4

CYSA+

DUYGU KAÇAR

- **Cihaz Izleme**

30.07.2024

İçindekiler

1. Cihaz İzleme

1.1. Güvenlik Duvarı Günlükleri

Büyük bir miktarda güvenlik verisini çeşitli ağ cihazlarımızdan, özellikle güvenlik duvarlarımız ve izinsiz giriş tespit ve önleme sistemlerimizden elde edebiliriz. Bu yazıda ağlarımızın güvenlik duruşunu belirlemek amacıyla güvenlik duvarı günlüklerinin gözden geçirilmesine odaklanacağız.

1.2.1. Güvenlik Duvarı Günlüklerinin İncelenmesi

Güvenlik duvarı günlüklerini analiz etmeye başladığınızda, size dört tür yararlı güvenlik verisi sağladıklarını fark edeceksiniz:

1. Bağlantı Durumu : Hangi bağlantılara izin verildiği veya reddedildiği.

2. Bağlantı Noktası ve Protokol Kullanımı : Hangi bağlantı noktalarının ve protokollerin kullanıldığı.

3. Bant Genişliği Kullanımı : Belirli bağlantıların süre ve hacim bilgileri.

4. Adres Çevirme Denetim Günlüğü : Ağ adresi çevirisi (NAT) veya bağlantı noktası adresi çevirisi (PAT) gibi işlemler.

1.2.2. Güvenlik Duvarı Günlüklerinin Yapısı

Iptables (Linux) Günlükleri

Iptables, Linux tabanlı bir güvenlik duvarıdır ve CIS günlük dosyası biçimini kullanır. İşte tipik bir iptables günlüğü girdisi:

```
Jul 28 14:33:03 hostname kernel: [LOG_LEVEL] IPTables-Dropped: IN=eth0
OUT= MAC=00:0c:29:68:22:5e:00:50:56:ac:00:08:08:00 SRC=192.168.0.2
DST=192.168.0.1 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=54321 DF
PROTO=TCP SPT=12345 DPT=80 WINDOW=14600 RES=0x00 SYN
URGP=0
```

Bu girdide, zaman damgası, cihaz kimliği, ana bilgisayar adı, işlemci (genellikle çekirdek) ve günlük seviyesi gibi bilgiler bulunur. Ardından, güvenlik duvarı kuralları, ağ arayüzleri, MAC adresleri, IP adresleri, bağlantı noktaları ve paket bilgileri gibi çeşitli öznitelik-değer çiftleri yer alır.

Windows Güvenlik Duvarı Günlükleri

Windows tabanlı güvenlik duvarı, W3C Genişletilmiş Günlük Dosyası Biçimini kullanır. Tipik bir Windows güvenlik duvarı günlüğü şu şekilde görünür:

```
#Software: Microsoft HTTPAPI 2.0
#Version: 1.0
#Date: 2022-05-02 17:42:15
#Fields: date time c-ip cs-username s-ip s-port cs-method cs-uri-stem cs-uri-query sc-status cs(User-Agent)
2022-05-02 17:42:15 172.22.255.255 - 172.30.255.255 80 GET /images/picture.jpg - 200 Mozilla/5.0
```

Bu günlük formatı, yazılım ve sürüm bilgilerini, tarih ve saat gibi üstbilgileri içerir. Günlük girişinde ise istemci IP adresi, sunucu IP adresi, sunucu bağlantı noktası, HTTP yöntemi, istek yapılan kaynak, durum kodu ve kullanıcı aracısı bilgileri yer alır.

Günlük Toplama ve Analiz

Güvenlik edilmesi, ağ güvenliğinin izlenmesi ve olaylara hızlı yanıt verilmesi açısından kritiktir. Günlük toplama araçları, büyük hacimli verileri toplamak ve analiz etmek için kullanılır. Örneğin, PF Sense gibi birleştirilmiş tehdit yönetimi araçları, günlük verilerini toplamak ve analiz etmek için kullanılabilir.

Sonuç

Güvenlik duvarı günlüklerini etkin bir şekilde analiz etmek, ağ güvenliğini sağlamak için önemlidir. Günlükleri okuyabilmek ve anlamak, potansiyel tehditleri belirlemek ve ağınızı korumak için gereklidir.

1.2. Güvenlik Duvarı Yapılandırmaları

1.2.1. Geçmiş ve Günümüz: Ağ Güvenliği

Geçmişte, ağ güvenliğimizin çoğu sınır savunmasına odaklandı. Düşünce süreci şuydu: Eğer sınırlarınızı güvence altına alabilirseniz, saldırganları durdurabilirsiniz. Ancak günümüzde, sadece bu yöntem yeterli değil.

1.2.2. Benzetme: Sınır Savunma Mantığı

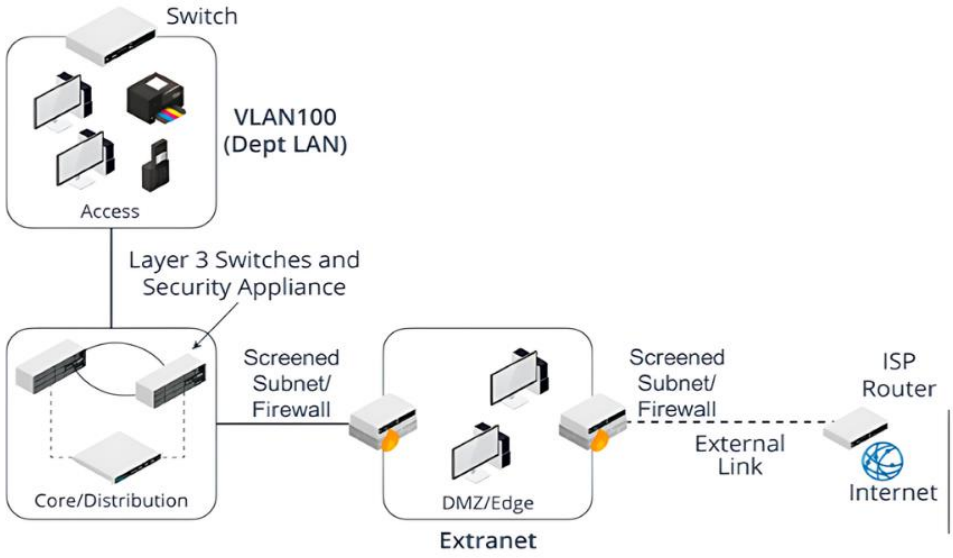
Bir benzetme kullanarak bu mantığın kusurlarını inceleyelim. Küçük, güvenlikli bir sitede yaşadığınızı varsayalım ve siteye giriş için büyük bir güvenlik kapısı kurduğunuzu düşünelim. Kapı kurulduktan sonra, kapılarınızı ve pencerelerinizi açık bırakmaya başlarsınız çünkü komşularınıza güveniyorsunuz. Ancak bir gün, ev ofisinizden dizüstü bilgisayarınızın çalındığını fark edersiniz. Güvenlik kamerasını kontrol ettiğinizde, komşunuzun çocuklarının misafirleri olduğunu ve bu misafirlerden birinin kapınızın açık olduğunu fark edip dizüstü bilgisayarınızı çaldığını görürsünüz.

Güvenlik Duvarlarının Sınırlamaları

Bu hikaye, sınır güvenliğine olan aşırı güvenin tehlikelerini vurgulamak için kullanılıyor. Güvenlik duvarları, bir kural setine dayalı olarak insanların içeri ve dışarı çıkmasına izin verir. Ancak, güvenilen kullanıcılar kötü niyetli bir faktörü içeri sokarsa, güvenlik duvarı tek başına yeterli olmaz.

Katmanlı Savunma Stratejisi

Güvenlik duvarları, katmanlı bir savunma stratejisinin önemli bir parçasıdır. Ancak, sadece bir güvenlik duvarına güvenmek yeterli değildir. Güvenlik duvarlarını, ana bilgisayar tabanlı savunmalar (örneğin, ana bilgisayar izinsiz giriş algılama ve önleme sistemleri) gibi diğer savunma yöntemleriyle birleştirmek önemlidir. Bu şekilde, tam bir koruma yelpazesine sahip olabiliriz.



Güvenlik Duvarı Konumlandırması

Güvenlik duvarlarını ağın birçok yerine koyabilirsiniz. En yaygın olanı dış cephede, ISS'niz veya yönlendiriciniz ve modeminiz ile güvenlik duvarı arasındadır. Bu güvenlik duvarı, dışarıdan gelen trafiği filtreleyerek ağınızı korur. Güvenilmeyen ağlardan gelen trafiği ekranlı alt ağlara (DMZ) yönlendirir ve ana ağa girmeden önce bir güvenlik duvarından daha geçmesini sağlar.

1.2.3. ACL'ler (Erişim Kontrol Listeleri)

ACL'ler, güvenlik duvarı kural kümeleridir ve baştan aşağı işlenir. Temel kurallar şunlardır:

1. **Özel IP Aralıklarından Gelen İstekleri Engelleme :** Örneğin, 192.168.x.x gibi yönlendirilemez IP'lerden gelen trafiği engellemelisiniz.

2. **Yalnızca Yerel Protokolleri Kullanma :** ICMP, DHCP, OSPF, SMB gibi protokoller yalnızca yerel ağda kullanılmalıdır.

3. **IPv6 Trafiğini Yönetme :** Tüm IPv6 trafiğini engellemek veya yalnızca yetkili ana bilgisayarlar ve bağlantı noktalarına izin vermek en iyisidir.

1.2.4. Güvenlik Duvarı Günlüğü Örneği

```
ip access-list extended From-DMZ
remark Responses to HTTP Requests
permit tcp 10.0.2.0 0.0.0.255 eq www any
established
permit tcp 10.0.2.0 0.0.0.255 eq 443 any established
remark ICMP and DNS
permit icmp 10.0.2.0 0.0.0.255 any echo-reply
permit udp 10.0.2.0 0.0.0.255 any eq domain
permit tcp 10.0.2.0 0.0.0.255 any eq domain
remark HTTP Requests
permit tcp 10.0.2.0 0.0.0.255 any eq www
permit tcp 10.0.2.0 0.0.0.255 any eq 443
permit udp 10.0.2.0 0.0.0.255 eq domain any
permit tcp 10.0.2.0 0.0.0.255 eq domain any
55 eq domain any
deny ip any any
```

Bir Cisco güvenlik duvarından gelen temel bir erişim listesini inceleyelim:

-ilk satırda erişim listesinin ne olduğunu söylüyor.

-ikinci satır http isteklerine verilen yanıt olduğunu açıklama satırıdır.

- üçüncü satırda:

- **TCP trafiğine izin veriliyor**: Bu kural, 10.0.2.something IP adresinden gelen tüm TCP trafiğine izin veriyor.
- **Eşit işareti (eq)**: eq ifadesi, belirli bir bağlantı noktasını işaret eder. Bu durumda, www ifadesi 80 numaralı bağlantı noktasına karşılık gelir.
- **Herhangi bir hedef IP**: Trafik, herhangi bir hedef IP adresine yönlendirilebilir.
- **Kurulu bağlantılar**: established ifadesi, yalnızca zaten kurulmuş olan bağlantılara izin verileceğini belirtir.

-dördüncü satır:

permit tcp 10.0.2.0 0.0.0.255 eq 443 any established: 10.0.2.0/24 IP aralığından gelen, port 443 (HTTPS) üzerindeki TCP trafiğine ve mevcut bağlantılara izin verir.

-Altıncı satır:

permit icmp 10.0.2.0 0.0.0.255 any echo-reply: 10.0.2.0/24 IP aralığından gelen ICMP echo-reply trafiğine izin verir.

-yedinci satır:

permit udp 10.0.2.0 0.0.0.255 any eq domain: 10.0.2.0/24 IP aralığından gelen, port 53 (DNS) üzerindeki UDP trafiğine izin verir.

-Sekizinci satır:

permit tcp 10.0.2.0 0.0.0.255 any eq domain: 10.0.2.0/24 IP aralığından gelen, port 53 (DNS) üzerindeki TCP trafiğine izin verir.

-Onuncu satır:

permit tcp 10.0.2.0 0.0.0.255 any eq www: 10.0.2.0/24 IP aralığından gelen, port 80 (HTTP) üzerindeki TCP trafiğine izin verir.

-On birinci satır:

permit tcp 10.0.2.0 0.0.0.255 any eq 443: 10.0.2.0/24 IP aralığından gelen, port 443 (HTTPS) üzerindeki TCP trafiğine izin verir.

-Onuncu satır:

permit udp 10.0.2.0 0.0.0.255 eq domain any: 10.0.2.0/24 IP aralığından gelen, port 53 (DNS) üzerindeki UDP trafiğine izin verir.

On üçüncü:

permit tcp 10.0.2.0 0.0.0.255 eq domain any: 10.0.2.0/24 IP aralığından gelen, port 53 (DNS) üzerindeki TCP trafiğine izin verir.

-On beşinci:

deny ip any any: Tüm diğer trafiği reddeder. Bu, yukarıdaki izin kurallarına uymayan herhangi bir trafiği engeller.

-İzin Beyanları : Belirli IP adreslerine (örneğin, 10.0.2.x) belirli bağlantı noktalarından (örneğin, 80 veya 443 numaralı bağlantı noktaları) gelen TCP trafiğine izin verir.

-Reddedilen Trafik : Açık izin verilmeyen herhangi bir trafiği reddeder.

Bu ACL'ler, yalnızca belirli trafiğe izin veren bir liste olarak yapılandırılır ve diğer her şeyi engeller. Bu şekilde, ağ güvenliğinizi sağlamak için güvenlik duvarı yapılandırmalarınızı optimize edebilirsiniz.

İnkar ve Düşürme(Drop and Reject)

İnkar ve düşürme iki farklı eylemdir:

- **Düşürme (Drop)**: Paket sessizce bırakılır, yani hiçbir geri bildirim gönderilmez.
- **İnkar (Reject)**: Paket açıkça reddedilir ve bir geri bildirim gönderilir.

Güvenlik duvarları, düşürme veya inkar eylemini gerçekleştirebilir. Düşürme, saldırganların güvenlik duvarı yapılandırmasını keşfetmesini zorlaştırır. Örneğin, bir saldırgan **firewalking** adlı bir teknikle güvenlik duvarını keşfetmeye çalışabilir. Bu teknik, açık portları ve arkalarındaki sunucuları tespit etmek için kullanılır.

Firewalking'i önlemek için, giden ICMP durum mesajlarını engellemek en kolay yoldur. Bu, saldırganların TTL (time to live) süresi dolan paketlerden geri bildirim almasını engeller.

Bu ACL kuralları, ağınızdan çıkan belirli trafik türlerini kontrol etmenizi sağlar ve sadece izin verilen trafiğin geçişine izin verir.

1.2.5. İnkar ve Düşürme Arasındaki Fark

Reddetme yerine düşürmeyi tercih etmenin nedeni, bir düşmanın ağınızdaki açık portları ve hizmetleri belirlemelerini zorlaştırmaktır. Ateş yürüyüşü (firewalking) adı verilen bir teknikle saldırganlar, ağınızdaki güvenlik duvarı kurallarını ve arkasındaki sistemleri belirleyebilirler. Bu teknik, TTL (Time to Live) değerlerini kullanarak güvenlik duvarını aşan trafiği analiz etmeye dayanır.

1.2.6. Firewalk ve Önlenmesi

Firewalk önlemenin en etkili yolu, giden ICMP hata mesajlarını engellemektir. Bu, saldırganların güvenlik duvarını aşan paketlerle ilgili geri bildirim almasını engeller ve böylece firewalk tekniklerini işe yaramaz hale getirir.

1.2.7. Çıkış Filtreleme

Çıkış filtrelemesi, ağınızdan ayrılan trafiği kontrol etmeyi amaçlar. Bu, kötü amaçlı yazılımların komuta ve kontrol (C2) sunucularına erişimini engelleyerek ağınızı korumanın önemli bir yoludur.

Beyaz Liste Yaklaşımı

Çıkış filtrelemesinde en iyi uygulamalardan biri, yalnızca belirli bağlantı noktalarına ve hedef adreslere izin vermektir. Örneğin, sadece 80, 443 ve 53 numaralı bağlantı noktalarına izin vererek trafiği sıkı bir şekilde kontrol edebilirsiniz.

DNS Aramalarını Kısıtlama

DNS aramalarını yalnızca güvenilir ve yetkili DNS hizmetlerine kısıtlamak, kötü niyetli DNS isteklerini engellemeye yardımcı olur. Güvenilir DNS sunucuları kullanarak, ağınızın güvenliğini artırabilirsiniz.

Kötü IP Adreslerine Erişimi Engelleme

Bilinen kötü IP adreslerine erişimi engellemek, birçok kötü amaçlı aktiviteyi önler. Bu IP'lerden gelen trafiği düşürmek, saldırganların ağınıza erişimini zorlaştırır.

İnternet Erişimini Engelleme

İnternete erişmesi gerekmeyen alt ağlardan tüm internet erişimini engellemek, özellikle ICS ve SCADA sistemleri gibi kritik altyapılar için önemlidir. Bu, hem iç hem de dış tehditlerden korunmayı sağlar.

1.2.8. Çıkış Yönetimi ve Kara Delik

Çıkış yönetimi, kötü amaçlı yazılımların C2 sunucularına erişimini tamamen ortadan kaldırmaz, ancak önemli ölçüde azaltır. Sosyal medya veya bulut tabanlı HTTPS bağlantıları üzerinden çalışan kötü amaçlı yazılımlar gibi durumlarda, çıkış filtrelemesi diğer tekniklerle desteklenmelidir.

1.2.9. Kara Delik Kavramı

Kara delik yönlendirmesi, hizmet reddi saldırılarını (DDoS) azaltmanın bir yoludur. Trafiği sessizce bırakarak, saldırganın hedefe ulaşmasını engeller. Kara delik yönlendirmesi, ACL'lere kıyasla daha az kaynak kullanır ve bu nedenle daha etkilidir.

1.2.10. Kara Delik ve Obruk (Sinkhole) Kullanımı

Kara delik ve obruk (sinkhole) teknikleri, ağ güvenliğini artırmanın yollarından biridir. Örneğin, CloudFlare veya Akamai gibi büyük ISS'ler, DDoS azaltma hizmetleri sunarak kötü amaçlı trafiği filtreleyebilir ve meşru trafiği size geri sağlayabilir. Bu, DDoS saldırıları sırasında ağınızı korumanın etkili bir yoludur.

1.3. Proxy Günlükleri

Bu derste, proxy sunucularının günlüklerini tartışacağız. Proxy sunucuları, internet ile kullanıcılar arasında bir ağ geçidi olarak işlev görür ve çeşitli işlevsellik, güvenlik ve gizlilik seviyeleri sağlar. Proxy günlükleri, kullanıcıların web etkinliklerini izlemek ve analiz etmek için kritik öneme sahiptir.

1.3.1. İleri Proxy (Forward Proxy)

İleri proxy, bir istemci ile başka bir sunucu arasındaki iletişimi arabuluculuk yapan bir sunucudur. İleri proxy'ler, iletişimleri filtreleyebilir veya değiştirebilir ve performansı artırmak için önbelleğe alma hizmetleri sağlayabilir. Temel olarak, ileri proxy, dahili ana makinenizin veya iş istasyonunuzun adına hareket eder ve HTTP isteklerini hedeflenen hedefe iletir.

Şeffaf ve Şeffaf Olmayan Proxy'ler (Transparent Proxy)

Proxy'ler şeffaf veya şeffaf olmayan olarak sınıflandırılabilir:

- Şeffaf Olmayan Proxy: İstekleri yönlendirir ve istemcinin yapılandırıldığı proxy adresi ve bağlantı noktası ile çalışır. Bu tür proxy'ler kullanıcıların varlığını bilir.

- Şeffaf Proxy: İstekleri ve yanıtları yeniden yönlendiren ancak istemciler tarafından açıkça yapılandırılmayan proxy'lerdir. Bu, kullanıcıların proxy'yi devre dışı bırakmasını engellemek için kullanılır.

1.3.2. Proxy Günlüklerinin Analizi

```
#Software: Microsoft HTTP Server API 2.0

#Version: 1.0 // the log file version as it's described by
"https://www.w3.org/TR/WD-logfile".

#Date: 2002-05-02 17:42:15 // when the first log file entry
was
recorded, which is when the entire log file was created.

#Fields: date time c-ip cs-username s-ip s-port cs-method
cs-uri-stem cs-uri-query sc-status c(User-Agent)

2002-05-02 17:42:15 172.22.255.255 - 172.30.255.255 80
GET
/images/nicture ina - 200 Mozilla/4.0+
(compatible:MSIE+5.5:+Windows+2000+Server)
```

Proxy günlükleri, kullanıcıların web etkinliklerini ve her isteğin içeriğini izleyebilir. Bu günlükler, kullanıcıların hangi web sitelerini ziyaret ettiğini ve ne kadar zaman harcadığını anlamaya yardımcı olabilir. En yaygın kullanılan günlük biçimlerinden biri, ortak günlük biçimidir. Bu günlük biçimi, web sunucuları tarafından kullanılanla aynıdır ve tarih, saat, varış noktası, istek türü, bağlantı noktası ve eylem gibi bilgileri içerir.

Analiz

Aşağıda bir Squid proxy sunucusunun erişim tablosunun bir örneği ve bu tablodan çıkarılan analizler yer almaktadır.

Date	IP	Status	Address
08.01.2020 14:30:57	10.1.0.101	TCP_DENIED/403	http://localhost:3128/squid-internal-static/icons/SN.png
08.01.2020 14:30:57	10.1.0.101	TCP_DENIED/403	http://www.515web.net/
08.01.2020 14:29:30	10.1.0.101	TCP_MISS/404	http://www.515web.net/favicon.ico
08.01.2020 14:29:30	10.1.0.101	TCP_MISS/200	http://www.515web.net/icons/ubuntu-logo.p
08.01.2020 14:29:30	10.1.0.101	TCP_MISS/200	http://www.515web.net/
08.01.2020 14:23:33	10.1.0.101	TCP_MISS/403	http://192.168.1.1/
08.01.2020 14:22:16	10.1.0.102	TCP_MISS/200	http://515web.net/icons/ubuntu-logo.png
08.01.2020 14:22:16	10.1.0.102	TCP_MISS/200	http://515web.net/
08.01.2020 14:18:28	10.1.0.102	TCP_MISS/403	http://192.168.1.1/
08.01.2020 14:17:58	10.1.0.102	TCP_MISS/403	http://192.168.1.1/

1. **Başarılı Erişimler (HTTP 200)**
    - **IP:** 10.1.0.101
        - **Adresler:**

            - http://www.515web.net/icons/ubuntu-logo.png (14:29:30)
            - http://www.515web.net/ (14:29:30)
    - **IP:** 10.1.0.102
        - **Adresler:**
            - http://515web.net/icons/ubuntu-logo.png (14:22:16)
            - http://515web.net/ (14:22:16)
2. **Başarısız Erişimler (HTTP 403)**
    - **IP:** 10.1.0.101
        - **Adresler:**
            - http://localhost:3128/squid-internal-static/icons/SN.png (14:30:57)
            - http://www.515web.net/ (14:30:57)
            - http://192.168.1.1/ (14:23:33)
    - **IP:** 10.1.0.102
        - **Adresler:**
            - http://192.168.1.1/ (14:18:28)
            - http://192.168.1.1/ (14:17:58)
3. **Diğer Hatalar (HTTP 404)**
    - **IP:** 10.1.0.101
        - **Adresler:**
            - http://www.515web.net/favicon.ico (14:29:30)

Örnek Durum İncelemesi

1. **Başarılı Erişim (HTTP 200):** 08.01.2020 14:29:30'da 10.1.0.101 IP adresi, http://www.515web.net/ ve http://www.515web.net/icons/ubuntu-logo.png adreslerine başarıyla erişmiştir.
2. **Başarısız Erişim (HTTP 403):** 08.01.2020 14:30:57'de 10.1.0.101 IP adresi, http://www.515web.net/ adresine erişmeye çalışmış ancak erişim engellenmiştir. Bu, 90 saniye önce politika güncellemesi nedeniyle gerçekleşmiştir.
3. **Diğer Hatalar (HTTP 404):** 08.01.2020 14:29:30'da 10.1.0.101 IP adresi, http://www.515web.net/favicon.ico adresine erişmeye çalışmış ancak kaynak bulunamamıştır.

Bu analiz, proxy sunucusu günlüklerinin incelenmesiyle kullanıcı aktiviteleri, erişim denemeleri ve engellemeler hakkında bilgi sağlar. Bu tür analizler, ağ güvenliğini ve kullanıcı aktivitelerini izlemek için önemlidir.

1.3.3. Ters Proxy (Reverse Proxy)

Ters proxy, sunucuları istemci isteklerinden doğrudan koruyan bir proxy sunucusudur. Ters proxy, protokol spesifik gelen trafiği yönetir. Dış internetten gelen bir istek önce ters proxy sunucusuna gider, ardından bu proxy sunucusu uygun isteği iç sunucuya iletir ve yanıtı dış istemciye geri gönderir. Bu, dış istemcinin iç sunuculara doğrudan erişimini engeller ve saldırgan trafiğinden koruma sağlar.

1.3.4. Ters Proxy Günlükleri

Ters proxy günlükleri, saldırı veya güvenlik ihlallerine yönelik göstergeleri analiz etmek için kullanılır. Bu günlükler, dış IP'lerin iç sunuculara olan isteklerini ve yanıtlarını içerir. Ters proxy, dış istemcilerden gelen tüm trafiği kontrol eder ve bu trafiği izleyerek şüpheli trendleri veya anormal sapmaları belirlemenize yardımcı olur.

Örnek Ters Proxy Günlüğü

Aşağıda, bir Squid ters proxy sunucusundan bir günlük örneği verilmiştir. Bu günlükte, dış IP'lerden gelen istekler ve iç sunuculara olan yönlendirmeler gösterilmektedir. Bu günlükler, protokol spesifik gelen trafiği ve isteklerin durum kodlarını analiz etmek için kullanılabilir.

Squid Logs					
Date	**IP**	**Status**	**Address**	**User**	**Destination**
15.10.2015 18:52:50	193.200.241.229	TCP_MISS/200	http://skami.steuff.net/	-	192.168.2.101
15.10.2015 18:52:44	83.163.239.102	TCP_MISS/304	http://test.steuff.net/	-	192.168.2.3
15.10.2015 18:52:43	83.163.239.102	TCP_MISS/304	http://test.steuff.net/	-	192.168.2.3
15.10.2015 18:52:43	83.163.239.102	TCP_MISS/304	http://test.steuff.net/	-	192.168.2.3
15.10.2015 18:52:42	83.163.239.102	TCP_MISS/304	http://test.steuff.net/	-	192.168.2.3
15.10.2015 18:52:42	83.163.239.102	TCP_MISS/304	http://test.steuff.net/	-	192.168.2.3
15.10.2015 18:52:41	83.163.239.102	TCP_MISS/404	http://test.steuff.net/favicon.ico	-	192.168.2.3
15.10.2015 18:52:41	83.163.239.102	TCP_MISS/200	http://test.steuff.net/	-	192.168.2.3
15.10.2015 18:51:06	83.163.239.102	TCP_MISS/200	http://skami.steuff.net/	-	192.168.2.101
15.10.2015 18:51:06	83.163.239.102	TCP_MISS/200	http://skami.steuff.net/	-	192.168.2.3
15.10.2015 18:51:06	83.163.239.102	TCP_MISS/200	http://skami.steuff.net/	-	192.168.2.101
15.10.2015 18:51:05	83.163.239.102	TCP_MISS/200	http://skami.steuff.net/	-	192.168.2.3
15.10.2015 18:51:05	83.163.239.102	TCP_MISS/200	http://skami.steuff.net/	-	192.168.2.101
15.10.2015 18:51:05	83.163.239.102	TCP_MISS/200	http://skami.steuff.net/	-	192.168.2.3
15.10.2015 18:51:05	83.163.239.102	TCP_MISS/200	http://skami.steuff.net/	-	192.168.2.101
15.10.2015 18:51:05	83.163.239.102	TCP_MISS/200	http://skami.steuff.net/	-	192.168.2.3
15.10.2015 18:51:04	83.163.239.102	TCP_MISS/200	http://skami.steuff.net/	-	192.168.2.101
15.10.2015 18:51:04	83.163.239.102	TCP_MISS/200	http://skami.steuff.net/	-	192.168.2.3
15.10.2015 18:51:04	83.163.239.102	TCP_MISS/200	http://skami.steuff.net/	-	192.168.2.101
15.10.2015 18:51:03	83.163.239.102	TCP_MISS/200	http://skami.steuff.net/	-	192.168.2.3
15.10.2015 18:51:03	83.163.239.102	TCP_MISS/200	http://skami.steuff.net/	-	192.168.2.101
15.10.2015 18:51:03	83.163.239.102	TCP_MISS/200	http://skami.steuff.net/	-	192.168.2.3
15.10.2015 18:51:02	83.163.239.102	TCP_MISS/200	http://skami.steuff.net/	-	192.168.2.101

Analiz

1. **Başarılı Erişimler (HTTP 200)**
    - **IP:** 193.200.241.229
        - **Adresler:**
            - http://skami.steuff.net/ (18:52:50)
        - **Hedef:** 192.168.2.101
    - **IP:** 83.163.239.102
        - **Adresler:**
            - http://skami.steuff.net/ (18:52:41, 18:51:08, 18:51:07, 18:51:07, 18:51:06, 18:51:06, 18:51:05, 18:51:04, 18:51:04, 18:51:04, 18:51:03, 18:51:03, 18:51:03)
        - **Hedef:** 192.168.2.3, 192.168.2.101
2. **Yönlendirilmiş Erişimler (HTTP 304)**
    - **IP:** 83.163.239.102
        - **Adresler:**
            - http://test.steuff.net/ (18:52:44, 18:52:43, 18:52:43, 18:52:43, 18:52:43, 18:52:42)
        - **Hedef:** 192.168.2.3

3. **Başarısız Erişimler (HTTP 404)**
    - **IP:** 83.163.239.102
        - **Adresler:**
            - http://test.steuff.net/favicon.ico (18:52:42)
        - **Hedef:** 192.168.2.3

Durum Analizi

1. **Başarılı Erişimler (HTTP 200):** 15.10.2015 tarihinde 83.163.239.102 IP adresi birçok kez http://skami.steuff.net/ adresine başarılı erişimler gerçekleştirmiştir.
2. **Yönlendirilmiş Erişimler (HTTP 304):** 15.10.2015 tarihinde 83.163.239.102 IP adresi http://test.steuff.net/ adresine birçok kez yönlendirilmiş erişim gerçekleştirmiştir.
3. **Başarısız Erişimler (HTTP 404):** 15.10.2015 tarihinde 83.163.239.102 IP adresi http://test.steuff.net/favicon.ico adresine erişim sağlamaya çalışmış ancak kaynak bulunamamıştır.

Bu analiz, ters proxy sunucusu günlüklerinin incelenmesiyle kullanıcı aktiviteleri, erişim denemeleri ve engellemeler hakkında bilgi sağlar. Bu tür analizler, ağ güvenliğini ve kullanıcı aktivitelerini izlemek için önemlidir.

1.4. Web Uygulaması Güvenlik Duvarı Günlükleri

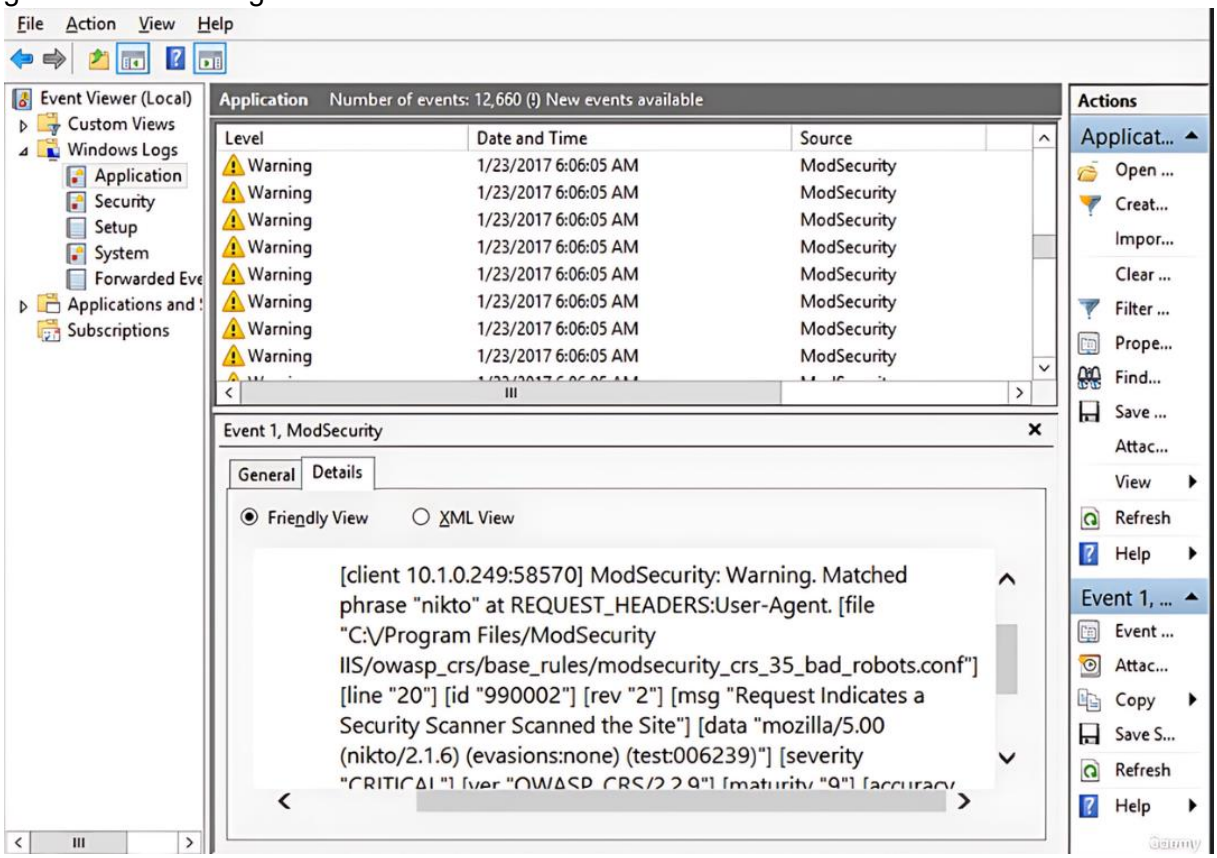
Bu derste, bir web uygulaması güvenlik duvarının (WAF) içindeki günlükler hakkında konuşacağız. Bir WAF, özellikle web sunucularında çalışan yazılımları ve arka uç veritabanlarını kod enjeksiyonu ve DoS saldırıları gibi tehditlerden korumak için tasarlanmış bir güvenlik duvarıdır.

1.4.1. WAF Nedir?

Bir WAF, web tabanlı istismarları ve güvenlik açıklarını, özellikle SQL enjeksiyonları, XML enjeksiyonları ve siteler arası betik saldırılarını önlemek için kullanılır. Standart paket filtreleme güvenlik duvarlarından farklı olarak, WAF'ler IP ve TCP/UDP katmanlarına dayalı kurallar uygulamak yerine, yanıt ve istek üst bilgilerini ayrıştırabilir ve HTML mesajlarının içeriğini görebilirler. Bu sayede, içeriklere göre filtreleme kuralları uygulayabilirler.

Örnek WAF Günlükleri

Aşağıda, bir web uygulaması güvenlik duvarının tetiklediği olayların sayısını ve detaylarını gösteren bir örnek görebilirsiniz.



Örneğin, tetiklenen bir olayı incelediğimizde, istemci, kullanılan bağlantı noktası ve mod güvenliği gibi bilgiler elde edilebilir. Bu özel kuralla tetiklenen olay, bir Nikto taraması olduğunu tespit etmiştir. Nikto, web uygulaması güvenlik açığı tarayıcısıdır ve genellikle kalem testçileri veya bilgisayar korsanları tarafından sunuculardaki güvenlik açıklarını test etmek için kullanılır.

Günlük Formatları

Web uygulaması güvenlik duvarları günlüklerini çeşitli formatlarda kaydedebilirler, ancak en yaygın olanlarından biri JSON (JavaScript Nesne Gösterimi) formatıdır. Bu günlükler, olayın zamanı, ciddiyeti, geçirilen URL parametreleri, yerel kaynak yolu, sorgu dizisi, kullanılan HTTP yöntemi (POST, GET vb.), ve kuralın bağlamı gibi birçok farklı bilgiyi içerebilir. Örneğin, bir Nikto taraması çıktısının bir kural setinin içinde referans olarak yer alması gibi.

Önemli Noktalar

Web Uygulaması Güvenlik Duvarları (WAF) hakkında bilinmesi gerekenler:

- Web sunucunuzu korumak için bir WAF kullanmalısınız.

- Özellikle SQL enjeksiyonları, XML enjeksiyonları ve siteler arası betik saldırılarına karşı koruma sağlar.

- OSI modelinin Katman 7'sinde (Uygulama Katmanı) çalışan saldırıları hedef alır.

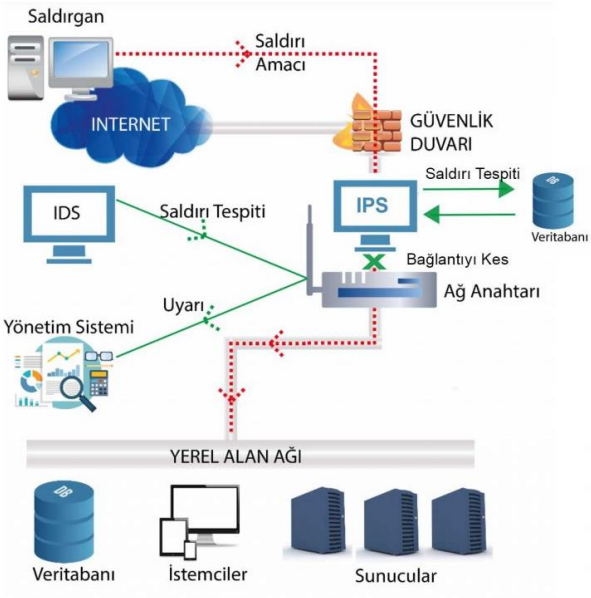
- Bir WAF, web sunucularınızdan birini hedefleyen saldırıları düzeltmek için ideal bir çözüm olabilir.

Sonuç

Web uygulaması güvenlik duvarları, web sunucularını ve arka uç veri tabanlarını çeşitli tehditlerden korumak için kritik öneme sahiptir. Günlükler, bu güvenlik duvarlarının etkinliğini izlemek ve olası tehditleri tespit etmek için önemli bir araçtır. WAF'ler, uygulama katmanında çalışan saldırıları etkili bir şekilde tespit eder ve engeller, bu nedenle web sunucularını korumak için vazgeçilmezdir.

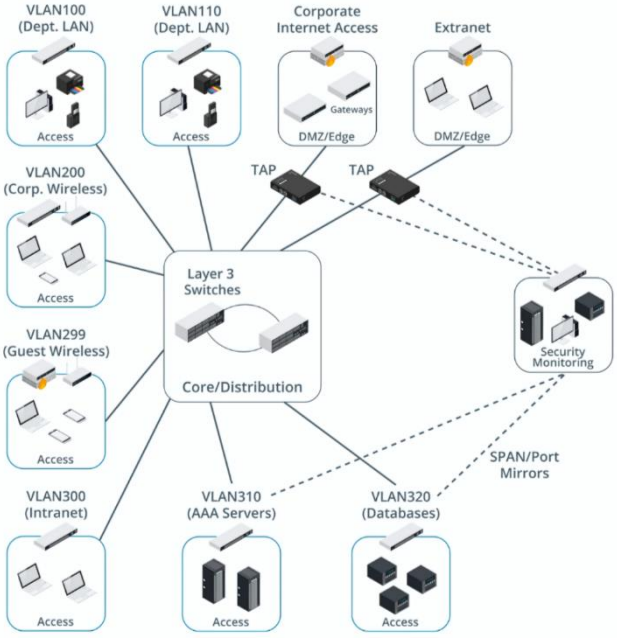
1.5. IDS ve IPS Yapılandırması

1.5.1. IDS Nedir?



Bir izinsiz giriş tespit sistemi (IDS), güvenlik altyapısını denetleyen ve izleyen bir yazılım veya donanım sistemidir. Bir IDS, ağdaki paketleri tarayan bir sensör içerir ve bu paketleri analiz motoruna gönderir. Analiz motoru, kural setlerini kullanarak trafiği inceler ve olay günlükleri, bildirimler veya uyarılar oluşturur.

IDS sensörleri genellikle güvenlik duvarının içinde veya korumaya çalıştığınız sunucunun yakınında yerleştirilir. Bu cihazlar sadece kötü amaçlı aktiviteleri tespit eder, ancak engelleyemezler. IDS'ler büyük ölçüde günlük kaydı yapmak için kullanılır.



Standart Topoloji

Kurumsal ağlarda, IDS sensörlerinin yerleştirilmesi için DMZ Edge ve extranet gibi stratejik noktalar tercih edilir. Bu noktalar, kurumsal ağın dış bağlantılarla etkileşimde bulunduğu alanlardır. IDS sensörleri, bu noktalardan gelen verileri analiz eder ve güvenlik izleme araçlarına geri bildirir.

IDS ve IPS Arasındaki Fark

- IDS (Intrusion Detection System): Kötü niyetli aktiviteleri tespit eder ve günlüğe kaydeder. Ancak, bu aktiviteleri durduramaz.

- IPS (Intrusion Prevention System): IDS'nin bir adım ötesine geçerek, saldırıları aktif olarak engelleyebilir. IPS, tıpkı IDS gibi kural setlerine dayalı olarak çalışır ve yüksek öncelikli olaylar için engelleme eylemi gerçekleştirebilir.

IPS, tüm trafiği analiz eder ve engellenmesi gereken trafiği kural setlerine dayanarak durdurur. Ayrıca, IPS yazılımları üçüncü taraf programlarda komut dosyalarını çalıştırabilir ve

sadece engelleme değil, günlüğe kaydetme ve uyarı verme gibi farklı türde eylemler gerçekleştirebilir.

IDS ve IPS Yazılımları

- Snort: Açık kaynaklı bir yazılımdır ve hem IDS hem de IPS olarak çalışabilir. Snort, kural setlerini kullanarak trafiği analiz eder ve Oinkcode adı verilen abonelik modeli ile güncellenmiş veri kural setlerine erişim sağlar. Snort, sniffer modu, yalnızca günlük modu ve aktif yanıt modu gibi farklı modlarda çalışabilir.

- Zeek (eski adıyla Bro): Unix ve Linux platformları için açık kaynaklı bir IDS'dir. Zeek, önemli olaylara etki etmek için bir komut dosyası motoru içerir ve bir IDS veya IPS olarak yapılandırılabilir.

```
@load base/utils/site
@load base/frameworks/sumstats

redef Site::local_nets += { 10.0.0.0/8 };

module MimeMetrics;

export {

    redef enum Log::ID += { LOG };

    type Info: record {
        ## Timestamp when the log line was finished and written.
        ts:         time   &log;
        ## Time interval that the log line covers.
        ts_delta:   interval &log;
        ## The mime type
        mtype:        string &log;
        ## The number of unique local hosts that fetched this mime type
        uniq_hosts: count  &log;
        ## The number of hits to the mime type
        hits:       count  &log;
        ## The total number of bytes received by this mime type
        bytes:      count  &log;
    };

    ## The frequency of logging the stats collected by this script.
    const break_interval = 5mins &redef;
}

event zeek_init() &priority=3
    {
    Log::create_stream(MimeMetrics::LOG, [$columns=Info, $path="mime_metrics"]);
    local r1: SumStats::Reducer = [$stream="mime.bytes",
                                   $apply=set(SumStats::SUM)];
    local r2: SumStats::Reducer = [$stream="mime.hits",
                                   $apply=set(SumStats::UNIQUE)];
    SumStats::create([$name="mime-metrics",
                      $epoch=break_interval,
                      $reducers=set(r1, r2),
                      $epoch_result(ts: time, key: SumStats::Key, result: SumStats::Result) =
                            {
                            local l: Info;
                            l$ts         = network_time();
                            l$ts_delta   = break_interval;
                            l$mtype      = key$str;
                            l$bytes      = double_to_count(floor(result["mime.bytes"]$sum));
                            l$hits       = result["mime.hits"]$num;
                            l$uniq_hosts = result["mime.hits"]$unique;
                            Log::write(MimeMetrics::LOG, l);
                            }]);
    }
```

- Security Onion: Linux tabanlı bir platformdur ve güvenlik izleme, olay müdahalesi ve tehdit avcılığı için kullanılır. Security Onion, Snort, Suricata, Zeek, Wireshark ve NetworkMiner gibi birçok farklı aracı bir araya getirir.

Sonuç

IDS ve IPS sistemleri, ağ güvenliği için kritik öneme sahiptir. Bu sistemler, ağdaki kötü niyetli aktiviteleri tespit etmek ve önlemek için kullanılır.

```
@load base/utils/site
@load base/frameworks/sumstats

redef Site::local_nets += { 10.0.0.0/8 };

module MimeMetrics;

export {

    redef enum Log::ID += { LOG };

    type Info: record {
        ## Timestamp when the log line was finished and written.
        ts:         time   &log;
        ## Time interval that the log line covers.
        ts_delta:   interval &log;
        ## The mime type
        mtype:        string &log;
        ## The number of unique local hosts that fetched this mime type
        uniq_hosts: count  &log;
        ## The number of hits to the mime type
        hits:       count  &log;
        ## The total number of bytes received by this mime type
        bytes:      count  &log;
    };

    ## The frequency of logging the stats collected by this script.
    const break_interval = 5mins &redef;
}

event zeek_init() &priority=3
    {
    Log::create_stream(MimeMetrics::LOG, [$columns=Info, $path="mime_metrics"]);
    local r1: SumStats::Reducer = [$stream="mime.bytes",
                                   $apply=set(SumStats::SUM)];
    local r2: SumStats::Reducer = [$stream="mime.hits",
                                   $apply=set(SumStats::UNIQUE)];
    SumStats::create([$name="mime-metrics",
                      $epoch=break_interval,
                      $reducers=set(r1, r2),
                      $epoch_result(ts: time, key: SumStats::Key, result: SumStats::Result) =
                            {
                            local l: Info;
                            l$ts         = network_time();
                            l$ts_delta   = break_interval;
                            l$mtype      = key$str;
                            l$bytes      = double_to_count(floor(result["mime.bytes"]$sum));
                            l$hits       = result["mime.hits"]$num;
                            l$uniq_hosts = result["mime.hits"]$unique;
                            Log::write(MimeMetrics::LOG, l);
                            }]);
    }
```

1.6. IDS ve IPS Günlükleri

Bir IDS veya IPS içinde bir kural her eşleştiğinde, bir günlük girişi oluşturulacak. Farklı yapılandırmanıza bağlı olarak, bu kural bir uyarı eylemini tetikleyebilir, bir bildirim

gerçekleştirebilir veya bir şeyi engelleyebilir. Çünkü yine, bir IDS iseniz, aynı zamanda bir IPS olma potansiyeline de sahipsiniz.

Şimdi, IDS'lerle ve IPS'lerle uğraşmaya başladığınızda karşılaşacağınız en büyük zorluklardan biri aşırı oturum açmadır. Çok fazla veri elde etmeye başladığınızda, kendinizi bunaltabilirsiniz ve sonunda hassasiyeti azaltırsınız. Bu, tüm bu bilgileri analiz eden analistinizin veya aracın kendisi için de geçerlidir. Bu nedenle, ayarlarını tam olarak doğru yaptığınızdan emin olmak istersiniz.

IDS ve IPS yazılımıyla ilgili bir başka harika şey de, size günlük girişlerinizin çıktısını almak için birçok farklı seçenek sunmalarıdır. Son derste Snort'tan bahsetmiştik, bu IDS araçlarından biri olarak. Şimdi Snort'un sağladığı bazı çıktı biçimlerine bir göz atalım. Beş ana biçim vardır:

1. Birleşik Çıktı (Unified output): Bu, size makine tarafından okunabilir bir ikili dosya verir. Makine tarafından okunabilir olduğu için siz bir insan olarak onu okuyamazsınız. Okumak isterseniz, bunu bir tercümana koymanız gerekir.

2. Syslog: Bu, standart Syslog formatıdır. IP adresiniz, bağlantı noktası numaranız ve eşleşen kural imzası gibi etkinlik ayrıntılarını kaydeder ve standart SIEM'lerinize veya syslog yazılımınıza konulabilir.

3. Virgülle Ayrılmış Değerler (CSV)(Comma Separated Values): Bu yaygın bir veri biçimidir ve sınırlamak için virgül karakterini kullanır. Bu, onu her türlü üçüncü taraf uygulamasına içe aktarmanıza veya normal ifadeler kullanarak ayrıştırmanıza ve Excel veya Google E-Tablolar'da bir e-tablo olarak açmanıza izin verir.

4. TCP Dökümü: Bu, her şeyi bir PCAP dosyası olarak çıkarır ve olayın altında yatan tüm paketleri yakalar. Böylece analiz edebilirsiniz.

5. SIEM Girişi: Tüm bu verileri alabilir ve merkezi bir depoya sağlayabilirsiniz. Ağdaki tüm farklı IDS'lerinizin yanı sıra diğer tüm ağ cihazlarınızın bir analist olarak bakabileceğiniz bir cihazda toplanmasını sağlar.

Tüm bu uyarılar aslında günlüklerinizde oluşturulur, bir kural olarak. Bir analist olarak, özel kurallar oluşturabilirsiniz. Varsayılan olarak, topluluk kuralları ve abonelik beslemeleri vardır, ancak bazen özel bir kural oluşturmanız gerekebilir. Bir kuralı okuyabilme ve değiştirebilme yeteneği, kendi ihtiyaçlarınıza göre özelleştirmenizi sağlar.

Her IDS kendi formatını kullanır, ancak en yaygın olanlardan biri Snort kuralı formatıdır. Snort kuralı formatı aşağıdaki gibi görünür:

- Eylem: Genellikle uyarı olarak ayarlanır, ancak günlüğe kaydetme, iletme, bırakma veya reddetme seçenekleri de vardır.

- Kaynak ve Hedef Adresler ve Portlar: Genellikle "any" olarak ayarlanır, ancak statik değerler veya değişkenler de kullanılabilir.

- Yön: Trafiğin yönünü belirtir (tek yönlü veya çift yönlü).

- Kural Seçenekleri: Birçok farklı kural seçeneği vardır (örn. MSG, flow, flags, track, reference, class-type, sid, rev).

İşte bir Snort kuralı örneği:

```
alert tep $EXTERNAL_NET any -> $HOME_NET 143(msg:"PROTOCOL-IMAP logon brute force attempt";flow:to_server,established,no_stream; content:"LOGON";fast_pattern:only; detection_filter:track by_dst, count 30,seconds 30; metadata:ruleset community, service imap;reference:url,attack.mitre.org/techniques/T1110;classtype:suspicious-logon; sid:2273; rev:12;)
```

Kural Detayları ve Açıklaması:

1. **Eylem (alert):**
    - Bu kural tetiklendiğinde bir uyarı verecek. "alert" eylemi, belirli bir trafik örüntüsü tespit edildiğinde kullanıcıya bildirim yapılmasını sağlar.
2. **Protokol (tcp):**
    - Bu kural TCP trafiği için geçerlidir. TCP, güvenilir ve bağlantı temelli bir protokoldür.
3. **Kaynak ve Hedef (src -> dst):**
    - Kaynak ağı ($EXTERNAL_NET): Bu, harici ağ anlamına gelir ve her hangi bir IP adresini kapsar.
    - Hedef ağı ($HOME_NET): Bu, iç ağ anlamına gelir ve korunacak olan IP adreslerini kapsar.
    - Kaynak ve hedef portları (any -> 143): Kaynak port herhangi bir port olabilir, hedef port ise 143'tür (IMAP protokolü için standart port).
4. **Mesaj (msg):**
    - "PROTOCOL-IMAP logon brute force attempt": Bu, kural tetiklendiğinde kullanıcıya gösterilecek uyarı mesajıdır. Bu durumda, IMAP oturum açma deneme yanılma girişimi (brute force) tespit edildiği anlamına gelir.
5. **Akış (flow):**
    - to_server,established,no_stream: Bu kural, sunucuya yönelmiş ve kurulmuş bir bağlantı üzerinden akan trafik için geçerlidir. Ayrıca, akışın mevcut bir TCP oturumu kullanıyor olması gereklidir.
6. **İçerik (content):**
    - content:"LOGON": Paket içinde "LOGON" kelimesinin aranacağını belirtir. Bu, IMAP oturum açma denemelerini tespit etmek için kullanılır.
7. **Hızlı Model (fast_pattern):**
    - fast_pattern:only: Hızlı model, belirli içerik aramalarını hızlandırmak için kullanılır. Bu durumda sadece "LOGON" içeriğine bakılır.
8. **Algılama Filtresi (detection_filter):**
    - track by_dst, count 30, seconds 30: Bu, belirli bir süre zarfında belirli sayıda olayın meydana gelmesi durumunda tetiklenecek hız sınırlayıcıdır. Burada, 30 saniye içinde aynı hedefe yönelik 30 girişim sayılırsa tetiklenecek şekilde ayarlanmıştır.

9. **Meta Veri (metadata):**
    - ruleset community, service imap: Bu, topluluk kurallarına ait olduğunu ve IMAP hizmetine yönelik olduğunu belirtir.
10. **Referans (reference):**
    - reference:url,attack.mitre.org/techniques/T1110: Bu, MITRE ATT&CK framework'ünde T1110 tekniğine atıfta bulunur. Bu teknik, brute force saldırılarını açıklamaktadır.
11. **Sınıf Türü (classtype):**
    - suspicious-logon: Bu, şüpheli oturum açma girişimleri olarak sınıflandırılır.
12. **Snort Kimliği ve Revizyonu (sid ve rev):**
    - sid:2273: Bu, Snort kural kimliğidir ve bu kural için benzersiz bir tanımlayıcıdır.
    - rev:12: Bu, kuralın 12. revizyonu olduğunu belirtir. Kurallar zamanla güncellenebilir ve her güncelleme bir revizyon numarası ile belirtilir.

1.7. Bağlantı Noktası Güvenlik Yapılandırması

Şimdiye kadar bazı ağ cihazlarından bahsettik: güvenlik duvarları, saldırı tespit sistemleri ve izinsiz giriş önleme sistemleri. Ancak bu cihazlar aynı zamanda düşman saldırıları için büyük bir hedef olabilir. Onları korumak için bağlantı noktası güvenliğini kullanıyoruz.

Bağlantı noktası güvenliği, yetkisiz uygulama hizmeti bağlantı noktalarını ana bilgisayarlarda, güvenlik duvarlarında ve yerel ağ üzerinde iletişim kurmak için kullanılan fiziksel ve uzaktan erişim bağlantı noktalarını engelleme anlamına gelir. Anahtarlar, yönlendiriciler ve güvenlik

duvarlarının tümü yazılım güvenlik açıklarına tabidir ve sunucular gibi yamalanmaları gerekir. Bu cihazları korumak, erişimi kontrol etmek ve güncellemeleri uygulamak önemlidir.

Ağ Cihazları ve Güvenlik:

- Gömülü İşletim Sistemleri: Ağ cihazlarının birçoğu gömülü işletim sistemlerinde çalışır. Örneğin, IDS veya IPS cihazları genellikle Linux üzerinde çalışır. Birçok ağ cihazının hala savunmasız çalıştığını ve eski veya güncellenmemiş Linux çekirdeklerini kullandığını görmekteyiz.

- Web Yönetim Arayüzleri: Birçok ağ cihazının web yönetim arayüzleri vardır. Bu arayüzler, siteler arası komut dosyası çalıştırma ve siteler arası istek sahteciliği gibi yazılım güvenlik açıklarına karşı savunmasız olabilir.

En İyi Uygulamalar:

1. Erişimi Kısıtlamak: ACLS kullanarak erişimi kısıtlayın. Sadece belirli ana cihazlar ve sınırlı sayıda dizüstü veya masaüstü bilgisayarın bu yönetim alanlarına girmesine izin verin.

2. İzlenen Arabirimler: Belirlenen arabirimlerin sayısını izleyin. Herhangi bir bağlantı noktasından bağlantıya izin vermeyin; sadece belirli bağlantı noktaları üzerinden erişime izin verin.

3. Uzaktan Yönetim: İnternet üzerinden uzaktan yönetim erişimini reddedin. Yönetim LAN'ınıza bir VPN üzerinden bağlanılmasını sağlayın ve doğrudan internete açılan yönetim bağlantı noktalarından kaçının.

4. Fiziksel Bağlantı Noktası Güvenliği: Fiziksel bağlantı noktası güvenliği uygulayın. Anahtarlarınızı ve anahtar donanımınızı sadece yetkili personele açık tutun. Bağlantı noktalarını yama panelinden fiziksel olarak çıkararak, izinsiz erişimi engelleyin.

5. MAC Filtreleme: MAC adresi filtrelemesi kullanarak erişim kontrol listeleri oluşturun. Ancak, MAC adreslerinin kolayca taklit edilebileceğini unutmayın.

6. Ağ Erişim Kontrolü (NAC): NAC, bir ağa erişimi doğrulayan ve yetkilendiren protokoller, politikalar ve donanımlar topluluğu için genel bir terimdir. NAC, cihaz düzeyinde erişimi kontrol ederek güvenliği artırır.

1.8. NAC Yapılandırması

Ağ erişim kontrolü (NAC), bir ağ bağlantısına izin verilmeden önce kullanıcıların kimliğini doğrulamak ve cihaz bütünlüğünü değerlendirmek için kullanılan araçlar sağlar. NAC, genellikle 802.1X olarak bilinen bir standarda dayanır. 802.1X, LAN veya kablosuz LAN üzerinden genişletilebilir kimlik doğrulama protokolü (EAP) kullanarak iletişim sağlar ve bağlantı noktası tabanlı kimlik doğrulama gerçekleştirir.

Peki, bağlantı noktası tabanlı kimlik doğrulama ve NAC nedir? Bağlantı noktası tabanlı NAC, bir anahtar veya yönlendiricinin, o bağlantı noktasına takılan cihazı doğruladıktan sonra etkinleştirilmesini sağlar. Eğer cihaz onaylanmamışsa, hiçbir işlem yapılmaz; ancak onaylanmışsa, cihaz bir süreçten geçerek ağa dahil edilir.

Bu süreç nasıl işler? Bir istemci (supplicant), EAPOL (LAN Üzerinden EAP) kullanarak ağa erişim isteğinde bulunur. Bu istek, kimlik doğrulayıcıya (genellikle bir anahtar) iletilir. Anahtar, isteği bir kimlik doğrulama sunucusuna (genellikle RADIUS veya Diameter gibi protokoller kullanılarak) gönderir. Kimlik doğrulama sunucusu, istemcinin kimlik bilgilerini denetler ve erişim izni verir veya reddeder. Eğer izin verilirse, istemci internete, diğer LAN kaynaklarına veya doğru VPN tüneline erişim sağlar. Eğer reddedilirse, istemci ya karantinaya alınır ya da tamamen erişimden mahrum bırakılır.

Bu, 802.1X'in en basit şeklidir, ancak daha geniş bir NAC çözümü, yöneticilerin politikalar veya profiller oluşturmasına olanak tanır. Bu politikalar, cihazların minimum güvenlik düzeyini karşılayıp karşılamadığını kontrol eder.

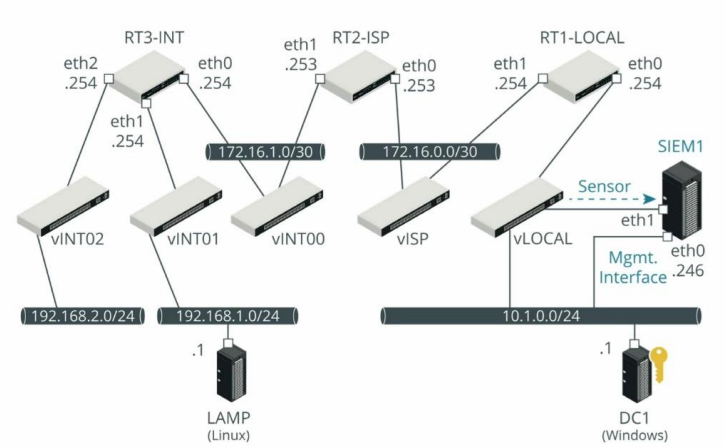
Bir NAC çözümünün bazı temel özelliklerine bakalım:

1. **Postür Değerlendirmesi:** Bu, bağlanacak cihazın sağlık politikasına uygun olup olmadığını değerlendirme sürecidir. Bu süreçte, cihazın virüs taraması, işletim sistemi yama seviyeleri, ana bilgisayar tabanlı güvenlik duvarı veya IDS gibi güvenlik bileşenleri kontrol edilir.
2. **Düzeltme:** Cihazınız belirlenen güvenlik standartlarını karşılamıyorsa, düzeltme süreci başlar. Örneğin, güncel olmayan antivirüs programınız varsa, size en son tanımları sağlayarak tekrar tarama yapmanızı isteyebiliriz.
3. **Kabul Öncesi ve Sonrası Kontroller:** Cihazlar, sağlık politikasına uyumlarına göre ağa kabul edilir veya reddedilir. Kabul sonrası süreçte, cihaz periyodik olarak kontrol edilebilir ve güvenlik standartlarını karşılamaya devam edip etmediği izlenir.

Ağ erişim kontrolünde kullanılan diğer özellikler şunlardır:

- **Zaman Tabanlı Erişim:** Bu, belirli zaman dilimlerinde ağ erişimine izin verir veya reddeder. Örneğin, bir şirketin çalışma saatleri dışındaki erişim talepleri reddedilebilir.
- **Konum Tabanlı Erişim:** Cihazın coğrafi konumunu kullanarak erişim izni verilir. Örneğin, bir kullanıcının normalde Florida'dan giriş yapması beklenirken, birdenbire İtalya'dan giriş yapması durumunda, bu durum bir uyarı olarak değerlendirilebilir.
- **Rol Tabanlı Erişim:** Bu, cihazın rolüne göre yetkilendirme sağlar. Örneğin, bir kullanıcı cihazı sunucu yönetim alt ağına bağlanmaya çalışırsa, bu erişim reddedilebilir.
- **Kural Tabanlı Erişim:** Kural tabanlı erişim, bir dizi kurala dayalı olarak erişim izni verir veya reddeder. Bu kurallar, belirli mantıksal ifadelerle yazılır ve bu kurallara uyumlu olup olmadığını değerlendirir.

1.9. Güvenlik Cihazlarının Analizi



Temel Konular:

1. SIEM ve Güvenlik Soğanı (Security Onion):

- Security Onion, bir dizi güvenlik aracını içeren bir SIEM platformudur. İçerisinde Snort, Suricata, Zeek (eski adıyla Bro) gibi IDS (Intrusion Detection System) araçları bulunur. Bu araçlar, ağdaki kötü amaçlı trafiği algılamak için kullanılır.

2. Paket Analizi:

- Paket analizi, ağ trafiğinin detaylı incelenmesi için kullanılır. Bu, Wireshark gibi araçlarla yapılabilir. Ancak manuel paket analizi zaman alıcı olabilir, bu yüzden IDS ve IPS gibi sistemler bu süreci otomatikleştirmeye yardımcı olur.

3. IDS ve IPS:

- IDS (Intrusion Detection System) ve IPS (Intrusion Prevention System) sistemleri, ağdaki kötü amaçlı trafiği algılar ve durdurur. Bu sistemler, SIEM ile entegre edilerek merkezi bir analiz ve yönetim platformu sağlar.

4. Sguil Kullanımı:

- Sguil, Security Onion içinde kullanılan bir araçtır. Gerçek zamanlı olarak güvenlik uyarılarını izleyebilir, analiz edebilir ve bu uyarılara göre aksiyon alabilirsiniz.

5. Özelleştirilmiş IDS Kuralları Oluşturma:

- IDS kurallarını manuel olarak oluşturarak, ağınızdaki belirli bir davranışa odaklanabilirsiniz. Bu dersin ilerleyen bölümlerinde, bu kuralların nasıl özelleştirileceği ve daha etkili hale getirileceği açıklanır.

6. Örnek Kural:

- Örneğin, bir kural, dış ağdan iç ağa yapılan belirli ICMP trafiğini algılamak için ayarlanabilir. Bu, dışarıdan gelen keşif saldırılarını belirlemek için kullanılır.

Pratik Uygulamalar:

- Security Onion Konfigürasyonu: Sensörler ve izleme araçları yapılandırılarak, ağ trafiğinin gerçek zamanlı olarak izlenmesi sağlanır.

- Uyarı Analizi: Security Onion içerisindeki uyarılar, önceliklerine göre analiz edilir ve olası tehditler değerlendirilir.

- IDS Kuralı Oluşturma ve Güncelleme: Nano editör kullanarak, özelleştirilmiş IDS kuralları oluşturulur ve güncellenir.

Sguil Kullanarak Güvenlik Olaylarının İzlenmesi ve Analizi

Sguil, Security Onion içinde kullanılan güçlü bir araçtır ve bu araçla gerçek zamanlı güvenlik uyarılarını izleyebilir, analiz edebilir ve bu uyarılara göre aksiyon alabilirsiniz. Bu adım adım süreçte, Sguil'in nasıl çalıştığını ve çeşitli olayları nasıl yönetebileceğinizi inceleyeceğiz.

1. Oturum Açma ve Ağ Seçimi:

- Sguil'i başlattığınızda, SIEM kullanıcı adınızı ve şifrenizi kullanarak oturum açarsınız.

- Ardından, izlemek istediğiniz ağı seçersiniz (örneğin, SIEM-eth1) ve Sguil'i başlat'a tıklarsınız.

2. Bilgilerin Görüntülenmesi:

- Sguil'in başlatılmasıyla, araç ekranında önceden kaydedilmiş örnek paket yakalamalarına dayanarak bazı bilgiler görüntülenecektir. Bu bilgiler, araçta yapılan testler sonucunda ortaya çıkan örnek olayları içerir.

3. Önceliklere Göre Uyarıların Görüntülenmesi:

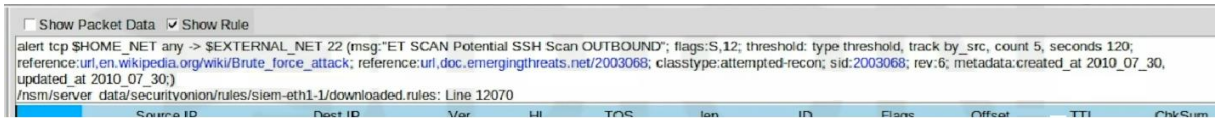
- Sol tarafta, renk kodlu olarak uyarıların öncelikleri görüntülenir. Kırmızı en yüksek önceliği, sarı ise daha düşük bir önceliği temsil eder.

- Uyarılar arasından birini seçtiğinizde, uyarıya dair daha fazla detay görebilirsiniz. Örneğin, belirli bir uyarı ID'sine tıklayarak etkinlik mesajını ve bu uyarıyı oluşturan kuralı görüntüleyebilirsiniz.

4. Kuralların Analizi:

- Bir uyarıya tıkladığınızda, uyarının sağ alt köşesindeki panelde ilgili Snort veya IDS kuralını görebilirsiniz. Bu kurallar, belirli bir ağ trafiği davranışına yanıt olarak tetiklenen uyarılar oluşturur.

- Örneğin, bir SSH taraması için yazılmış bir kural, belirli sayıda paketin belirli bir zaman diliminde belirli bir bağlantı noktasına gönderilmesini izler ve buna göre uyarı verir.



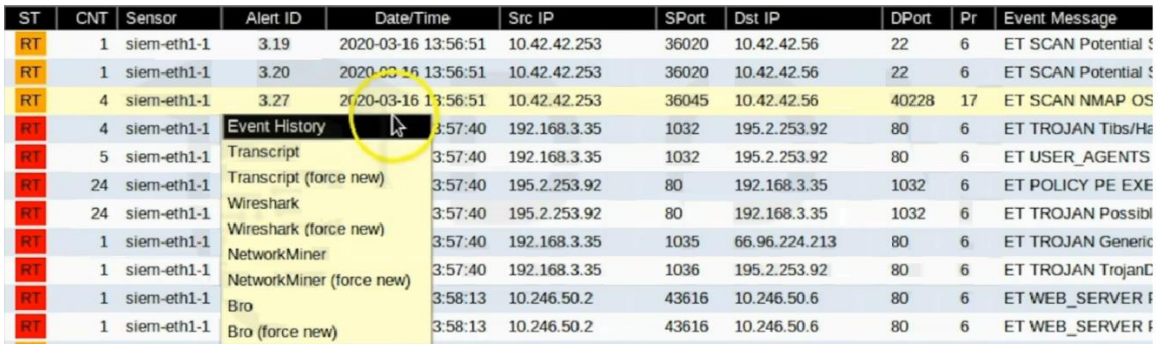
5. Uyarılar Üzerinde İşlemler:

- Uyarılar üzerinde sağ tıklayarak bağlamsal menüler açabilir ve çeşitli işlemler gerçekleştirebilirsiniz. Örneğin, bir uyarıyı seçip ilişkili olayları görüntüleyebilir, bu olaylarla ilgili daha derinlemesine analiz yapabilirsiniz.

- Ayrıca, farklı araçları kullanarak (Wireshark, NetworkMiner, Bro) ek analizler gerçekleştirebilirsiniz.

6. Veri Kaynakları ve Tehdit İstihbaratı:

- Kaynak IP adresine sağ tıklayarak, bu IP hakkında veritabanında depolanmış bilgileri görüntüleyebilir veya internet tehdit istihbaratı kaynaklarından arama yapabilirsiniz. Bu, IP adresinin geçmişte tehdit olup olmadığını belirlemenize yardımcı olur.



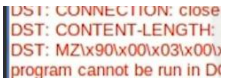
7. Etkinlik Durumu Güncelleme:

- Uyarıların durumunu güncelleyerek, bunları belirli kategorilere ayırabilirsiniz (örneğin, CAT VI: Keşif, sondalar ve taramalar).

- Durumu güncellemek, olayın incelendiğini ve ne olduğunun anlaşıldığını belirtir.

8. Örnek Olayların Analizi:

- Örneğin, bir HTML veya metin dosyası olarak görünen bir dosyanın aslında bir yürütülebilir dosya olup olmadığını belirlemek için dosya başındaki ikili koda bakabilirsiniz. Bu bilgi, dosyanın gerçek türünü anlamanıza yardımcı olur ve dosyayı daha derinlemesine analiz etmenizi sağlar. Örneğin eğer mz yazarsa yürütülebilir dosya olduğu anlamına gelir.



Sguil, bu olay akışını izleyerek güvenlik olaylarını daha etkili bir şekilde yönetmenizi ve analiz etmenizi sağlar. Security Onion ortamında Sguil'i kullanarak ağ trafiğinizi izleyebilir, güvenlik olaylarına hızlı bir şekilde yanıt verebilirsiniz.