Paket Filtreleyen Güvenlik Duvarı Anomalilerinin Saptanmasında Yapay Zeka Yaklaşımları

Hazırlayan: Ahmet KAŞİF
Danışman: Dr.Öğr.Üyesi Cengiz TOĞAY

İçerik

- Giriş
- Güvenlik Duvarları
- Güvenlik Duvarı
  - Anomalileri
  - Anomali Tespit Yöntemleri
- Mantıksal Programlama ile Anomali Tespiti
- Sonuçlar
- Tartışma

Giriş

Bilgisayar ağları, birbiri ile haberleştirilen cihazlardan oluşan hiyerarşik bir yapıdır. Başlangıçta, şimdiki yerel ağlara karşılık gelen (LAN) ufak ağlar kullanılmakta idiyse de, sonradan bu ağlar birleşerek daha büyük ağları oluşturmuşlardır.

Bu şekilde cihazların birbirleri ile çok uzak mesafeden haberleşebilir hale gelmeleri, bu cihazlar üzerinden yapılan işlemlere yönelik saldırıların da boyut değiştirmesine sebep olmuştur ve bilgisayar ağları güvenliği konusu önem kazanmıştır.

Güvenlik Duvarları

- Yerel ağı, ağın dışında gelen saldırılara karşı koruma
- OSI katmanları içerisinde
  - ağ (network)
  - taşıma (transport)
  - uygulama (application)
- Güvenlik duvarlarında bulunması gereken özellikler:
  1. Yerel ağ ile dış dünya arasındaki giden ve gelen trafiğin tamamını gözlemleyebilmek
  2. Sadece yetkilendirilmiş bağlantılara izin vermek
  3. Doğrudan kendisine yapılacak saldırılara bağışık olmak

Güvenlik Duvarları

Güvenlik Duvarı Türleri

1. Paket Filtreleyen (Packet Filtering)
2. Durumlu (State Based)
3. Uygulama (Application)

Order	Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
1	tcp	192.152.1.*	Any	128.172.26.*	80	ALLOW
2	tcp	192.152.1.72	Any	128.172.*.*	80	DENY
3	tcp	192.152.1.*	Any	128.172.26.2	80	ALLOW
4	tcp	Any	Any	Any	80	DENY
5	tcp	151.*.*.*	Any	108.56.56.1	53	DENY
6	tcp	151.126.*.*	Any	108.56.56.1	53	ALLOW
7	tcp	151.51.*.*	Any	108.56.*.*	53	ALLOW
8	tcp	Any	Any	Any	Any	DENY
9	udp	216.22.14.*	Any	124.24.*.*	53	ALLOW
10	udp	216.22.14.1	Any	124.24.*.*	53	DENY
11	udp	216.22.14.1	Any	124.*.*.*	53	DENY
12	udp	25.12.*.*	Any	124.*.*.*	Any	ALLOW
13	udp	Any	Any	Any	Any	DENY

Güvenlik Duvarları (Paket Filtreleyen)

- Ağdaki gelen ve giden tüm paketleri, sıralı bir kural listesiyle filtreler.
- Gelen ve giden paketler, kural listesindeki tüm kurallarla tek tek eşleştirilir, eşleşen kuralın belirttiği aksiyon alınır.
- Düşük maliyet ve kaynak kullanımına sahiptirler.
- Problem:
  - Sürekli büyüyen kural listeleri, listenin yönetimini zorlaştırır ve zaafiyetleri oluşmasına zemin hazırlar.
  - Büyüyen kural listeleri, paketlerin eşleşmesi gereken kuralların da artmasına ve ağ performansının düşmesine neden olur.
  - Sadece ağ veya taşıma katmanında koruma sağlayabilirler.
  - DOS saldırılarına karşı bağışıklıkları azdır.

Güvenlik Duvarları (Durumlu)

- Durumlu güvenlik duvarları, paketlerin geldikleri ve gittikleri adresleri bir önbellek yapısında saklar.
- Oluşturulan bağlantı vasıtasıyla güvenlik duvarını aşıp sisteme giriş yapan ilk paketin bilgileri önbellekte kaydedilir. Aynı bağlantı üzerinden gelen diğer paketler, önbellekten kabul edilerek doğrudan sisteme giriş yapar.
- Problem:
  - Önbelleğin yetersiz kalması. DOS gibi saldırılara karşı etkisi düşüktür.

Güvenlik Duvarları (Uygulama katmanı)

- Uygulama katmanı güvenlik duvarları, korunacak uygulamaya özel geliştirilen çözümlerdir.
  - Web Application Firewall
  - SIP Firewall
- Genel amaç güvenlik duvarlarına göre daha yüksek seviye koruma sağlarlar, fakat bakım ve geliştirme maliyetleri çok yüksektir.

Güvenlik Duvarı Anomalileri

- Paket filtreleyen güvenlik duvarlarında, anomaliler beş türde incelenmektedir.
  1. Gölgeleme (Shadowing)
  2. Gereksizlik (Redundancy)
  3. Kesişim (Correlation)
  4. Genelleştirme (Generalization)
  5. İlişiksizlik (Irrelevancy)

Güvenlik Duvarı Anomalileri (Gölgeleme)

- Gölgeleme anomalisi, kural listesindeki iki kuraldan öncelikli olanın diğerini gölgeleyerek, çalışmasına mani olması manasına gelir.
- Aşağıdaki durumlarda iki kural arasında gölgeleme anomalisi oluşur.
  - R_x ve R_γ güvenlik duvarında bulunan iki kural olup, R_x listede işlem önceliğine sahiptir.
  - Tüm filtreleme alanları için, $R_x \supseteq R_\gamma$ olmalıdır.
  - R_x ve R_γ kurallarının aksiyonları farklı olmalıdır.

5	tcp	151.*.*.*	Any	108.56.56.1	53	DENY
6	tcp	151.126.*.*	Any	108.56.56.1	53	ALLOW

Güvenlik Duvarı Anomalileri (Gereksizlik)

- Gereksizlik anomalisi, kural listesinde aynı paket grubuna aynı aksiyonu uygulayan iki kural bulunduğu durumlarda oluşur. Listede öncelikli olan kural her zaman işletileceğinden diğer kural gereksizdir.
- Aşağıdaki durumlarda iki kural arasında gereksizlik anomalisi oluşur.
  - R_x ve R_γ güvenlik duvarında bulunan iki kural olup, R_x listede işlem önceliğine sahiptir.
  - Aşağıdaki koşulların bir tanesi sağlanmalıdır.
    - Tüm filtreleme alanları için, $R_x \supseteq R_\gamma$ olmalıdır ve R_x ve R_γ kurallarının aksiyonları aynı olmalıdır.
    - Aşağıdaki koşulların tamamı sağlanmalıdır.
      - Tüm filtreleme alanları için, $R_\gamma \supseteq R_x$ olmalıdır.
      - Listede R_γ'den öncelikli fakat R_x'ten öncelikli olmayan, üçüncü bir kural $R_\square$ olmamalıdır. Var ise, $R_\square \supseteq R_x$ olmamalı ve $R_\square$ ve R_x'in aksiyonları farklı olmalıdır.

Güvenlik Duvarı Anomalileri (Gereksizlik)

10	udp	216 . 22 . 14 . 1	Any	124 . 24 . * . *	53	DENY
11	udp	216 . 22 . 14 . 1	Any	124 . * . * . *	53	DENY

Güvenlik Duvarı Anomalileri (Kesişim)

- Kesişim anomalisi, iki kuralın filtreleme alanları birbiriyle kesiştiğinde ve bu kurallar farklı aksiyonlara sahip olduğunda oluşur.
- Aşağıdaki durumlarda iki kural arasında kesişim anomalisi oluşur.
  - R_x ve R_Y'nin filtreleme alanları birbiriyle kesişmelidir, her kural en az 1 filtreleme alanında diğer kuralın filtreleme alanını kapsamalıdır.
  - R_x ve R_Y kurallarının aksiyonları farklı olmalıdır.

9	udp	216.22.14.*	Any	124.24.*.*	53	ALLOW
11	udp	216.22.14.1	Any	124.*.*.*	53	DENY

Güvenlik Duvarı Anomalileri (Genelleştirme)

- Genelleştirme anomalisi, öncelikli bir kuralın, önceliksiz bir kural tarafından kapsanması durumunda oluşur. Bu durumdaki paketler
- Aşağıdaki durumlarda iki kural arasında genelleştirme anomalisi oluşur.
  - R_x ve R_γ güvenlik duvarında bulunan iki kural olup, R_x listede işlem önceliğine sahiptir.
  - Tüm filtreleme alanları için, $R_\gamma \supseteq R_x$ olmalıdır.
  - R_x ve R_γ kurallarının aksiyonları farklı olmalıdır.

1	tcp	192 . 152 . 1 . *	Any	128 . 172 . 26 . *	80	ALLOW
4	tcp	Any	Any	Any	80	DENY

Güvenlik Duvarı Anomali Tespit Yöntemleri

- Güvenlik duvarlarında en çok kullanılan anomali tespit algoritmaları
  - Statik analiz yöntemleri
    - Durum Diyagramları
    - İkili Karar Ağaçları
  - Paket simülasyon yöntemi

Durum Diyagramları ile Anomali Tespiti

- Durum diyagramları ikili kural karşılaştırmalarında çoklukla kullanılan bir yöntemdir.
- Kurallar, sırasıyla 6 alanda kapsama, eşitlik ve eşitsizlik durumlarına göre bir sonraki duruma geçerler. Son durumda aralarındaki ilişki belirlenmiş olur.
- Analizin yapılması için güvenlik duvarı konfigürasyon dosyası yeterlidir, bu sebeple çevrimdışı analize imkan tanır.

İkili Karar Ağaçları ile Anomali Tespiti

- Kuralların içerdiği tüm alanlar ikili formatta tekrar kodlanarak ifade edilir, bu sayede bit seviyesinde karşılaştırma yapma imkanı sağlanarak daha hızlı ve kesin sonuç alınması hedeflenir.
- Kurallar, yeni kodlamaya göre özelleştirilmiş bir durum diyagramı yardımıyla karşılaştırılır.

Paket Simülasyonu ile Anomali Tespiti

- Güvenlik duvarının birçok paket ile sınanması ile çalışmayan kuralların tespit edilmesini sağlar.
- Tüm olası paketlerin sınanması yüksek işlem gücü gerektiren bir işlemdir.

Önerilen Yaklaşım

- Literatürde paket filtreleme tabanlı güvenlik duvarı anomalilerinin saptanması yoğunlukla çalışılan bir konudur.
- Bu çalışmada Mantık Programlama tabanlı bir programlama dili olan Prolog kullanılarak paket filtreleme tabanlı güvenlik duvarı kural listelerinde kural anomalilerinin saptanması üzerine çalışılmıştır.
- Çalışma kapsamında, Java Servlet tabanlı ve istemci-sunucu yaklaşımını kullanan bir web uygulaması geliştirilmiş, tüm güvenlik duvarı kuralları SQL tabanlı bir veritabanında saklanmış ve çalışma zamanında uygulama tarafından çekilerek işlenmiştir.

Anomali Tespit Uygulaması

Rule List

Add Rule

Order	Protocol	Source IP	Source Port	Destination IP	Destination Port	Action	Edit	Delete
1	tcp	192 . 152 . 1 . *	Any	128 . 172 . 26 . *	80	ALLOW	Edit	Delete
2	tcp	192 . 152 . 1 . 72	Any	128 . 172 . * . *	80	DENY	Edit	Delete
3	tcp	192 . 152 . 1 . *	Any	128 . 172 . 26 . 2	80	ALLOW	Edit	Delete
4	tcp	Any	Any	Any	80	DENY	Edit	Delete
5	tcp	151 . * . * . *	Any	108 . 56 . 56 . 1	53	DENY	Edit	Delete
6	tcp	151 . 126 . * . *	Any	108 . 56 . 56 . 1	53	ALLOW	Edit	Delete
7	tcp	151 . 51 . * . *	Any	108 . 56 . * . *	53	ALLOW	Edit	Delete
8	tcp	Any	Any	Any	Any	DENY	Edit	Delete
9	udp	216 . 22 . 14 . *	Any	124 . 24 . * . *	53	ALLOW	Edit	Delete
10	udp	216 . 22 . 14 . 1	Any	124 . 24 . * . *	53	DENY	Edit	Delete
11	udp	216 . 22 . 14 . 1	Any	124 . * . * . *	53	DENY	Edit	Delete
12	udp	25 . 12 . * . *	Any	124 . * . * . *	Any	ALLOW	Edit	Delete
13	udp	Any	Any	Any	Any	DENY	Edit	Delete

Anomali Tespit Uygulaması

Anomalies in Firewall

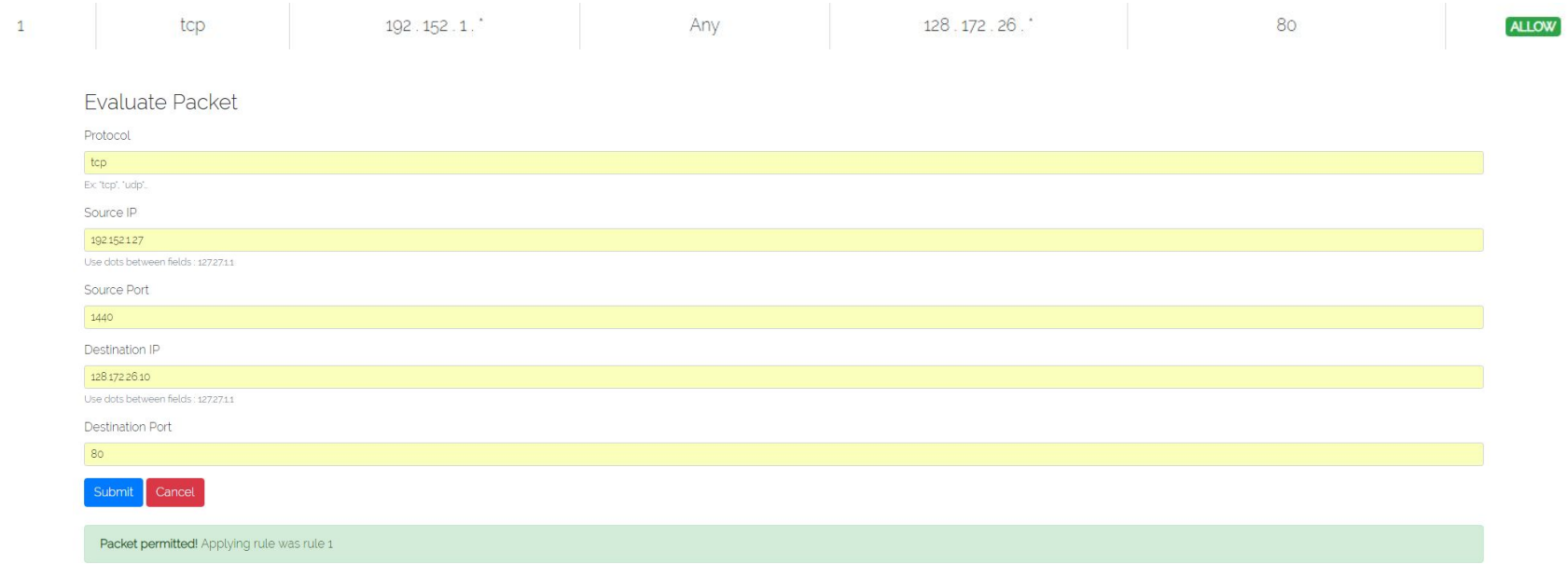
Show Policy Tree

#	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}	R_{12}	R_{13}
R_1	-	C	R	G	-	-	-	G	-	-	-	-	-
R_2	-	-	C	R	-	-	-	R	-	-	-	-	-
R_3	-	-	-	G	-	-	-	G	-	-	-	-	-
R_4	-	-	-	-	-	-	-	R	-	-	-	-	-
R_5	-	-	-	-	-	S	C	R	-	-	-	-	-
R_6	-	-	-	-	-	-	-	G	-	-	-	-	-
R_7	-	-	-	-	-	-	-	G	-	-	-	-	-
R_8	-	-	-	-	-	-	-	-	-	-	-	-	-
R_9	-	-	-	-	-	-	-	-	-	S	C	-	G
R_{10}	-	-	-	-	-	-	-	-	-	-	R	-	R
R_{11}	-	-	-	-	-	-	-	-	-	-	-	-	R
R_{12}	-	-	-	-	-	-	-	-	-	-	-	-	G
R_{13}	-	-	-	-	-	-	-	-	-	-	-	-	-

S: Shadowing, R: Redundancy, G: Generalization, C: Correlation, - : No Anomaly

Anomali Tespit Uygulaması

- Arayüz üzerinden, paket simülasyon motoruna erişim sağlanabilmekte, bu sayede herhangi bir kuralın çalışıp çalışmadığı kontrol edilebilmektedir.



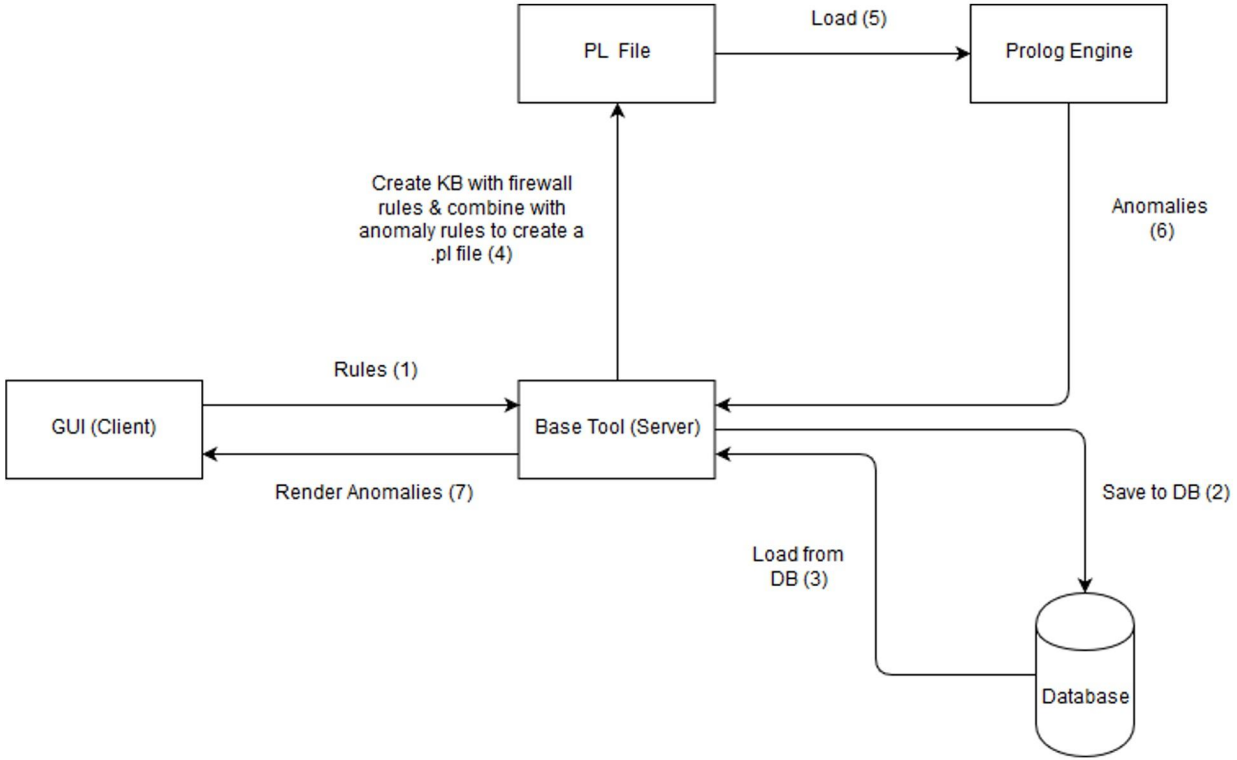
Mantık Programlama ile Anomali Tespiti

- Mevcut bilgi kümesi ve kurallar yardımıyla yeni bilgi türetilmesinde kullanılan mantık programlamanın, kural tabanlı paket filtreleyen güvenlik duvarlarında anomali tespiti yapmada yararlı olabileceği düşünülmüştür.
- Kural anomalilerinin tespiti için, güvenlik duvarı politika kuralları bilgi kümesinde yeniden tanımlanmıştır.
- Kural anomalileri tespit algoritmaları, mantık programlama yapısındaki “clause” ve “rule” yapıları kullanılarak oluşturulan “predicate”’lar ile tanımlanmıştır.

Mantık Programlama ile Anomali Tespiti

```
policyrule(1, tcp, source([192, 152, 1, 256]), 256, destination([128, 172, 26, 256]), 80, allow).
policyrule(2, tcp, source([192, 152, 1, 72]), 256, destination([128, 172, 256, 256]), 80, deny).
policyrule(3, tcp, source([192, 152, 1, 256]), 256, destination([128, 172, 26, 2]), 80, allow).
policyrule(4, tcp, source([256, 256, 256, 256]), 256, destination([256, 256, 256, 256]), 80, deny).
policyrule(5, tcp, source([151, 256, 256, 256]), 256, destination([108, 56, 56, 1]), 53, deny).
policyrule(6, tcp, source([151, 126, 256, 256]), 256, destination([108, 56, 56, 1]), 53, allow).
policyrule(7, tcp, source([151, 51, 256, 256]), 256, destination([108, 56, 256, 256]), 53, allow).
policyrule(8, tcp, source([256, 256, 256, 256]), 256, destination([256, 256, 256, 256]), 256, deny).
policyrule(9, udp, source([216, 22, 14, 256]), 256, destination([124, 24, 256, 256]), 53, allow).
policyrule(10, udp, source([216, 22, 14, 1]), 256, destination([124, 24, 256, 256]), 53, deny).
policyrule(11, udp, source([216, 22, 14, 1]), 256, destination([124, 256, 256, 256]), 53, deny).
policyrule(12, udp, source([25, 12, 256, 256]), 256, destination([124, 256, 256, 256]), 256, allow).
policyrule(13, udp, source([256, 256, 256, 256]), 256, destination([256, 256, 256, 256]), 256, deny).
```

Mantık Programlama ile Anomali Tespiti

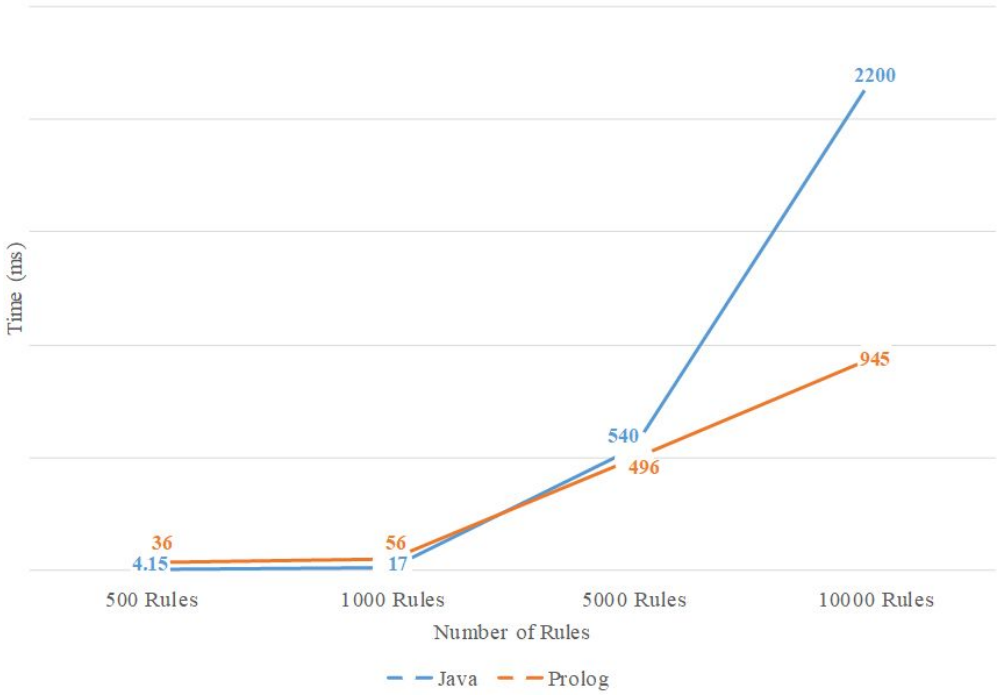


Güvenlik Duvarı Oluşturucu Uygulaması

- Uygulamanın ihtiyaç duyduğu yüksek kural sayısına sahip güvenlik duvarı konfigürasyonlarını elde etmek için, Java dilinde kural oluşturucu uygulaması geliştirilmiştir.
- Kural oluşturucu yardımcı uygulaması, istenen sayıda kural ve istenen oranda anomali içeren paket filtreleme tabanlı güvenlik duvarı konfigürasyonlarının oluşturulmasını sağlayan Java tabanlı bir masaüstü uygulamasıdır.

Sonuçlar

- 1000 den fazla kurallar için performans sağladığımız görülmektedir.
- Mantık programlama ile daha fazla çıkarsama yapma imkanına kavuşuldu.
- Güvenlik Duvarlarından elde edilen listeden doğrudan kural üretilmesi mümkün olmuştur.



Sonuçlar

- Daha performanslı bir çözüm sunuldu.
- Mantık programlamanın, kural tabanlı bilgi çıkarımı gerektiren güvenlik duvarı anomali saptama alanında imperatif programlama yaklaşımlarına nazaran çok daha yorumlanabilir ve performanslı yazılımların geliştirilmesine imkan sağladığı görülmektedir.

Teşekkürler