak arşivlenir; ancak her log dosyası da önceden tanımlanmış bazı boyutlarla sınırlandırılır.

Zamana ve boyuta dayalı rotasyon işlemi için Log4Net yapılandırması aşağıdaki gibidir:

```
<configuration>
  <configSections>
    <section name="log4net" type="log4net.Config.Log4NetConfigurationSectionHandler, Log4net"/>
  </configSections>
  <log4net>
    <root>
      <level value="DEBUG"/>
      <appender-ref ref="RollingFile" />
    </root>
    <appender name="RollingFile" type="log4net.Appender.RollingFileAppender" >
      <file value="logfile"/>
      <appendToFile value="true" />
      <rollingStyle value="Composite" />
    <datePattern value=".yyyyMMdd" />
      <maxSizeRollBackups value="-1" />
      <maximumFileSize value="5MB" />
      <staticLogFileName value="false" />
      <layout type="log4net.Layout.PatternLayout">
        <conversionPattern value="%date %level [%thread] %logger - %message%n" />
      </layout>
    </appender>
  </log4net>
</configuration>
```

Bu örnek yapılandırma dosyası, log dosyası 5 MB'ye ulaştığında veya her gün rotasyon işlemini yapar. Bu, çok fazla mesaj alan uygulamalar için yararlı bir teknik olabilir. Rotasyon işleminden önce 5 MB'a kadar büyüyen bir log dosyası yerine, daha yönetilebilir log veri yığını elde edilebilir.

8.2.4 Kötü Loglama Alışkanlıkları

Loglamadan bahsederken geliştiricilerin kötü alışkanlıklarına da değinmek gerekir. Kaçınılması gereken yaygın bazı kötü alışkanlıklar şu şekildedir:

- **Eksik zaman damgası ve saat dilimi:** Bu bilgi olmadan, logun ne zaman meydana geldiğini bilmek zorlaşır. Bu da soruşturma prosedürlerine, veri aramaya vb. zarar verebilir.
- **Sihirli veya gizli numaralar:** Log mesajlarında sihirli veya gizli numaralarla çok sık karşılaşılır. Buradaki sorun ise, numaranın neyi ifade

ettiğini anlamak için dokümanların bulunmamasıdır. Bu, yalnızca log incelerken yanlış bilgilere yol açmakla kalmaz, aynı zamanda hayal kırıklığına da sebep olabilir.

- **Belirsiz veya eksik açıklamalar:** Log girdileri açık, kısa ve anlaşılır olmalıdır. Belirsiz veya eksik açıklamalar yalnızca bir mesajı çözmeyi zorlaştırmakla kalmaz, aynı zamanda sistem kesintilerini veya olası güvenlik sorunlarını araştırırken zaman kaybettirebilirler.
- **Kaynak ve hedef IP, makine adı ve portun olmaması:** Tüm uygulamalar bağlantı yönelimli değildir; ancak istemci ve sunucu olan sistemler için, kaynak ve hedef IP, makine adı ve port bilgileri log girdisine dâhil edilmelidir.
- **Benzersiz mesaj tanımlayıcısı olmaması:** Her log mesajının benzersiz bir tanımlayıcısının olması önemlidir. Bu tanımlayıcı genellikle monoton olarak artan bir tamsayıdır. Bir mesajı benzersiz şekilde tanımlamak için bir yöntemin olması, log mesajında arama ve diğer işlemler için önemlidir. Pek çok programlama dili, evrensel benzersiz tanımlayıcılar üretmek için önceden tanımlanmış metotlar içerir.
- **Benzersiz mesaj türü tanımlayıcısının olmaması:** Benzersiz bir tanımlayıcıdan farklıdır. Mesajı, bir mesaj türü veya sınıfına ait olarak tanımlayan bir değerdir. (Chuvakin vd. 2013)

8.2.5 Log Mesajını Biçimlendirme

En iyi log mesajları, manuel veya otomatikleştirilmiş analize kendileri katkıda bulunanlardır. Bir log mesajı, bütün olarak alındığında daha fazla anlamın elde edilmesini sağlayan bileşenlerden oluşur. Aşağıdaki liste, yararlı bir log girdisine neyin ekleneceği konusunda bir fikir sunmaktadır.

- **Zaman damgası ve saat dilimi:** Ne zaman sorusunun yanıtlamasına yardımcı olur. Zaman dilimi, dağıtık uygulamalar için çok önemlidir. Zaman damgasına ve saat dilimine ek olarak, bazı yüksek hacimli sistemler bir işlem kimliği kullanır.
- **Kullanıcı adı:** Kullanıcı veya yönetici etkinlikleriyle alakalı olaylar için kim sorusunun yanıtlanmasına yardımcı olur. Buna ilave olarak,

mümkünse kullanıcı adını doğrulayan kimlik sağlayıcısının veya güvenlik alanının adını eklemek de yararlıdır.

- **Nesne:** Etkilenen sistem bileşenini veya nesneyi belirterek, ne sorusunun yanıtlanmasına yardımcı olur.
- **Sistem, uygulama veya bileşen:** Nerede sorusunun yanıtlanmasına yardımcı olur. Başlatıcı ve hedef sistemler, uygulamalar veya bileşenler gibi ilgili uygulama bağlamını sağlamalıdır.
- **Kaynak**: Ağ bağlantısıyla veya dağıtık uygulama operasyonuyla ilgili mesajlar için nereden sorusunun yanıtlanmasına yardımcı olur. Kaynak, IP adresi veya makine adı biçiminde olabilir. Uygulamaya bağlı olarak kullanılması gereken ilgili bileşenler hedef, kaynak port ve hedef porttur.
- **Nedeni:** Neden sorusunun yanıtlanmasına yardımcı olur. Böylece log analizinde gizli nedeni araştırmak için fazlaca uğraşmaya gerek duyulmaz.
- **Eylem:** Olayın doğasını sağlayarak nasıl sorusunun yanıtlanmasına yardımcı olur.
- **Durum:** Nesneye yönelik eylemin başarılı veya başarısız olduğunu açıklar ve ne sorusunun yanıtına da yardımcı olur.
- **Öncelik:** Olayın önemini belirtmeye yardımcı olur. Olayları derecelendirmek için tek bir ölçek belirlemek imkânsızdır. Çünkü farklı kurumlar farklı önceliklere sahip olacaklardır. Örneğin; kurumların bilgi kullanımına karşı gizlilik konusunda farklı politikaları olabilir.
- **Benzersiz oturum kimliği:** İlişkili mesajları birden çok işlem ve iş parçacığı arasında gruplamak için yardımcı olabilir.
- **İşlem kimliği ve iş parçacığı kimliği:** Çalışan bir uygulamanın log kayıtlarıyla ilişkilendirilebilmesine yardımcı olur. Uygulama, diğer uygulamaların paylaştığı bir log dosyasına yazıyorsa bu oldukça yararlıdır.
- **Etkinlik ölçümü:** Tüm uygulamaların loglaması gereken bir bileşen değildir. Örneğin; uygulama, bir kişi veya başka uygulamalar adına bir yerden başka bir yere veri aktarmak için hareket ediyorsa bu kullanılabilir. Bu bileşen loglandığı zaman, birilerinin beklenenden daha büyük bir veri yığınını ne zaman aktarmaya çalıştıklarını tespit etmek için analiz sistemleri tarafından kullanılabilir. (Chuvakin vd. 2013)

Yukarıdakileri temel alarak oluşturulan yararlı bir log mesajı örnek olarak aşağıda verilmektedir:

2017/02/27 11:20:05AM GMT+3 user=abc, object=database, system=xyzserver, source=172.21.19.55, module=dboperation, reason="could not open a connection", action=insert, status=failed, priority=2

Bu mesaj bahsedilen birçok bileşeni içerir. Her bileşen için ad/değer çiftleri kullanılır. İlgili çiftleri eşleştirmek için eşit karakteri kullanılır. Her bileşen için oluşturulan ad/değer çiftlerini ayırmak için virgül kullanılır. Bu şekilde, mesajın ayrıştırılmasına ilişkin bir belirsizlik de söz konusu olmaz. Ayrıştırma işleminde önce ad/değer çiftlerinin listesi alınarak virgüllere göre ayrılır. Ardından eşit karakterine göre her bir ad/değer çifti ayrılarak hem ad hem de değerine ulaşılır.

Log mesajının cevaplaması gereken sorular, yukarıda verilen örneğe uygulanırsa alınan cevaplar şu şekilde olacaktır:

- Ne oldu: Veritabanı işlemi başarısız oldu
- Nerede oldu: Veritabanında
- Ne zaman oldu: 2017/02/27 11:20:05AM GMT+3
- Neden oldu: Bağlantı açılamadı
- Nasıl oldu: Olmayan bir veritabanına kayıt eklenmeye çalışıldı
- Kim katıldı: abc

Burada görüldüğü gibi kullanılan biçim, log mesajı incelenirken sorulması gereken soruları kolayca cevaplamayı sağlar. Neyin olduğunu anlama konusundaki açıklık, gereken çözümün zamanında oluşturulabilmesini sağlar.

8.3 Güvenlik Hususları

Bir log mesajının hiçbir zaman içermemesi gereken bazı bilgiler vardır. İnsanları, finansal kayıtları, doğum tarihlerini ve bunun gibi diğer öğeleri tanımlamak için kullanılabilecek herhangi bir verinin loglanmaması gerekir. Bu veriler şunlardır:

- Şifreler
- Telefon Numaraları
- Kredi Kartı Numaraları
- Sosyal Güvenlik Numaraları

- Doğum Tarihleri
- Tam İsimler
- Sigorta Bilgisi
- Sürücü Belgesi Numaraları
- Genetik Bilgi
- Biyometrik Bilgi
- Her tür tanımlayıcı numara

8.4 Performans Hususları

Loglama yaparken uygulamanın performansının nasıl etkileneceği konusuna da değinmek gerekir. Kayıtları bir veritabanından okuyan ve işleyen sözde kod örneği aşağıda verilmektedir.

```
While(sonucSeti) {
        Log("Kayıt alınıyor!");
        Kayit = kaydiAl();
        Log("Kayıt alındı!");
        kaydiIsle(Kayit);
        Log("Kayıt başarıyla işlendi!");
}
```

Veritabanından bir sonuç seti döndüğünde yalnızca birkaç yüz kayıt işleniyorsa, bu önemli değildir. Fakat sonuç setinde binlerce, on binlerce veya hatta yüz binlerce kayıt olursa ve her dakika bu miktarda kayıt bu koda verilirse sonucu düşünmek gerekir. Log() metodu her çağrıldığında log dosyasını açar ve log mesajını yazar. Burada işletim sisteminin çıktıyı arabelleğe alıp almadığı hesaba katılmayarak Log() öğesi her çağrıldığında doğrudan log dosyasına yazdığı varsayılırsa her kayıt için üç log mesajı yazılacağı anlamına gelir. Yüz binlerce kayıt geldiği düşünülürse çok daha fazla log mesajı oluşturulacaktır. Uygulamanın yürüttüğü her işlemi loglamak yerine, bir loglama seviyesi şeması olması daha iyi bir fikirdir.

Çoğu loglama API'si, gerçek log mesajının kendisi ile birlikte bir loglama seviyesinin belirlenmesine izin verir. Loglama seviyeleri genellikle *Debug*, *Info*, *Warn*, *Error* ve *Fatal* olmaktadır. Her loglama seviyesinin anlamı oldukça açıklayıcıdır. Bu API'ler, log dosyasına hangi loglama seviyelerinin uygulanacağını yapılandırma ayarlarından açıp kapatmaya da izin verirler. Tablo 8.3'te örnek bir loglama seviyesi şeması gösterilmektedir.

Tablo 8.3 Örnek Loglama Seviyesi Şeması

Loglama Seviyesi	Açıklama	Varsayılan loglama
Debug	Hata ayıklama ile geliştiricilere yardımcı olabilecek log mesajlarını belirlemek için kullanılır.	Hayır
Info	Normal işlem mesajlarını ve bir uygulamanın durumunu loglamak için kullanılır.	Hayır
Warn	Zararlı olması muhtemel log mesajlarına dikkat çekmek için kullanılır.	Evet
Error	Uygulamanın çalışması için anormal bir durumdur; ancak uygulamanın çökmesine neden olmaz.	Evet
Fatal	Uygulamanın durmasına neden olacak ciddi durumlar içindir.	Evet

(Chuvakin vd. 2013)

Burada amaç, varsayılan olarak loglamak istenilen seviyeleri ve bir sorun oluştuğunda hangi seviyelerin açık olması gerektiğini belirleyebilmektir. Örneğin; bir uygulama düzgün çalışmıyorsa, aşağıdaki adımlar takip edilebilir:

1. Yapılandırma dosyasında *Debug* loglama seviyesinin etkinleştirilmesi.
2. Uygulamanın bir süre için hata ayıklama düzeyinde log mesajları yazmasına izin verilmesi.
3. Yapılandırma dosyasında *Debug* loglama seviyesinin devre dışı bırakılması.

Bu, gereksinim duyulan log mesajlarını toplamaya izin verirken uygulamanın en iyi şekilde çalıştığından da emin olmayı sağlar. Dikkat edilmesi gereken bir diğer husus da log mesajı ne kadar büyük olursa o kadar çok disk alanı kaplayacağıdır.

9. LOGLAR VE UYUMLULUK

Loglar, BT yöneticileri tarafından az değer görürken, güvenlik yönetimi için yararlı bilgiler sağlayabilir. Bununla birlikte, bu verilere ulaşmak zaman ve enerji gerektirir; her ikisi de kurumlar arasında sıklıkla yetersiz kalmaktadır. Verilerin hacmi ve niteliği göz önüne alındığında, başlangıçta korkutucu görünebilir. Stratejik değerleri göz önünde bulundurulduğunda loglar, sistem yöneticileri için bir veri kaynağı olmaktan daha fazlasıdır. Çoğu kullanıcı ve sistem eylemleri loglara kaydedilebildiğinden, bu tür etkinliği log yönetimi yazılımı veya diğer araçlar yoluyla izlemek, bilgi işlem hesap verebilirliğinin birincil aracıdır. Bir kurum için farklı hesap verebilirlik yolları da vardır; ancak loglar tüm BT'yi kapsayan, teknoloji sınırlarının ötesine uzanan tek mekanizmadır. BT operasyonu hesap verebilir değilse, bu işletmenin hesap verebilir olmadığı anlamına gelir.

Kurum logları ciddiye almıyorsa, BT hesap verebilirliği açısından ne kadar özenli olduğu da ele alınması gereken bir konudur. Bu nedenle loglama, PCI DSS, HIPAA, FISMA ve ISO27001 gibi bir dizi düzenleme ve yasalar tarafından zorunlu kılınan bir uyumluluk teknolojisidir.

Tüm BT kullanıcıları ister kötü niyetli ister iyi niyetli olsun, çeşitli loglarda etkinliklerinin izlerini bırakmaktadır. Bunlar dijital parmak izleri, kullanıcılara ait masaüstü bilgisayarları, sunucular ve güvenlik duvarları, yönlendiriciler, veritabanları ve iş uygulamaları gibi bir dizi BT bileşeni tarafından üretilir. Bu tür kayıtlar zamanla birikerek farklı türlerdeki log verilerinden yığınlar oluşturur.

Bütün düzenlemeler, log verileriyle ilgili olarak log kaydı, log yönetimi ve güvenlik izleme konularında aşağıdaki gereksinimlerin bazılarını veya tümünü gerektirir:

- Yeterli loglamayı sağlayın: Düzenlemeler, yeterlilik anlamına sahip olma konusunda önemli ölçüde değişir. Bazı düzenlemeler sadece bir kurumda denetim loglaması bulunması gerektiğini belirtir.
- Logları merkezi olarak toplayın: Bazı düzenlemeler log toplama, merkezi depolama ve analizi öngörür.
- Log verilerini gözden geçirin: Pek çok düzenlemenin en zahmetli kısmı, log incelemesi için bir zorunluluğun olmasıdır. Örneğin; PCI DSS,

sistemlerdeki logları günlük olarak incelemeyi gerektirir. Bu, her log girdisinin bir kişi tarafından okunması gerektiği anlamına gelmez.

- Logları belirli bir süre tutun: Düzenlemeler, loglar için aylardan yıllara kadar çeşitli tutma süreleri öngörür. Bazıları süre belirtmeden kurumun bir log saklama politikasına sahip olması gerektiğini söyler.
- Güvenliği izleyin: Bazı düzenlemeler, ağ ve web uyarılarını incelemeyi öngörür ve gerektiğinde olaya müdahale işlemine yer verir. Log verilerinin korunması, zaman senkronizasyonu gibi ilave görevler içerebilir. (Chuvakin, Log Management and Compliance)

Bu bölümde birkaç yaygın düzenleme ele alınarak bunların loglama, log analizi ve log yönetimi ile nasıl ilişkili oldukları incelenecektir.

9.1 PCI DSS

Bu kısım, PCI DSS düzenlemesiyle loglamanın temellerini ve PCI DSS'e göre neyin gerekli olduğunu kapsar. PCI DSS'de Gereksinim 10 başta olmak üzere diğer gereksinimlerin bazı bölümleri, loglama ve izlemenin gerekliliğinden bahseder.

9.1.1 Gereksinim 10

Gereksinim 10.1, özellikle sistem bileşenlerine yapılan tüm erişimin her bir kullanıcıya bağlanması sürecini kapsar. Logların bulunması veya bir loglama sürecinin oluşturulması zorunluluğu yerine, logların bireysel kişilere (üretildiği bilgisayarlar veya cihazlar değil) bağlı olması gerekliliğinden bahseder. Son kullanıcıların gerçek kullanıcılara eşleştirilmesi, çoğu zaman ilave zorluklar çıkarır. Bir kurumun tüm kullanıcılara, sistem bileşenlerine veya kart sahibi verilerine erişmesine izin vermeden önce benzersiz bir kimlik atamasını zorlayan PCI DSS Gereksinim 8.1, logları burada daha kullanışlı yapmaya yardımcı olmaktadır. (Chuvakin, Complete PCI DSS Log Review Procedures)

Gereksinim 10.2, loglanacak sistem olaylarının asgari bir listesini tanımlar. Bu gereklilikler, kredi kartı verilerini etkileyebilecek kullanıcı eylemlerini ve diğer olayları (sistem arızaları gibi) değerlendirmek ve izlemek ihtiyacı nedeniyledir. Loglanması gereken olayların listesi şu şekildedir: (Chuvakin, Complete PCI DSS Log Review Procedures)

- 10.2.1: Kart sahibi verilerine erişen tüm bireysel kullanıcılar
- 10.2.2: *root* veya yönetici ayrıcalıklarına sahip herhangi bir kişi tarafından gerçekleştirilen tüm işlemler
- 10.2.3: Tüm denetim izlerine erişim
- 10.2.4: Geçersiz mantıksal erişim denemeleri
- 10.2.5: Kimlik tespiti ve doğrulama mekanizmalarının kullanımı
- 10.2.6: Denetim loglarının başlatılması
- 10.2.7: Sistem seviyesindeki nesnelerin oluşturulması ve silinmesi

Yukarıdaki olaylar, veri erişimi, ayrıcalıklı kullanıcı eylemleri, log erişimi ve başlatma, başarısız erişim denemeleri, kimlik doğrulama ve yetkilendirme kararları ve sistem nesnesi değişikliklerini kapsar.

Gereksinim 10.3, her bir olay için loglanması gereken belirli veri alanlarını içerir. Bu alanlar, çeşitli loglama mekanizmaları ile genellikle aşılmış olan minimum gereksinimlerdir ve olay analizi ile soruşturma için gerekli olan tüm temel nitelikleri içerirler; ne zaman, kim, nerede, ne ve nereden. Belirtilen ilgili alanlar şunlardır:

- 10.3.1 Kullanıcı kimliği
- 10.3.2 Olayın türü
- 10.3.3 Tarih ve saat
- 10.3.4 Başarı veya başarısızlık göstergesi
- 10.3.5 Olayın kaynağı
- 10.3.6 Etkilenen verilerin, sistem bileşeninin veya kaynağın kimliği veya adı (PCI DSS, 2016)

Gereksinim 10.4, tüm loglarda doğru ve tutarlı bir zamana sahip olma gerekliliğinden bahseder. Bu gereksinim, zaman ve güvenlik olay denetimini birbiriyle bütünleştirmektedir. Sistem saatinin, ev veya küçük ofis ağında isteğe bağlı olduğu sıklıkla görülür. Bu gereksinime göre sistemlerin, Ağ Zaman Protokolü (NTP) sunucuları gibi güvenilir bir kaynaktan zaman senkronizasyonu elde etmek üzere yapılandırılması gerekir.

PCI DSS, logların gizlilik, bütünlük ve erişilebilirlik konularını da ele alır. Gereksinim 10.5.1, gizlilik konusunu kapsayarak denetim izlerinin işle ilgili ihtiyacı

olan kişilere sınırlandırılması gerekliliğinden bahseder. Bu gereksinim, işlerini yerine getirmek için logları görmesi gereken kişilerin sadece görebileceği anlamına gelir. Bunun nedenlerinden biri, kimlik doğrulama ile ilgili logların her zaman kullanıcı adlarını içermesidir. Kullanıcı adı bilgileri gizli olmamasına rağmen kimlik doğrulama için gerekli bilgilerin yüzde 50'sini oluşturur. Ayrıca kimlik bilgilerini yanlış yazan kullanıcılar nedeniyle parolaların loglarda görünmesi de olasıdır. Kötü yazılmış web uygulamaları, web sunucusu loglarında URL ile birlikte bir parolanın loglanmasına neden olabilir.

Gereksinim 10.5.2, bütünlük konusuna değinerek denetim iz dosyalarının yetkisiz değişikliklerden korunması gerekliliğinden bahseder. Logların yetkisiz kişilerce değiştirilebilmesi, sistem ve kullanıcı etkinliklerini objektif bir değerlendirme izi olmaktan çıkarır.

Log dosyalarını yalnızca kötü niyetli kullanıcılardan değil aynı zamanda sistem arızaları ve sistem yapılandırma hatalarının sonuçlarından da korumak gerekir. Bu hem log verilerinin erişilebilirliğini hem de bütünlüğünü kapsar. Gereksinim 10.5.3, denetim iz dosyalarının merkezi bir log sunucusuna veya değiştirilmesi zor ortamlara hızlıca yedeklenmesi gerekliliğinden bahseder. Logları, log analizinde kullanılabilecek bir sunucuya veya bir sunucu grubuna merkezileştirmek hem log koruması hem de log kullanışlılığının artırılması için gereklidir. Logları CD'lere, DVD'lere ya da kasetlere yedeklemek bu gerekliliğin bir başka sonucudur.

Yönlendiriciler ve anahtarlar gibi birçok ağ altyapısı, harici bir sunucuya loglamak ve cihazın üstündeki logları minimumda tutmak üzere tasarlanmıştır. Bu nedenle, bu sistemler için logların merkezileştirilmesi çok kritiktir. Gereksinim 10.5.4, kablosuz ağlar için logları yerel alan ağı üzerindeki bir log sunucusuna kopyalamanın gerekliliğinden bahseder.

Log değişikliği riskini daha da azaltmak ve değişiklik olmadığını ispatlamak için Gereksinim 10.5.5, mevcut log verilerinin uyarı vermeden değiştirilememesini sağlamak amacıyla loglar üzerinde dosya bütünlüğü izleme ve değişiklik algılama yazılımı kullanma gerekliliğinden bahseder. Aynı zamanda log dosyaları büyüme eğiliminde olduğundan bir log dosyasına yeni log verileri eklerken uyarı üretilmemelidir. Dosya bütünlüğü izleme sistemleri, dosyaları bilinen iyi bir kopyayla

karşılaştırmak için kriptografik özetleme (hashing) algoritmaları kullanır. Log dosyalarına yeni kayıt eklenmesi bütünlük izlemeyi zayıflatır. Dolayısıyla bütünlük izleme, yalnızca loglama bileşenleri tarafından aktif olarak yazılmayan logların bütünlüğünü garanti eder.

Gereksinim 10.6, tüm sistem bileşenlerinin loglarını en az günlük olarak incelemeyi ve bu incelemelerin IDS/IPS gibi güvenlik işlevleri olan sunucuları içermesi gerekliliğinden bahseder. Bu gereksinim, yalnızca loglamak için yapılandırılmayan, bununla birlikte korunmuş ve merkezileştirilmiş loglara sahip olan günlük olarak incelenmesi gereken log kaynaklarını kapsar. Büyük bir BT ortamının günde yüz gigabayt log üretebileceği göz önüne alındığında, tüm logları okumak insan tarafından imkânsızdır. Bu nedenle PCI DSS, log toplama, ayrıştırma ve uyarı araçlarının bu gereksinime uyum sağlamak için kullanılabileceğini belirten bir not eklemektedir.

Gereksinim 10.7, log tutma konusu ile ilgili olup denetim izi geçmişini en az 3 ayı çevrimiçi erişim olmak üzere en az 1 yıllığına tutulmasının gerekliliğinden bahseder. Dolayısıyla, bir yıl geriye dönük olarak loglara bakılamıyorsa, bu gereksinim ihlal edilmiş demektir.

PCI DSS'de loglama konusunda buraya kadar anlatılanların özeti şunlardır:

- Önceden tanımlanmış ayrıntı seviyesiyle birlikte kapsamda yer alan tüm sistemlerde belirli olaylar loglanmalıdır.
- Tüm loglanmış eylemlerle gerçek kullanıcılar ilişkilendirilmelidir.
- Kapsamda yer alan sistemlerin zamanları senkronize edilmelidir.
- Toplanan tüm logların gizlilik, bütünlük ve erişilebilirlikleri korunmalıdır.
- Belirli loglar en az günlük olmak üzere loglar düzenli olarak incelenmelidir.
- Kapsamda yer alan tüm loglar en az 1 yıl muhafaza edilmelidir.

9.1.2 Diğer Gereksinimler

Loglar yalnızca Gereksinim 10'da değil, diğer PCI gereksinimlerinde de tamamen açık olmasa da genel olarak ifade edilmektedir.

Gereksinim 1, kart sahibi verilerini korumak için bir güvenlik duvarı yapılandırması kurulmasını, kurumların tüm harici ağ bağlantıları ve güvenlik duvarı yapılandırmasında yapılan değişiklikleri onaylamak ve test etmek için resmi bir sürece sahip olmalarını gerektirir. Böyle bir işlemin ardından güvenlik duvarı yapılandırması değişikliklerinin yetkiyle ve belgelenmiş değişim yönetim prosedürlerine uygun olarak gerçekleştiği doğrulanmalıdır. Loglama, nelerin gerçekleştiğini gösterebilmesi sebebiyle burada kullanışlı hale gelmektedir.

Gereksinim 1.3'ün tamamı, gelen ve giden bağlantıyla ilgili özel ifadeler içeren güvenlik duvarı yapılandırmasına yönelik bir kılavuz içerir. Bunu doğrulamak için güvenlik duvarı loglarını kullanmak gerekir. Loglar sadece nasıl yapılandırıldığını değil nasıl gerçekleştirildiğini de gösterdiğinden yalnızca yapılandırmanın gözden geçirilmesi yeterli olmayacaktır.

Gereksinim 2, şifre yönetimi ve gereksiz hizmetleri çalıştırmamak gibi genel güvenlik pratiklerinden bahseder. Loglar, daha önce devre dışı bırakılan hizmetlerin bilgisiz sistem yöneticileri ya da saldırganlar tarafından ne zaman başlatıldığını gösterebilir.

Gereksinim 3, veri şifrelemeyi ele alarak loglamaya doğrudan bağlantılar içermektedir. Gereksinim 3.6'nın tamamı, bu tür bir etkinliğin fiilen gerçekleştiğini doğrulamak için logların bulunması anlamına gelir. Özellikle, anahtar üretimi, dağıtımı ve iptali çoğu şifreleme sistemi tarafından loglanır ve bu loglar bu gereksinimi karşılamak için kritik önem taşır. Şifreleme ile ilgili Gereksinim 4'te benzer nedenlerle loglama çıkarımlarına sahiptir.

Gereksinim 5, antivirüs korumalarını ele alarak tüm antivirüs mekanizmalarının güncel olduğundan, aktif şekilde çalıştığından ve denetim logları üretebildiğinden emin olmanın gerekliliğinden bahseder. Bu gereksinimi yerine getirmek için bahsedilen logları görmek gerekir. Antivirüs yazılımı kullanma ve düzenli olarak güncelleme gereksinimi bile, antivirüs loglarında mevcut olduğundan değerlendirme sırasında log verileri için ihtiyaç oluşturacaktır. Ayrıca başarısız antivirüs güncellemelerinin kurumu zararlı yazılım risklerine maruz bıraktığı ve bunların da loglara yansıtıldığı bilinmektedir. Son imza güncellemelerine sahip olmayan antivirüs, yanlış bir güvenlik hissi oluşturur ve uyumluluk çabasını zayıflatır.

Gereksinim 6, güvenli sistem ve uygulamaların geliştirilmesi ve devamlılığının gerekliliğinden bahseder. Bu gereksinim, loglama fonksiyonlarının değerlendirilmesi ve uygulama güvenliğinin izlenmesi olmadan düşünülemez.

Gereksinim 7, kart sahibine ait verilere erişimin iş gereksinimlerine göre kısıtlanmasını gerekliliğinden bahseder. Bu gereksinim, söz konusu verilere kimlerin erişebildiğinin doğrulaması için logları gerektirir.

Gereksinim 8, bilgisayar erişimi olan her kişiye benzersiz bir kimlik atamanın gerekliliğinden bahseder. Sisteme erişen her kullanıcıya benzersiz bir kimlik atamak, diğer güvenlik pratikleri ile uyuşmaktadır. Gereksinim 8.5.1, kullanıcı kimliklerinin, kimlik bilgilerinin ve diğer tanımlayıcı nesnelerin eklenmesi, silinmesi ve değiştirilmesinin kontrol edilmesi gerekliliğinden bahseder. Günümüzde çoğu sistem bu tür etkinlikleri loglamaktadır. Gereksinim 8.5.9, kullanıcı şifrelerini en az her üç ayda bir değiştirme gerekliliğinden bahseder. Bu gereksinim, tüm hesapların şifrelerinin en az üç ayda bir değiştiğinden emin olmak için sunucudaki log dosyalarını incelemeyi gerektirir.

Gereksinim 9, ziyaretçi etkinliğinin fiziksel bir değerlendirme izinin sağlanılması için bir ziyaretçi logu kullanmanın ve yasayla aksine kısıtlamalar olmadıkça bu logu en az üç ay süreyle saklamanın gerekliliğinden bahseder.

Gereksinim 11, güvenlik açıkları için kapsamda yer alan sistemleri tarama gerekliliğinden bahseder. Gereksinim 11.4, IDS/IPS kullanımına değinerek tüm ağ trafiğini izlemek ve şüphelenilen tavizlere karşı personeli uyarmak için ağ saldırı tespit sistemleri, sunucu tabanlı saldırı tespit sistemleri ve saldırı önleme sistemleri kullanılması ile saldırı tespit ve önleme cihazlarının güncel tutulması gerekliliğinden bahseder. Saldırı tespiti yalnızca loglar ve uyarılar gözden geçirilirse yararlı olur.

Gereksinim 12, güvenlik standartları ve günlük operasyonel prosedürlerin yanı sıra üst düzey bir güvenlik politikasından bahseder. Loglamanın değerlendirilmesi her güvenlik politikasının bir parçası olması gerektiğinden loglama çıkarımları da vardır. Buna ilave olarak, her durumun zamanında ve etkili bir şekilde ele alınmasını sağlamak amacıyla güvenlik olayına müdahale ve yükseltme prosedürlerinin oluşturulması, belgelenmesi ve dağıtılması gereksinimi; log verilerinin etkin bir

şekilde toplanması ve zamanında incelenmesi işlemi gerçekleştirilmeden düşünülemez.

Sonuç olarak PCI DSS programındaki olay loglama ve güvenlik izleme, Gereksinim 10'un çok ötesine geçer. Sadece dikkatli veri toplama ve analiz yoluyla kurumlar, PCI'ın geniş gereksinimlerini karşılayabilirler.

9.2 ISO27001

ISO27001, ISO27000 standartlarına aittir. Uluslararası Standartlar Organizasyonu (ISO) tarafından yayınlanan bir Bilgi Güvenliği Yönetim Sistemi (BGYS) standardıdır. Standardın tam adı ISO/IEC 27001:2013 "Bilgi teknolojisi - Güvenlik teknikleri - Bilgi güvenliği yönetim sistemleri - Gereksinimler" dir.

Loglama ve izleme ile ilgili belirli ISO kontrolleri, yetkisiz bilgi işlem faaliyetlerini tespit etmek amacına yönelik olarak A.12.4 "Kaydetme ve İzleme" bölümünde yoğunlaşmaktadır.

A.12.4.1 "Olay Kaydetme" bölümü, kullanıcı işlemleri, kural dışılıklar, hatalar ve bilgi güvenliği olaylarını kaydeden olay kayıtlarının üretilmesi, saklanması ve düzenli olarak gözden geçirilmesi gerekliliğinden bahseder. Bu, logların bulunmasını ve önceden belirlenmiş bir süre boyunca tutulmasını öngörür. Ayrıca, loglarda hangi olay türlerinin kaydedilmesi gerektiğini vurgular.

A.12.4.3 "Yönetici ve Operatör Kayıtları" bölümü, sistem yöneticileri ve sistem operatörlerinin işlemlerinin kayıt altına alınması, kayıtların korunması ve düzenli olarak gözden geçirilmesi gerekliliğinden bahseder. Tablo 9.1'de A.12.4.1 kontrolüne A.12.4.3 ek gereksinimleri de dâhil edilerek ISO ortamında tüm sistemlerde loglanması gereken olay türleri özetlenmektedir.

Tablo 9.1'deki konular, Windows'ta oturum açılışlarından politika değişikliklerine, uygulama güncellemelerinden veri ile ilgili kullanıcı işlemlerine kadar geniş bir yelpazedeki olayları kapsar. Ağ cihazlarında bunlar, güvenlik ve erişilebilirlik sorunlarını kapsamaktadır.

Sistem yöneticisi ve sistem operatörü etkinlikleri gibi ayrıcalıklı kullanıcıları izlemeye özel önem verilir. Bunlar, bir ISO programında oluşturulacak en önemli

loglardan bazıları olup BT yöneticileri için hesap verebilirliğin ana aracıdır. Bununla birlikte bu izlemelerin yararlı olması için, logların bu tür yöneticilerin kontrolünden bir log yönetimi çözümü aracılığıyla çıkarılması gerekir.

Tablo 9.1 ISO Ortamında Loglanması Gereken Olay Türleri

Loglanan Mesaj Kategorisi	ISO27001 Kapsamında Kayıt ve İzleme
Kimlik doğrulama, yetkilendirme ve erişim olayları; başarılı ve başarısız kimlik doğrulama kararları, sistem erişimi, veri erişimi, uygulama bileşeni erişimi ve uzaktan erişim	Yetkili kullanıcılar, saldırganlar ve içerdeki kötü niyetliler tarafından sistemlere yapılan erişimi izlemek ve incelemek
Sistem veya uygulama değişiklikleri, veri değişiklikleri, uygulama ve bileşen kurulumları ve güncellemeleri gibi değişiklikler	Sistemin saldırılara maruz kalabileceği değişiklikleri ve saldırganlar ile içerdeki kötü niyetliler tarafından yapılan değişiklikleri izlemek
Sistemlerin, uygulamaların ve uygulama bileşenlerinin başlatılması ve kapatılması gibi erişilebilirlik sorunları; arızalar ve hatalar, özellikle uygulamanın erişilebilirliğini ve veri güvenliğini etkileyen hatalar	Operasyonel sorunların yanı sıra güvenlikle ilgili arızaları ve sistem erişilebilirliğini izlemek
Tükenen kaynakları, aşılan kapasiteleri, ağ bağlantısı erişilebilirlik sorunlarını ve ulaşılan çeşitli sınırları kapsayan kaynak sorunları	Güvenlik ve operasyonel nedenlerle kaynak sorunlarını tespit etmek
Geçersiz girişler ve diğer muhtemel uygulama kötüye kullanımı gibi bilinen tehditler, istismarlar ve bilinen saldırılar	Saldırıları tespit etmek ve engellemekle birlikte bunların sonuçlarını araştırmak

(Chuvakin vd. 2013)

Yukarıda ele alınan bölümlerde, log tutma ve soruşturma için ek gereksinimler de bulunmaktadır. Açık olarak 1 yıllık log tutma talimatı veren PCI DSS'nin aksine ISO, yalnızca log tutmaya değinir ve log tutma süresini tanımlamak için ayrı bir politika gerektirir. Kurumların log tutmayı, bütün ortamlarda hem fiziksel hem de sanal bileşenler üzerinden gerçekleştirmeleri önerilir.

ISO27002, ISO serisi uygulanırken pratikte yararlı olan ilave ayrıntıları sağlar. Örneğin; A.12.4.1 "Olay Kaydetme" bölümü için ek kılavuzda ayrıcalıkların, sistem araçlarının ve uygulamaların kullanımının loglanması gerekliliğinden bahseder.

ISO27001, logları yetkisiz değiştirme ve gözetim altına alma konusunu kapsar. A.12.4.2 “Kayıt Bilgisinin Korunması” bölümü, kaydetme olanaklarının ve kayıt bilgilerinin kurcalama ve yetkisiz erişime karşı korunması gerekliliğinden bahseder. Log korumak için iki önemli noktaya dikkat edilmesi gerekir: erişim kontrolü ve bütünlüğün korunması. Çoğu log yönetimi ve SIEM sistemi, role dayalı katı erişim kontrolleri sunmaktadır. Bunun yanında sistemler olası log değişikliklerini saptamak

için kriptografik mekanizmalar kullanan bütünlük denetimi uygularlar. En son ihtimal olarak, loglar DVD veya ağ yedekleme arşivleri gibi ortamlara yazılabilir. Ayrıca ISO27002'ye göre kurum, olayları kaydetmede başarısızlıkla sonuçlanan veya daha önceden kaydedilen olayların üzerine yazılmasına neden olan log dosyası ortamının depolama kapasitesini izlemelidir.

A.12.4.4 “Saat Senkronizasyonu” bölümü, bir kurum veya güvenlik alanında yer alan bilgi işleme sistemlerinin saatlerinin tek bir referans zaman kaynağına göre senkronize edilmesi gerekliliğinden bahseder. PCI DSS'de olduğu gibi log zamanlaması, güvenlik izleme, adli bilişim ve sorun giderme için önemlidir. Toplanan ve analiz edilen tüm logların doğru zaman damgalarına sahip olduğundan ve ilgili saat dilimi bilgilerinin korunduğundan emin olmak gerekir. Loglar için NTP veya diğer güvenilir zamanlama mekanizmalarını kullanmak, olayların doğru sıralamasının ve gerçek zamanlarının korunması için gereklidir.

9.3 HIPAA

Sağlık bilgileri için hem elektronik hem de fiziksel olmak üzere güvenlik ve gizlilikle ilgili standartları özetlemektedir. Yasanın temel amacı, sağlık sigortası ve sağlık hizmeti sunumunda dolandırıcılık ve istismarlarla mücadele etmek için toplulukta ve bireysel piyasalardaki sağlık sigortası kapsamının taşınabilirliğini ve sürekliliğini sağlamaktır. (HIPAA Act of 1996).

HIPAA'nın yeni bir geliştirmesine HITECH (Health Information Technology for Economic and Clinical Health) Yasası adı verilir. Yasa, sağlık bilgi teknolojisinin benimsenmesini ve anlamlı bir şekilde kullanılmasını teşvik ederek HIPAA kurallarının hukuki ve cezai uygulamasını güçlendiren çeşitli hükümler vasıtasıyla sağlık bilgilerinin elektronik olarak iletilmesiyle ilgili gizlilik ve güvenlik konularını ele almaktadır. (HITECH Act of 2009)

Aşağıdaki HIPAA gereksinimleri loglama, log incelemesi ve güvenlik izleme için genel olarak uygulanabilir.

- 164.308 (a) "Giriş İzleme" bölümü, oturum açma ve erişim için hasta bilgilerine değinen sistemlerin izlenmesini gerektirir. Gereklilik, başarılı ve başarısız sonuçlanan bütün oturum açma girişimleri için uygulanır.

- 164.312 (b) "Denetim Kontrolleri" bölümü, hassas sağlık bilgileriyle ilgilenen sistemlerin üzerindeki denetim loglama ve diğer denetim izlerini kapsar. Denetim loglarının incelenmesi bu gereklilikle ifade edilmektedir.
- 164.308 (a) "Bilgi Sistemi Faaliyet Değerlendirmesi" bölümü, loglar, sistem kullanım raporları, olay raporları ve güvenlikle ilgili faaliyetlerin diğer göstergeleri gibi IT etkinliklerinin incelenmesini öngörür. (NIST, 2008)

Yukarıdaki gereksinimlere göre HIPAA içindeki loglama ve izleme gereksinimi PCI DSS ile karşılaştırıldığında, kurumların loglama ve log yönetimini yerleştirmesi ve işletmesi için gerekli olan önemli soruların hem teknik hem de prosedür açısından cevaplamasına yardımcı değildir. Özellikle şu sorular cevaplanmamaktadır:

- Denetim kontrolleri tarafından hangi etkinlikler ve olaylar loglanmalıdır?
- Her etkinlik veya olay için hangi ayrıntılar verilmelidir?
- Loglar merkezi olarak toplanmalı mıdır?
- Kayıtlar ne kadar süreyle tutulmalıdır?
- Hangi etkinlikler ne sıklıkla incelenmelidir?
- Güvenlik izleme ve giriş izlemesi nasıl yapılmalıdır?
- Denetim kayıtları nasıl korunmalıdır?

Bu bilgiler ışığında HIPAA log toplama ve incelemesinin, her ölçekteki kurum için bir engel olduğu görülmektedir. Log gereksinimleri, karmaşık sistemler bulunan kurumlar veya uzmanlıktan yoksun bazı küçük şirketler için zor olabilir. Belirli olmayan rehberlik, kurumların loglama ve log incelemesi yapma konusunda motive olmalarına yardımcı değildir. Bununla birlikte loglama ve log incelemesi karmaşıklığı, özel olarak geliştirilmiş uygulamalar da kapsama dâhil edildiğinde çok fazla artar.

HIPAA güvenlik kuralının uygulanması için bazı ek ayrıntılar NIST 800-66 "HIPAA Güvenlik Kuralının Uygulanmasına Yönelik Bir Giriş Kaynak Kılavuzu" nda bahsedilmektedir. NIST 800-66 kılavuzu, HIPAA güvenlik kuralına dayalı elektronik korumalı sağlık bilgilerinin korunması için log yönetimi gereksinimlerini ayrıntılı olarak açıklamaktadır.

9.3.1 NIST 800-66

NIST 800-66 kılavuzunda yer alan 4.1 bölümü, denetim logları, bilgi ve sistem erişimi raporları ve güvenlik olayı izleme raporları gibi bilgi sistemi etkinliğinin düzenli olarak incelenmesi ihtiyacını açıklamaktadır. Bölüm, cevap vermek yerine ne sıklıkta incelemelerin gerçekleştireceği ve denetim bilgilerinin nerede bulunacağıyla ilgili sorular sormaktadır.

4.15 bölümü, denetim kontrolleri ile ilgili ilave rehberlik sağlamaya çalışmaktadır. Uygulayıcılara belirli bir yöntem sağlamak için hangi faaliyetlerin izleneceği ve denetim kayıtlarının neler içermesi gerektiği gibi sorular sormaktadır. Bunların uygulanması noktasındaki ihtiyaçlar ele alınmamaktadır.

4.22 bölümü, eylemlerin ve etkinliklerin dokümantasyonunun en az 6 yıl süreyle tutulması gerektiğini belirterek logların uygulayıcılara dokümantasyon olarak düşünülüp düşünülmeyeceği konusunu ele almaz.

NIST 800-66'daki yönlendirici sorular kullanılarak bir politikanın neler içermesi gerektiği formüle edilebilir; gereksinim uygulanabilirliği, kaydedilen etkinlikler ve kaydedilen ayrıntılar, inceleme prosedürleri, istisna izleme süreci gibi.

- Genel sürecin ve sonuçların sorumlusu kimdir?
- İncelemeler ne sıklıkla yapılacaktır?
- İnceleme sonuçları ne sıklıkta analiz edilecektir?
- Çalışanların ihlalleri için kurumun yaptırım politikası nedir?
- Denetim bilgileri nerede bulunacaktır? (NIST, 2008)

Kurum hem loglama hem de log incelemesi için yukarıdaki yöntemi uygulamalıdır. Bu şekilde, kapsamda yer alan sistemlerde log kayıtlarının oluşturulduğundan ve bu kayıtların yeterli ayrıntıya sahip olduğundan emin olunur. Bu tür ayrıntılar, ilgili PCI DSS kılavuzundan alınabilir. Ayrıca, denetim logları, erişim raporları ve güvenlik olayı izleme raporları gibi bilgi sistemi etkinliklerinin kayıtlarını düzenli olarak incelemek için prosedürler oluşturulması gerekir.

HIPAA, loglama ve log yönetimi hakkında detaylı bir rehberlik sağlamamakta ve uygulanması gereken güvenlik kontrolleri ve teknolojileri seviyesine inmemektedir. Dikkat edilmesi gereken bir husus, ödeme kartlarını kabul eden sigorta şirketleri ve

birçok hastanenin hem HIPAA hem de PCI DSS'ye tabi olmasıdır. Ödeme işlemi sistemleri, hastanın sağlık bilgilerini saklamamaları gerektiğinden kurum genelinde uygulanabilirliğinin kapsamı farklı olabilir. Yine de her iki yönetmelik için de aynı teknik ve idari kontrollerin göz önünde bulundurulması hem kısa hem de uzun vadede ihtiyatlı olacaktır.

NIST 800-66, HIPAA açıklamalarının yanı sıra HIPAA Güvenlik Gereksinimini karşılamak için neyin loglanacağı konusunda da bilgiler vermektedir.

Denetim logları, tam olarak ne olduğunu ne zaman, nerede ve nasıl gerçekleştiğini ve kimin yer aldığını anlatır. Bu loglar manuel, yarı otomatik ve otomatik analiz için uygundur. İdeal olarak, bunları üreten uygulamaya gerek kalmadan ve uygulama geliştiricisini çağırmadan analiz edilebilirler. Sağlık hizmeti uygulamaları söz konusu olduğunda bu tür geliştiriciler her zaman erişilebilir olmayabilir ve güvenlik ekibi kendi başlarına işlem yapmak zorunda kalabilir. Log yönetimi bakış açısından, bu loglar analiz ve tutma için merkezileştirilebilir. Bu loglar sistemi yavaşlatmamalıdır ve adli delil olarak kullanılıyorsa güvenilir oldukları kanıtlanabilir.

Her zaman kaydedilmesi gereken etkinlik veya olay türleri vardır. Örneğin; kimlik doğrulama kararları, sağlık bilgisi erişimi ve sistem değişiklikleri daima loglarda görünmelidir. Kaydedilen bir olayın her türü için ister bir insan ister otomatik bir sistem tarafından olsun, yorumlanması için zorunlu olan ayrıntılar bulunur. Örneğin; her logda güvenilir bir zaman damgası olmalıdır ve kullanıcı etkinliğiyle ilgili her logda ilgili kullanıcının kullanıcı adı bulunmalıdır. Bununla birlikte bazı ayrıntıların asla loglanmaması gerekir. Örneğin; uygulama veya sistem parolaları hiçbir zaman loglarda görünmemelidir. Sağlık bilgileri de loglardan uzak tutulmalıdır.

Sağlık bilgilerine kimin, ne zaman ve neden eriştiğini bilmek gerekir. Ayrıca, ekleyen, değiştiren veya silen kullanıcılar da bilinmelidir. Aynı zamanda bilgileri okumak, değiştirmek veya silmek için kimin deneme yaparak başarısız olduğunun da not edilmesi gerekir. Her bir veriye erişim kaydedilemiyorsa, uygulamanın kendisine olan tüm erişim dikkatlice kaydedilmelidir.

Sağlık bilgilerini işleyen sistemlerde kimlerin başka etkinliklerde bulunacağı bilinmelidir; çünkü bu tür faaliyetler gelecekteki sağlık verilerine erişimi etkileyebilir. Örneğin; birisinin oturumu kapattığı ya da verilere serbest erişime izin verecek yeni bir bileşen eklediği loglanmalıdır. Buna ek olarak, sağlık bilgi sistemlerinde önemsenen diğer kritik olaylar kaydedilmelidir; çünkü bu tür olaylar, yetkisiz erişime ilişkin önemli delilleri ortaya çıkarabilir. (Chuvakin, Logging for HIPAA Part 2)

9.4 FISMA

Her bir federal ajansın operasyonlarını ve varlıklarını destekleyen, başka bir ajans, yüklenici veya başka bir kaynak tarafından sağlanan veya yönetilenler de dâhil olmak üzere bilgi ve bilgi sistemleri için bilgi güvenliği sağlamak amacıyla doküman geliştirmesi ve ajans çapında bir program uygulaması gerekir. (FISMA Act of 2002)

FISMA, tamamının dokümantasyon olması ve hiçbir eylem olmamasından dolayı eleştirilirken bu yasa, her Federal ajansın operasyonlarını ve varlıklarını destekleyen bilgi sistemlerini güvence altına alacak kurum çapında bir program geliştirmesi, dokümante etmesi ve uygulamasının gerekliliğini vurgular.

Yasa, yüksek seviye bir politika, planlama ve risk olarak kaldığı için federal sistemlere loglama, log yönetimi veya güvenlik izleme kılavuzu vermez. Bu yasaya uygun olarak, FISMA uyumluluğunun özelliklerini kapsayacak şekilde NIST (National Institute of Standards and Technology) tarafından ayrıntılı bir kılavuz oluşturulmuştur. Bu kılavuz, NIST 800-53 "Federal Bilgi Sistemleri ve Kurumları için Güvenlik ve Gizlilik Kontrolleri" dir. Bu doküman, denetim kayıtlarının oluşturulması, gözden geçirilmesi, korunması ve tutulması dâhil olmak üzere log yönetimi kontrollerini açıklar ve denetim arızası durumunda atılması gereken adımları içerir. (Chuvakin, Detailed FISMA guidance and more)

9.4.1 NIST 800-53 Loglama Kılavuzu

AU-1 "Denetim ve Hesap Verebilirlik Politikası ve Prosedürleri" bölümü, denetim ve hesap verebilirlik politikasının uygulanmasını kolaylaştıracak formal, dokümante edilmiş prosedürleri ve bunlarla ilgili denetim ve hesap verebilirlik kontrolleri üzerine odaklanmaktadır. Bu kontrol, log toplama ve incelemesi için loglama politikası ve prosedürlerinden başlayarak denetim loglamaya girmeyi sağlar.

AU-2 "Denetlenebilir Olaylar" bölümü, NIST 800-92'yi ifade eder. Denetlenebilir olayların bir listesini oluşturmak için risk değerlendirmesi ve diğer organizasyonel birimlerin loglama ihtiyaçları göz önüne alınmalıdır. Özel koşullar kapsamında denetlenen olaylar da burada tanımlanmaktadır.

Denetlenecek olayların listesi oluşturulduktan sonra, AU-3 "Denetim Kayıtlarının İçeriği" bölümü, her olay için kaydedilen ayrıntıların seviyesini belirtir. Zaman damgaları, kaynak ve hedef adresleri, kullanıcı/süreç tanımlayıcıları, olay açıklamaları, başarı/başarısızlık göstergeleri, ilgili dosya adı ve çağrılan erişim kontrolü veya akış kontrol kuralları gibi her zaman iyi kayıtlarda olması gereken örnekler verilmektedir.

AU-4 "Denetim Depolama Kapasitesi" ve AU-11 "Denetim Kaydı Tutma" bölümleri, birçok kurum için kritik konu olan log tutmayı kapsar.

AU-5 "Denetim İşleme Arızalarına Yanıt" bölümü, loglama ve log analizinin önemli ama gözden kaçırılmış bir yönünü zorunlu kılar. Bu, loglama başarısız olduğunda eyleme geçilmesi gerekliliğidir. Yazılım/donanım hataları, denetim yakalama mekanizmalarındaki arızalar, denetim depolama kapasitesinin tükenmesi ve loglamayı etkileyen diğer sorunlarda eyleme geçilmesi gerektiğinden bahsedilir.

AU-6 "Denetim İnceleme, Analiz ve Raporlama" bölümü, toplanan log verileriyle ne olacağı hakkındadır. Kurumun uygun olmayan veya olağandışı etkinlik belirtileri için bilgi sistemi denetim kayıtlarını incelemesini ve analiz etmesini önermektedir.

AU-7 "Denetim Azaltma ve Rapor Üretme" bölümü, log verilerini incelemek için en yaygın yol olan raporlama ve özetleme ile ilgilidir.

AU-8 "Zaman Damgaları", AU-9 "Denetim Bilgilerinin Korunması" ve AU-10 "İnkâr Edememe" bölümleri, araştırma ve izleme amaçlarıyla log güvenilirliğini ele alır. Loglar, değişiklikleri önleyecek şekilde doğru zamanlanmalı ve saklanmalıdır. Burada sözü edilen bir seçenek donanım tarafında bir kez yazılabilir medyadır. Kriptografinin kullanımı ise bahsedilen başka bir yöntemdir.

AU-12 "Denetim Üretme" bölümü, kurumun AU-3'te tanımlanan içerik ile AU-2'de tanımlanan denetlenen olayların listesine uygun denetim kayıtları ürettiğinden emin olmasını sağlar.

AU-13 "Bilgi İfşası İçin İzleme" bölümü bilgi hırsızlığı (hassas verileri dışarı sızdırma) ve AU-14 "Oturum Denetimi" bölümleri kullanıcı etkinliğinin (oturum verisi) kayıt ve analizini kapsar.

Genel olarak, FISMA odaklı başarılı bir log yönetimi uygulaması için gereksinimi adreslemenin yolu, tüm log yönetimi projelerinde olduğu gibi bir kurumun türüne göre değişiklik gösterebilir.

FISMA/NIST kılavuzundan elde edilen çıkarımlar şunlardır:

- Loglama politikası oluşturun. (AU-1).
- Politikaya göre hangi olayın loglanacağını (AU-2) ve her olay için hangi ayrıntıların üretileceğini ve kaydedileceğini (AU-3) belirleyin. Loglamayı, AU-12'ye göre başlatın.
- Verilerin dışarı sızdırılmasını (AU-13) tespit etmek için tüm giden bağlantıları loglayın ve kullanıcı erişim oturumlarının kaydedildiğinden emin olun (AU-14).
- Log depolama yöntemlerini ve tutma sürelerini (AU-4, AU-11) tanımlayın ve oluşturulan logları tutun.
- Logları değişikliklerden koruyun, logların kanıtlayıcılığını korumak için zamanı doğru tutun (AU-8, AU-9, AU-10).
- Politikaya göre log inceleme prosedürlerini uygulayın ve rapor oluşturun (AU-6, AU-7). Raporları, bilgileri görmesi gereken taraflara dağıtın.

Bazı kurumların yaptığı bir hata, politika ve prosedür seviyesinde kalarak gerçek sistemleri loglamak için yapılandırmamaktır. Unutulmamalıdır ki dokümanlar zararlı bilgisayar korsanlarını durdurmaz, politikalar log verileri eksik olduğunda olayları soruşturmaya yardımcı olmaz ve uyum stratejisi hakkında konuşmak kurumu güvenli hale getirmez. (Chuvakin, FISMA How To)

9.4.2 NIST 800-92 Log Yönetim Kılavuzu

NIST 800-53'ün üzerine, NIST 800-92 "Bilgisayar Güvenliği Log Yönetimi Kılavuzu" oluşturulmuştur. Bu kılavuzda şu ifade yer almaktadır: "Aşağıdaki tavsiyelerin uygulanması, Federal bölümler ve ajanslar için daha efektif ve verimli log yönetiminin kolaylaştırılmasına yardımcı olmalıdır." (Kent vd. 2006)

NIST 800-92 kılavuzu, bilgisayar güvenlik log yönetimine girişle başlar ve üç ana bölümden oluşur:

- Log yönetimi altyapısı
- Log yönetimi planlaması
- Log yönetimi operasyonel süreçleri

Kılavuz, log yönetimini "bilgisayar güvenlik log verilerini üretmek, iletmek, saklamak, analiz etmek ve imha etmek için süreç" olarak tanımlar. Güvenlik log yönetimi, hem güvenlik uygulamalarından gelen logları (IPS alarmları gibi) hem de uygulamalardan gelen güvenlik loglarını (kullanıcı kimlik doğrulama kararları gibi) kapsamalıdır. Bunlardan yalnız birisine odaklanılması hatadır.

Kılavuz, log yönetimi altyapısı bölümünde log yönetimi mimarisinin üç katmanını tanımlar:

- Log oluşturma
- Log analizi ve saklama
- Log izleme

Birçok kurum ne yazık ki log politikalarını düşünmeden pahalı bir aracı satın almayla başlar ve alınan aracı loglar için kullanır. Aracı düşünmeden önce ihtiyaçları (loglardan almak istenilenler) ve logları (hangi logların yardımcı olabileceği) düşünmek, kurumu birçok sıkıntıdan kurtaracaktır.

Log yönetimi projesi planlanırken organizasyondaki roller önemlidir. Log yönetimi doğal olarak yataydır ve bir kurumun birçok alanına dokunmaktadır. NIST, sistem ve ağ yöneticilerinin, güvenlik yöneticilerinin, olaya müdahale edenlerin, güvenlik sorumlularının, denetçilerin ve hatta kurum içi uygulama geliştiricilerin log yönetimi projesine dâhil edilmesini önermektedir. Böyle bir birliktelik, kurumun log

yönetimi sorusuna doğru cevabı seçip uygulamasına yardımcı olacaktır. (Chuvakin, FISMA How To)

Yaygın bir düşünce, güvenliğin bir politikadan başladığıdır. Bu, loglama için de geçerlidir. Kılavuza göre, bir loglama politikasının şunları içermesi gerekir:

- Log üretme: Hangi olaylar hangi ayrıntı seviyesiyle birlikte loglanmalıdır?
- Log iletimi: Loglar tüm ortamda nasıl toplanır ve merkezileştirilir?
- Log saklama ve imha etme: Loglar nasıl ve nerede tutulur ve imha edilir?
- Log analizi: Loglanan olaylar nasıl yorumlanır ve sonuç olarak ne gibi eylemler gerçekleştirilir?

Yapılması gerekenleri içeren politikanın oluşturulmasından sonra yapılandırma araçlarının olması gerekir. İlk sırada hedefler, ikinci sırada altyapı tercihleridir. Gizlilik ve diğer düzenlemeler söz konusu olduğunda, hukuk biriminin de görüşleri alınmalıdır.

Politikaları tanımladıktan, sistemleri politikayı uygulamak için kurduktan ve log kaynaklarını yapılandırdıktan sonra süreklilik programının oluşturulması gerekir. Böyle bir programın özü, log verilerini periyodik olarak analiz etmek ve belirlenen istisnalara uygun müdahaleleri yapmakla ilgilidir. (Chuvakin, Detailed FISMA guidance and more)

Log verisi hacmi terabayta veya daha fazla veriye giderek zor bir sorun haline gelmektedir. NIST 800-92 arşivlenecek veriler için bir log biçimi seçmeyi (orijinal, ayrıştırılmış gibi) önerir. Ayrıca, log kayıtlarının güvenli bir şekilde saklanması ve bütünlüğünün doğrulanması için kılavuzluk eder.

NIST 800-92'nin federal hükümet dışında bir bağlayıcılığı yoktur; ancak kurumlar bu kılavuzdan fayda sağlayabilirler. Bu kılavuzla ilgili olarak elde edilen çıkarımlar şunlardır:

- Bir log yönetimi programı oluşturmak için sağlam bir temel sağlar. Diğer birçok zorunluluk araçlara odaklanır; ancak bu, log analistinin loglara bakarak tükenmesini önlemek için yararlı program yönetimi ipuçları içerir.

- Log yönetiminin gözden kaçırılmış yönleri hakkında bilgi edinmek için bu kılavuz kullanılabilir; log koruması, depolama yönetimi gibi.
- Log yönetimi alanındaki kararları haklı çıkarmanın bir yolunu sunar.
- Çoğunlukla süreçler olmak üzere daha az bitler ve baytlardan bahseder.

Sonuç olarak, NIST 800-53 ve NIST 800-92, FISMA uyumluluğu için olsun veya olmasın, çok yönlü programlar veya basitçe güvenlik ve operasyonları iyileştirmek için şunları yapmayı öğretir:

- Loglamanın zorunlu olduğu kritik sistemleri bulun.
- Loglamayı etkinleştirin ve logların standartlarda belirtilen iyi log kriterlerini karşıladığından emin olun.
- Loglama çalışmalarına farklı ekipleri katın.
- Mümkün olduğunda log yönetimini otomatikleştirin ve tüm alanlarda katı tekrarlanabilir süreçlere sahip olun. (Chuvakin, Detailed FISMA guidance and more)

9.5 5651 Sayılı Kanun

Türkiye'de 04/05/2007 tarihinde kabul edilen 5651 sayılı kanun, internet ortamında işlenen suçlarla içerik, yer ve erişim sağlayıcıları üzerinden mücadeleye ilişkin esas ve usülleri düzenlemektedir.

Bu yasa toplu internet kullanımı olan yerlerde internet trafiğinin kaydedilmesi ve kontrol altına alınması gerekliliğinden bahseder. İçerik, erişim veya yer sağlayıcı tüm kurumlar bu yasanın yükümlülüklerini yerine getirmek zorundadır. Bu sebeple ilgili yasa uyarınca, denetim loglarının ilgili kurum tarafından kaydedilmesi gerekir. Yasada loglanması gereken alanlardan bahsedilmemektedir. Bununla birlikte yasada yer sağlayıcılar için en az bir yıl ve en fazla iki yıl, erişim sağlayıcılar içinse en az altı ay ve en fazla iki yıl süreyle log tutmanın zorunluluğundan bahsedilmektedir.

10. KURUMDAKİ MEVCUT DURUM VE ÖNERİLER

Bu bölümde tez çalışması kapsamında anlatılan konularla ilgili olarak kurumdaki mevcut durum ve yapılan çalışmalar ele alınacak, diğer kurumlardaki mevcut duruma değinilecek ve bununla birlikte kuruma gerekli öneriler sunulacaktır.

Kurumda loglarla ilgili olarak hâlihazırda Bilgi Güvenliği ve Olay Yönetimi (SIEM) ürünü ile 5651 sayılı kanun kapsamında log toplayan bir yazılım yer almaktadır:

- SIEM ürünü ile toplanan loglar, saldırı tespiti için analiz edilmekte olup herhangi bir soruşturma kapsamında kullanılmamaktadır. Bu ürünün kullanımı ve gerekli çalışmaların yapılması amacıyla uzman 1 kişi, kurumda tam zamanlı olarak görev yapmaktadır.
- 5651 sayılı kanun kapsamında toplanan loglar için, çok hızlı tam metin arama yapabilen açık kaynak kodlu bir platform yer almaktadır. Bu yazılımın kullanımı ve gerekli çalışmaların yapılması amacıyla uzman 1 kişi, kurumda tam zamanlı olarak görev yapmaktadır.

10.1 Log Toplama

Kurumda SIEM ürünü ile ilgili olarak sistem ve uygulamaların envanteri çıkarılmıştır. Bu sistem ve uygulamalar ile ilgili sahiplik, kullanım amacı, kritiklik vb. bilgiler toplanmıştır. Envanterde yer alan sistem ve uygulamalar için ilgili ekiplerden logları SIEM'e göndermesi talep edilmiştir. Logları SIEM'e iletilen sistem ve uygulamalar için gerekli korelasyonlar gerçekleştirilmiş, rapor üretimi sağlanmış ve ilgili taraflara gönderilmiştir. SIEM kapsamında kurumdaki sistem ve uygulamaların loglaması istenilen olaylar şunlardır:

- Yönetici/kullanıcı hatalı erişim istekleri
- Yapılan yönetimsel değişiklikler (yönetici hakkı verilmesi, kullanıcı açılması gibi)
- Veri değişikliği, silinmesi, eklenmesi gibi durumlar
- Kullanıcının tüm erişimleri ve hareketleri

Kurumda SIEM ürünü ile uygulama logları (IIS gibi), sistem logları (denetim güvenlik, uygulama, powershell gibi), erişim logları (güvenlik duvarı, web gateway,

eposta sunucusu gibi), DNS logları (DHCP gibi) ve endpoint logları (son kullanıcılarla ilgili etkinlikler) toplanmaktadır. Kurum içi geliştirilen uygulamaların, veritabanı sistemlerinin ve Linux sunucuların logları hâlihazırda toplanmamaktadır. Veritabanı sistemleri için ilerleyen zamanlarda devreye alınacak olan bir veritabanı güvenlik duvarının anomali loglarının toplanması planlanmaktadır.

Kurumda SIEM ürünü ile log kaynaklarından syslog, dosya paylaşımı (SCP, CIFS, NFS) ve veritabanı aracılığıyla loglar toplanmaktadır. Günde ortalama 160 milyon log toplanmakta olup bu da yaklaşık olarak 30 GB'lık bir veri oluşturmaktadır.

Kurumda 5651 sayılı kanun kapsamında, barındırma hizmeti veren sunucuların, barındırma sunucularına bağlı uygulamaların ve internet kullanıcılarının loglanması gereklidir. Bunların dışında kurumun loglanmasını istediği farklı sistemler de vardır. Bunların hepsini 5651 sayılı kanun kapsamında kurumda bulunan log toplama yazılımı sağlamaktadır. Bu kapsamda SSL VPN, sunucu, Active Directory, IIS, Exchange, DHCP, DNS, web gateway, güvenlik duvarı ve IPS gibi cihaz ve sistemlerin logları toplanmaktadır. Kurum içi geliştirilen uygulamaların, veritabanı sistemlerinin ve Linux sunucuların logları burada da hâlihazırda toplanmamaktadır.

Kurumda 5651 sayılı kanun kapsamında yer alan log toplama yazılımı, log toplama işlemlerini Ankara Merkez, Gölbaşı ve 80 İl Müdürlüğü'nde yapmaktadır. İl Müdürlükleri'nde log toplama işlemi için etki alanı (domain) sunucularında ajan kullanılarak ve ilgili kayıtlar sıkıştırılarak merkezi sisteme gönderilmektedir. Tüm bilgiler yedekli tutulmakta ve bir sunucuda yaşanan kesinti sistemin çalışmasını etkilememektedir.

Kurumda 5651 sayılı kanun kapsamında yer alan log toplama yazılımı ile log kaynaklarından syslog, dosya paylaşımı ve veritabanı aracılığıyla loglar toplanmaktadır. Günde ortalama 320 milyon log toplanmakta olup bu da yaklaşık olarak 150 GB'lık bir veri oluşturmaktadır. SIEM ürününde toplanan loglar ile karşılaştırıldığında buradaki log sayısının iki katı olmasına rağmen veri boyutunun 5 katı olması, bu yazılımın logları veritabanında ve ham log verisi olarak saklamasından kaynaklanmaktadır. SIEM ürünü ise logları indekslenmiş düz metin dosyaları olarak ve ham log verisini normalleştirerek saklamaktadır.

Kurumda, zaman senkronizasyonu için bir NTP sunucusu bulunmamaktadır. Bu nedenle log kaynaklarının oluşturduğu log mesajlarında yer alan zaman bilgisi cihaz ve sistemlerde farklılık gösterebilmektedir. Windows sunucuları tarafında FSMO rolü olan etki alanı sunucusu zaman senkronizasyonu sağlamaktadır; ancak ağ cihazları için herhangi bir senkronizasyon sunucusu bulunmayıp zaman elle girilmektedir. SIEM ürünü tarafında toplanan loglara verilen alınma zamanı da elle sağlanmaktadır. 5651 kapsamında log toplayan yazılım, toplanan log mesajlarına alınma zamanını yetkili bir zaman sunucuyla senkronizasyon sağlayarak vermektedir.

10.2 Log Saklama

Kurumda SIEM ürünü ile toplanan loglar indekslenmiş düz metin dosyalarında şifrelenmiş şekilde saklanmaktadır. Toplanan loglar çevrimçi sistemde 6 ay saklanmakta ve sonrasında imha edilmektedir. SIEM'de toplanan veriler belirli bir biçimde normalleştirilerek tutulmaktadır. Bununla birlikte ham veri olmaması sebebiyle arşivleme işlemi yapılmamaktadır.

Kurumda 5651 sayılı kanun kapsamında yer alan log toplama yazılımı logları büyük veri (big data) platformunda saklamaktadır. Toplanan loglar çevrimiçi sistemde 6 ay olarak saklanmakta, daha sonra arşivleme için sıkıştırılarak belirlenen arşiv diskine taşınmaktadır. Arşivde ne kadar kalması gerektiğiyle ilgili kurumda herhangi bir politika bulunmamaktadır.

10.3 Log Analizi

Kurumda SIEM ürünü ile saldırı tespiti, önleme, iyileştirme, izleme, uyarı ve raporlama için log analizi yapılmaktadır. Kurumdaki log analizin aşamaları şunlardır:

- Toplanan log verisi filtreleme işleminden geçirilir.
- Filtrelen log verisinin ilgili alanları belirlenen bir biçimde ayrıştırılır, zenginleştirilir ve kategorize edilir.
- Normalleştirilen log verisi korelasyon motoruna gönderilir.
- Korelasyon motoru belirli kurallara göre ilişkilendirme işlemini gerçekleştirir.
- İlişkilendirme işlemi sonucunda eylem gerektiren bir olayla karşılaşılırsa belirlenen eylem tetiklenir (ilgililere uyarı gönderilmesi gibi).

Kurumda SIEM ürünü içerisinde yer alan korelasyon kuralları ile birlikte özel olarak hazırlanan korelasyon kuralları da kullanılmaktadır. Örneğin; mesai saatleri dışında tarama etkinliğinin gerçekleşmesi.

Kurumda saldırı tespitiyle ilgili olarak SIEM ürünü dışarıdan gelen bir saldırının IP adresini otomatik olarak kara listeye almakta ve belirlenen süre sonunda ilgili IP adresini kara listeden çıkartarak bağlanmasına izin vermektedir. Kritik bir saldırı gerçekleşirse belirli raporlar oluşturularak ilgili IP adresi kalıcı olarak kara listeye alınmaktadır. İçerden gelen saldırılar konusunda da yasal süreçler başlatılmaktadır.

Kurumda eyleme geçilmesi gereken kritik loglar SIEM ürünü içerisinde belirlenmiştir. Kritik olmayan fakat eylem gerektiren loglar konusu da kurumda ele alınmaktadır. Örneğin; Amazon IP adreslerine sürekli sorgu atan bir kullanıcı, günlük oluşturulan analiz raporlarıyla belirlenerek ilgili kullanıcılara bilgilendirme yapılmaktadır. Bu kritik bir durum değildir fakat belirli aralıklarla gerçekleştiği takdirde özetleme metodu ile eyleme geçilmesi gereken bir olaya dönüşmektedir.

10.4 Yapay Zekâ ile Log Analizi

Kurumda log kayıtlarının yapay zekâ ile analizi konusunda çalışmalar yapılmaktadır. Bununla ilgili olarak gerekli bir ürün, kurum envanterinde yer almaktadır. Log kayıtlarının yapay zekâ ile analizi projesindeki temel amaçlar şunlardır:

- Anormal kayıtların tespit edilmesi.
- Yapay zekâ ve makine öğrenimi algoritmaları kullanarak saldırıyı belirli bir zaman diliminde olasılıksal olarak tespit edebilen modeller kurulması.
- Kurulan modellerin akan veride çalışacak şekilde sisteme engre edilip sonuçlarının skorlanması.

Bu kapsamda sadece dışarıdan kurumun web sitesine akan ağ trafiğin ayrıntılı inceleme sonucu belirlenecek saldırı tipleri ile sınırlandırılabileceği kararlaştırılmıştır. Dışarıdan içeriye akan ağ trafiğinin de güvenlik duvarı logları üzerinden incelenmesine karar verilmiştir. Daha sonra yapay zekâ algoritmalarına girdi olarak verilecek log alanları belirlenerek loglar normalleştirilmiştir. Normalleştirilen loglar

belirli yapay zekâ algoritmalarına tabi tutularak gerekli analiz çalışması yapılmıştır. Bu konu ile bir doküman hazırlanmış olup gerekli çalışmalar devam etmektedir.

10.5 Raporlar

Kurumda SIEM ürünü ile toplanan loglar korelasyonlara tabi tutularak, her korelasyon için alarmlar/raporlar oluşturulmaktadır. Bu raporlar, Bilgi Güvenliği ekibi tarafından bildirilen kişilere iletilmekte ve ayrıca SIEM arayüzünden görülebilmektedir. Oluşturulan günlük raporlar genel olarak şunlardır:

- En Çok Oturum Açanlar
- DNS İstekleri
- DoS Etkinliği
- Etkinlik Raporu (Kim En Çok Nereye Erişmiş)
- Harici DNS'lere Erişim
- Enfekte Olmuş Dosyalar
- Tor IP'lerine Erişim
- Yetkisiz E-posta Kullanımı
- Botnet IP'lerine Erişim
- Potansiyel Taviz Verilmiş Bilgisayarlar

10.6 Geliştirilen Uygulamalardaki Loglar

Kurumda, kurum içi geliştirilen uygulamalarla ilgili olarak bir loglama standardı bulunmamaktadır. Hangi alanların loglanması gerektiği net olarak belirlenmemiştir. Kurumda yer alan .NET geliştirme altyapısı, loglama için belirli bir metot sağlamakta fakat hangi alanların loglanması gerektiği konusunda bir bilgi vermemektedir.

Kurumda bir yazılım projesi kapsamında açık kaynak kodlu bir veritabanı kulllanan loglama sistemi geliştirilmiştir. Bu sistem, ilgili yazılım projesine bağlı olan aynı yazılım altyapısı içerisindeki uygulama bileşenlerinin kuyruk mantığıyla asenkron olarak merkezi bir veritabanına loglama yapmasını sağlamaktadır. Bu yazılım projesi dışındaki geliştirilen diğer uygulamaların merkezi veritabanına loglayabilmesi için bir web servis çağrısı sağlamakta fakat bu çağrı asenkron olmaması

sebebiyle performans problemlerine yol açacağı düşünülmektedir. Bu yüzden geliştirilen diğer uygulamalar için bu loglama sistemi kullanılmamaktadır.

10.7 Politika ve Prosedürler

Kurumda loglama ve log yönetimiyle ilgili olarak hazırlanmış bir doküman bulunmamaktadır. Bununla birlikte ISO 27001 çalışmaları kapsamında A.12.4 "Kaydetme ve İzleme" bölümünde yer alan gereksinime atıfta bulunan "CSB.BGYS.PR.20 Olay Kaydetme ve İzleme Prosedürü" hazırlanmıştır. Bu prosedür, gerçekleşen olayları takip etme, kaydetme ve kanıt üretebilme eylemlerinin nasıl gerçekleştirilmesi gerektiğini tanımlamak için hazırlanmıştır. Bu prosedür kapsamında ISO 27002'de de belirtildiği şekilde loglanan olay kayıtlarının şunları içermesi gerekir:

- Kullanıcı kimlikleri,
- Sistem faaliyetleri,
- Oturum açma ve oturum kapatma gibi anahtar olayların tarihleri, saatleri ve detayları,
- Mümkünse cihaz kimliği ya da yeri ve sistem tanımlayıcısı,
- Başarılı ve reddedilmiş, sistem erişim girişimlerinin kayıtları,
- Başarılı ve reddedilmiş, veri ve diğer kaynaklara erişim girişimlerinin kayıtları,
- Ayrıcalıkların kullanımı,
- Sistem yapılandırma değişiklikleri,
- Sistem araçları ve uygulamaların kullanımı,
- Erişilen dosyalar ve erişim türü,
- Ağ adresi ve protokolleri,
- Erişim kontrol sistemi tarafından üretilen alarmlar,
- Antivirüs sistemleri ve saldırı tespit sistemleri gibi koruma sistemlerinin etkinleştirilmesi ve devre dışı bırakılması,
- Uygulamalarda kullanıcılar tarafından yürütülen işlemlerin kayıtları.

CSB.BGYS.PR.20 prosedüründe Bakanlık ve İl Müdürlükleri'nde bulunan cihazların dâhil oldukları etki alanı sunucusundan zaman senkronizasyonunu yaptığı, etki alanı sunucusun ise zaman senkronizasyonunu *time.windows.com* kaynağından

sağladığı belirtilmektedir. Bununla birlikte etki alanı sunucusuna dâhil olmayan ve hâlihazırda log üreten cihazların zaman senkronizasyonunu nasıl yapacağı konusuna dair bir bilgi verilmemektedir.

10.8 Diğer Kurumlardaki Durum

Tez çalışması kapsamında, belirlenen 7 kuruma ziyaretler yapılmış ve ilgili kurumlardaki mevcut durum tespit edilmeye çalışılmıştır. Bu noktada elde edilen veriler, kurumdaki mevcut durumun analiz edilmesine ve gerekli önerilerin sunulmasına yönelik olarak kullanılmıştır.

Diğer kurumlar ile karşılaştırıldığında log yönetimi konusunda kurumda yapılan çalışmaların orta seviyenin üstünde olduğu görülmüştür. Diğer kurumlar arasında hâlihazırda SIEM ürününe adapte olmaya çalışanlardan başka farklı projeler için birden fazla SIEM ürünü kullanılmasına yönelik çalışma yapanlara kadar çeşitli seviyede kurumlar yer almaktadır. Bu kurumlar arasında log yönetim politikası hazırlamış ama uygulamaya koymamış olanlar ile loglama konusunda herhangi bir dokümantasyonu olmayanlar da yer almaktadır.

Bu kurumların hepsi 5651 sayılı kanun kapsamında loglama gereksinimini sağlamaktadır. Bu kurumların birçoğu, kurumda hâlihazırdaki mevcut durum gibi, veritabanı ve uygulama loglarını merkezileştirme çalışmasına başlamamıştır. Bazı kurumların yazılım geliştirme birimleri ise geliştirilen uygulamaların denetim ve hata ayıklama loglarını özel olarak ele alarak açık kaynak kodlu sistemlerle merkezi loglamayı sağlayıp bu loglardan gerekli istatistikleri almaktadır. Bu kurumların çoğunda geliştirilen uygulamalarla ilgili loglama standartları ve geliştiricilere yönelik loglama önerileri bulunmamakla birlikte hassas verilerin loglanmaması adına genel olarak bir bilinç yerleşmiş durumdadır.

Bu kurumların çoğunda loglarla ilgili olarak yapay zekâ ve veri madenciliği konusunda çalışmalar yapılmamaktadır. Bu kurumların çoğunda log tutma ile ilgili olarak log kaydının ne kadar süreyle çevrimiçi ve ne kadar süreyle arşivde kalacağı hususunda bir belirli bir politika bulunmamaktadır; ancak bununla birlikte 5651 sayılı kanun kapsamında log kayıtları, 2 yıl zorunlu olarak saklanmaktadır. Bu kurumların bazıları ISO 27001'i referans alırken bazıları da ödeme kartlarıyla ilgili olarak PCI

DSS'i referans almaktadır. Bununla birlikte bu kurumlar arasında hiçbir düzenleyici kuruluşu referans almayan kurumlar da yer almaktadır.

10.9 Öneriler

Hâlihazırda kurumda geliştirilen uygulamaların ve veritabanlarının etkinlik ve denetim logları toplanmamaktadır. Bununla ilgili olarak uygulama sahipleriyle ve veritabanı yöneticileriyle görüşülerek logların toplanması, ilgili korelasyonların yapılması ve belirlenecek raporların hazırlanması gerekir.

Kurumda geliştirilen uygulamalar için loglanacak alanlar yazılım geliştiricileri, güvenlik yöneticileri ve diğer paydaşlarla birlikte değerlendirmeli ve ihtiyaç duyulacak yeni alanların loglama sistemine dâhil edilmesi sağlanmalıdır. Aynı zamanda yeni geliştirilecek her yazılım projesi için bu paydaşlardan oluşturulacak bir grup, birlikte kararlar alarak ilgili projenin loglama konusunu değerlendirmelidir. Geliştirilecek projeye göre kurumdaki güvenlik ve uyumluluğun iyileştirilmesi için kritik alanlar belirlenerek loglamaya dâhil edilmelidir.

Kurumda geliştirilen uygulamalarla ilgili diğer bir konu da hata ayıklama loglarıdır. Kurumda hali hazırda bu loglar dikkate alınmamaktadır. Hâlbuki bu loglar geliştirilen uygulamaya geri bildirim sağlayarak uygulama kodundaki sorunların tanımlanmasına ve bu sorunların hızlı bir şekilde giderilmesine yardımcı olur. Kurum, hata ayıklama loglarını değerlendirecek altyapıyı ve gerekli uygulama bileşenlerini temin ederek yazılım geliştirme süreçlerinin iyileştirilmesini ve gerekli müdahalelerin hızlı bir şekilde yapılabilmesini sağlayabilir. Günümüzde verimli ve performanslı çalışan, açık kaynak kodlu yetenekli sistemler vardır. Bu sistemler kuruma fazla maliyet oluşturulmaması adına kullanılabilir. Bu noktada kurumda açık kaynak kodlu sistemler için yetkin personelin olmaması sebebiyle danışmanlık hizmeti alınması veya ilgili personele eğitim verilmesi gereklidir. Ayrıca kurum, çeşitli araçlar kullanarak hata ayıklama loglarıyla gerekli istatistiklerin çıkartılıp yazılım yaşam döngüsü, proje ve süreç yönetimiyle ilgili olarak performansın değerlendirilmesi hususunda da fayda sağlayabilir.

Kurumdaki uygulamaların birçoğunda uygulama tarafında yapılan işlemlerin veritabanındaki hareketleri kullanıcı ile ilişkilendirilmemekte, veri değişiklikleri veritabanında yetkili olan uygulama kullanıcısı aracılığıyla gerçekleştirilmektedir. Bu

nedenle, kurumda kullanılan birçok uygulama için veri değişikliği ile kullanıcı hareketlerinin eşleştirilmesi gerçekleştirilememektedir. Buna yönelik olarak uygulama tarafında, veritabanında değişiklik yapan kullanıcıların veritabanı denetim loglarının tutulması ve ilgili logların SIEM ürününe gönderilmesi gerekir.

Kurumda geliştirilen uygulamalar için bir yazılım projesi kapsamında oluşturulan loglama sistemi, performans problemleri nedeniyle yetersiz kalmaktadır. Bununla ilgili olarak geliştirilen her bir uygulamanın kendi içerisinde loglama yapması, oluşturulan logların merkezi log sunucusuna gönderilmesi ve merkezi log sunucusunda toplanan denetim loglarının da SIEM ürününe iletilmesi, yönetilebilirlik ve verimlilik açısından faydalı olabilir.

Kurumda olayı üreten log kaynaklarında kullanılan zaman bilgisi için bir NTP sunucusu kurularak zaman senkronizasyonunun sağlanması gerekir. Böyle bir işlem, olayların doğru sıralamasının ve gerçek zamanlarının korunması için gerekli olup herhangi bir soruşturma veya analiz kapsamında oluşacak yanılgıları engelleyecektir.

Kurumda SIEM ürünü tarafından İl Müdürlükleri'nin logları toplanmamakta ve bu sebeple saldırı tespiti ve analiz işlemleri burada yer alan cihaz ve sistemler için yapılamamaktadır. Kurum, gerekli kaynağı sağlayarak buradaki logları da değerlendirmeye almalıdır.

Kurumda log yönetimi ile saldırı tespiti ve müdahale amacına yönelik olarak kurum personelinin eğitilmesi ve bu tarz işlerin dış kaynaktan kurtarılması gereklidir.

Kurumda yer alan yazılım geliştiriciler, veritabanı yöneticileri, sistem yöneticileri, güvenlik yöneticileri ve diğer paydaşlar birlikte çalışarak logların merkezileştirilmesi çalışmasının hızlandırılmasına, SIEM'in log kaynaklarından daha etkin ve verimli olarak beslenmesine, daha iyi analiz ve raporların geliştirilmesine katkı sağlamalıdır.

Kurumda loglama ve log tutma için log yönetim politikası geliştirilmelidir. 5651 sayılı kanun için loglanan verilerin arşivde ne kadar tutulacağıyla ilgili kurumda bir log tutma politikası yoktur. Bu nedenle 8 yıllık bir log verisi bile hâlâ arşivde gereksiz yer kaplamaktadır. Bu politikaların belirlenmesi ile birlikte daha yönetilebilir bir ortam oluşturulacak ve kaynakların verimli kullanılması sağlanacaktır.

SONUÇ

Loglar çok farklı biçimlerde oluşturulabilir, bununla birlikte log verilerindeki benzer paydalara bakıldığında benzerlikler görülebilir. Hangi logların neyi içerdiğini ve gerçekte neyi içermesi gerektiği arasındaki farklar, log analizini zorlaştırmaktadır.

Logların merkezileştirilmesi, çoklu sistemlerde veya dağıtık bir uygulamanın çoklu uygulama bileşenleri arasında, dağıtık log analizi için gereklidir. Syslog, kolay UDP teslimatı ve log teslimatı için modern platformlarda bulunan uygulama altyapısı nedeniyle merkezileştirme konusunda yaygın olarak kullanılır. Logların merkezileştirilmesiyle birlikte güvenlik izleme aracı, loglanan belirli olay türleriyle ilgili tüm logları neredeyse gerçek zamanlı olarak alabilir.

Artan uygulama karmaşıklığıyla beraber loglamanın önemi de artmaktır. Özellikle, uygulama davranışının analiz edilmesi ihtiyacı ve hassas bilgilerin dağıtık ve bulut tabanlı uygulamalara taşınması sebebiyle loglama, kurumlarda kontrol altına alınması gereken bir konudur.

Ağ cihazlarındaki ve işletim sistemlerindeki loglama altyapısı, uygulama seviyesindeki tehditleri tespit etmez ve araştırmaz. Bu nedenle yazılım mimarlarının loglama altyapısını kurmaları gerekir. Log incelemesinden sorumlu olan güvenlik ekibi, geliştiricilere izleme ve soruşturma etkinlikleri için kullanılabilecek yararlı ve efektif loglama yönünde yol göstermelidir.

Loglarla ilgili projeler planlanıp işletirken ve ilgili araçlar kurulup kullanırken loglamayla ilgili evrensel gerçekleri akılda tutmakta fayda vardır. Ayrıca, yapılan yaygın hatalardan kaçınmak, log yönetimini bir üst seviyeye taşıyacak ve mevcut güvenlik ve loglama altyapılarının verimliliğini artıracaktır.

Geliştirilen uygulamaların açık, özlü ve ayrıştırılabilir log mesajları oluşturduğundan emin olunmalıdır. Bu uygulamalardaki gerekli log rotasyon işlemleri, logların yönetilebilirliği açısından önemlidir. Bir log mesajına gizlilik içeren bilgilerin yerleştirilmemesine dikkat edilmelidir. Ayrıca loglamayla ilgili temel performans hususları göz önünde bulundurulmalıdır.

Loglama ve mevzuata uygunluk yakından ilişkilidir ve gelecekte de bu şekilde olmaya devam edecektir. Loglarla ilgili zorluklara rağmen çoğu kullanıcı ve sistem

eylemleri loglara kaydedilebildiğinden dolayı loglama, BT hesap verebilirliğinin birincil aracıdır. Bu nedenle loglama, birçok yönetmelik ve kanunla zorunlu kılınan mükemmel bir uyumluluk teknolojisidir.

Tez çalışmasındaki konu belirlenirken kurumda logların merkezileştirilmesiyle ilgili çalışmalar yapılmamakta olup gerekli log yönetimi araçları ve yetkin personel bulunmamaktaydı. Sadece 5651 sayılı kanun kapsamında loglama ve log toplama işlemi yapılmakta fakat buradaki loglar anormal durumları tespit etmek amacıyla ilişkilendirilmemekte, analiz edilmemekte ve izlenmemekteydi. Bu çalışma yapılırken kurumda 5651 sayılı kanun kapsamında yer alan log toplama yazılımı güncellenerek gelişmiş özelliklere sahip çok hızlı tam metin tarama yapabilen büyük veri platformunda bir yazılım getirilmiştir. Bununla birlikte logların merkezileştirilerek saldırı tespiti ve müdahalenin gerçekleştirilmesi amacıyla bir SIEM ürünü alınmış ve gerekli çalışmalara başlanmıştır.

Kurumda logların merkezileştirilmesi ve log yönetimi konusunda gerekli ürünlerin tedarik edilmesinden dolayı ticari ve açık kaynak kodlu ürünlerin tez kapsamında incelenmesine ihtiyaç duyulmamıştır. Merkezileştirme ve log yönetimi konusunda kurumdaki yazılım geliştiricilerin ve güvenlik yöneticilerinin genel olarak ihtiyaç duyacağı kavramlar, log biçimleri, çeşitli sistem ve uygulamalardan logları toplamaya yönelik mekanizmalar, log tutma ve saklama stratejileri, analiz kavramları, log madenciliği, loglama kuralları, loglama politikaları ve uyumluluk konularına değinilmiştir.

Bu çalışmada logların merkezileştirilmesi ve yönetimi konusunda farkındalık oluşturularak, kurumda yapılan çalışmalara ışık tutmak, bu çalışmalardaki eksiklerin belirlenmesine yardımcı olmak ve belirlenen eksiklerle ilgili olarak gerekli eylemlerin alınmasını sağlamak amaçlanmıştır. Bu amaçlar doğrultusunda tez çalışması gerçekleştirilmiş ve alınması gereken eylemler tespit edilerek kuruma öneri olarak sunulmuştur.

KAYNAKLAR

BEJTLICH Richard, 2004, The Tao of Network Security Monitoring: Beyond Intrusion Detection, Addison-Wesley Professional

BRAY Rory, CID Daniel, HAY Andrew, 2008, "OSSEC Host-Based Intrusion Detection Guide", Syngress Publishing

CHUVAKIN Anton, "Complete PCI DSS Log Review Procedures, Part 2", http://chuvakin.blogspot.com.tr/2010/11/complete-pci-dss-log-review-procedures_30.html, (19.03.2017)

CHUVAKIN Anton, "HIPAA Logging Howto; New attack bypasses AV protection", 2010, https://www.eventtracker.com/newsletters/hipaa-logging-howto-new-attack-bypasses-av-protection/, (21.03.2017)

CHUVAKIN Anton, "EventTracker 7 is here; Detailed FISMA guidance and more", 2010, https://www.eventtracker.com/newsletters/eventtracker-7-is-here-detailed-fisma-guidance-and-more/, (27.03.2017)

CHUVAKIN Anton, "FISMA How To; Preview EventTracker 7 and more", 2010, https://www.eventtracker.com/newsletters/fisma-how-to-preview-eventtracker-7-and-more/, (25.03.2017)

CHUVAKIN Anton, "How to Do Application Logging Right", http://arctecgroup.net/pdf/howtoapplogging.pdf, (21.02.2017)

CHUVAKIN Anton, "Log management and compliance: What's the real story", http://searchcompliance.techtarget.com/tip/Log-management-and-compliance-Whats-the-real-story, (14.03.2017)

CHUVAKIN Anton, "Logging for HIPAA Part 2; Secure auditing in Linux", 2010, https://www.eventtracker.com/newsletters/logging-for-hipaa-part-2-secure-auditing-in-linux/, (23.03.2017)

CHUVAKIN Anton, Application Logging: Worst Practices, https://www.slideshare.net/anton_chuvakin/application-logging-good-bad-ugly-beautiful-presentation, (15.02.2017)

CHUVAKIN Anton, SCHMIDT Kevin, PHILLIPS Chris, 2013, “Logging and Log Management”, Elsevier

CISCO, 2009, “Cisco Intrusion Detection Event Exchange (CIDEE) Specification”, http://www.cisco.com/c/en/us/td/docs/security/ips/specs/CIDEE_Specification.html, (15.12.2016)

“Date and Time on the Internet: Timestamps”, 2002, https://www.ietf.org/rfc/rfc3339.txt, (25.12.2016)

“Event Viewer”, 2016, https://en.wikipedia.org/wiki/Event_Viewer, (13.12.2016)

ENCYCLOPEDIA BRITANNICA, “Data Mining”, https://global.britannica.com/technology/data-mining, (06.01.2017)

FISMA, 2002, “Federal Information Security Management Act of 2002”, https://oig.federalreserve.gov/fisma.htm, (03.04.2017)

GARFINKEL Simson, 2005, “Another Look at Log Files”, http://www.csoonline.com/article/2118768/investigations-forensics/another-look-at-log-files.html, (23.02.2017)

HEWLETT PACKARD, 2016, “HPE Security ArcSight Common Event Format”, https://www.protect724.hpe.com/docs/DOC-1072, (15.12.2016)

HIPAA, 1996, “Health Insurance Portability and Accountability Act of 1996”, https://aspe.hhs.gov/report/health-insurance-portability-and-accountability-act-1996, (29.03.2017)

KENT Karen, SOUPPAYA Murigiah, 2006, “Guide to computer security log management”, NIST, Special Publication 800-92

KORFF Yanek, HOPE Paco, POTTER Bruce, 2005, “Mastering FreeBSD and OpenBSD Security”, O’Reilly Media

MERRIAM-WEBSTER, “correlation”, https://www.merriam-webster.com/dictionary/correlation, (05.01.2017)

MITRE, 2010, “Common Event Expression: Architecture Overview”, Version 0.5

NIST, 2008, “An Introductory Resource Guide for Implementing the Health Insurance Portability and Accountability Act (HIPAA) Security Rule”, http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-66r1.pdf (22.03.2017)

PCI DSS, 2016, “Requirements and Security Assessment Procedures”, Version 3.2

Ranum, Marcus, 2004, “System Logging and Log Analysis”, http://ranum.com/security/computer_security/archives/logging-notes.pdf, (13.02.2017)

SCOTT Jonathan, 2009, “Top 5 Features of Windows Event Viewer”, https://www.razorleaf.com/2009/11/event-viewer-top-5/, (17.01.2017)

“syslog”, 2016, https://en.wikipedia.org/wiki/Syslog, (13.12.2016)

“The Intrusion Detection Message Exchange Format (IDMEF)”, 2007, http://www.ietf.org/rfc/rfc4765.txt, (15.12.2016)

“The Syslog Protocol”, 2009, https://tools.ietf.org/rfc/rfc5424, (15.12.2016)

“The BSD syslog Protocol”, 2001, https://www.ietf.org/rfc/rfc3164.txt, (25.12.2016)

TROAN Erik, BROWN Preston, 2002, “Logrotate - System Administrator’s Manual”, http://linuxcommand.org/man_pages/logrotate8.html, (02.01.2017)

“utmp”, 2016, https://en.wikipedia.org/wiki/Utmp, (19.12.2016)

VERIZON, 2011, “2011 Data Breach Investigations Report”, http://www.verizonbusiness.com/resources/reports/rp_data-breach-investigations-report-2011_en_xg.pdf, (25.02.2017)

W3C, “Extended Log File Format”, http://www.w3.org/TR/WD-logfile.html, (15.12.2016)

EK-1 ÖNERİLEN LOG YÖNETİM POLİTİKASI

ÖZGEÇMİŞ

1989 yılında Ankara'da doğdu. Lise öğrenimini Çubuk Anadolu Lisesi'nde tamamladı. 2011 yılında Kocaeli Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nden mezun oldu. 2012 yılında Kocaeli Üniversitesi Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümü'nden mezun oldu. Halen Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Bölümü'nde yüksek lisans eğitimine devam etmektedir. 2012-2014 yılları arasında Vodafone Teknoloji Hizmetleri A.Ş.'de yazılım geliştirme uzmanı olarak görev yaptı. 2014 yılında Çevre ve Şehircilik Bakanlığı'nda Çevre ve Şehircilik Uzman Yardımcısı olarak göreve başladı.

ETİK KURALLARA UYGUNLUK BEYANI

Uzmanlık tezi olarak sunduğum bu çalışmayı, bilimsel ahlak ve geleneklere aykırı düşecek bir yol ve yardıma başvurmaksızın yazdığımı, yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, bunlardan her seferinde değinme yaparak yararlandığımı ve Çevre ve Şehircilik Uzmanlığı Yönetmeliğine uygun olarak hazırladığımı belirtir, bunu onurumla doğrularım. Çevre ve Şehircilik Bakanlığı tarafından belli bir zamana bağlı olmaksızın, tezimle ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara katlanacağımı bildiririm.

10.05.2017

Samet CAN