1. Proje Tanımı ve Amacı

Sosyal mühendislik, her bireyin ve modern organizasyonun karşılaştığı en tehlikeli tehditlerden biridir. Kimlik avı, iyi bilinen, bilgisayar tabanlı bir sosyal mühendislik tekniğidir. Saldırganlar, büyük şirketleri hedef almak için gizli e-posta adreslerini bir silah olarak kullanır. Her gün çok sayıda oltalama e-postası alındığından şirketler bunların hepsini tespit edemiyor. Bu nedenle, kimlik avına karşı savunmak için yeni tekniklere ve güvenlik önlemlerine ihtiyaç vardır.

Python makine öğrenimi kitaplıkları kullanılarak, kimlik avı girişimlerini tespit etmek için çözümler sunacaktır.

- Sosyal mühendisliğe genel bakış
- Sosyal Mühendislik Katılım Çerçevesi (SMKÇ)
- Sosyal mühendislik sızma testi adımları
- Makine öğrenimi kullanarak gerçek zamanlı bir phishing tespiti
    - Lojistik regresyon ile kimlik avı tespiti
    - Karar ağaçlarıyla kimlik avı algılama

Teknik Gereksinimler:

- scikit-learn Python (≥ 2.7 veya ≥ 3.3)
- NumPy (≥ 1.8.2)
- NLTK

1.1 Sosyal Mühendisliğe Genel Bakış

Sosyal mühendislik, bir kişinin daha sonra bir sistemi tehlikeye atmak için kullanılabilecek yararlı ve hassas bilgiler elde etmek için psikolojik manipülasyonudur. Başka bir deyişle, suçlular, insan davranışından yararlanarak insanlardan gizli bilgiler elde etmek için sosyal mühendisliği kullanır.

1.2 Sosyal Mühendislik Katılım Çerçevesi (SMKÇ)

Sosyal Mühendislik Katılım Çerçevesi (SEEF), Dominique C. Brack ve Alexander Bahmram tarafından geliştirilen bir frameworktur. Bilgi güvenliği ve sosyal mühendisliğe karşı savunma konusunda yılların deneyimini özetler. Çerçevenin paydaşları kuruluşlar, hükümetler ve bireylerdir (kişiler). Sosyal mühendislik katılım yönetimi üç adımdan geçer:

- Katılım öncesi süreç: Sosyal mühendislik operasyonunun hazırlanması
- Katılım süreci sırasında: Katılım gerçekleşir
- Katılım sonrası süreç: Bir raporun teslim edilmesi

Suçlular tarafından kullanılan birçok sosyal mühendislik tekniği vardır:

1.2.1 Yemleme

Yemleme saldırıları, kurbanın açgözlülüğünü veya merakını kışkırtmak için sahte bir söz kullanır. Saldırganlar, kullanıcıları kişisel bilgilerini çalan veya sistemlerine kötü amaçlı yazılım bulaştıran bir tuzağa çekerler.

Tuzağın en çok eleştirilen biçimi, kötü amaçlı yazılımları dağıtmak için fiziksel medyayı kullanır. Örneğin, saldırganlar yemi (genellikle kötü amaçlı yazılım bulaşmış flash sürücüler) potansiyel kurbanların onları göreceğinden emin oldukları göze çarpan alanlarda (ör. banyolar, asansörler, hedeflenen bir şirketin otoparkı) bırakırlar. Yem, şirketin bordro listesi olarak sunan bir etiket gibi özgün bir görünüme sahiptir.

Kurbanlar, yemi meraktan alır ve bir iş veya ev bilgisayarına yerleştirir, bu da sisteme otomatik olarak kötü amaçlı yazılım yüklenmesine neden olur.

Yemleme dolandırıcılıklarının mutlaka fiziksel dünyada yapılması gerekmez. Çevrimiçi yemleme biçimleri, kötü amaçlı sitelere yönlendiren veya kullanıcıları kötü amaçlı yazılım bulaşmış bir uygulamayı indirmeye teşvik eden baştan çıkarıcı reklamlardan oluşur.

1.2.2 Kimliğe Bürünme

Kimliğe bürünme saldırısını içeren dolandırıcılıklar, her büyüklükteki şirket için önemli bir tehlike oluşturur. Kimliğe bürünme saldırısı, kötü niyetli URL'ler veya ekler kullanmak yerine, bir çalışanı farkında olmadan sahte bir hesaba para aktarması veya siber suçlularla hassas verileri paylaşması için kandırmak için sosyal mühendislik ve kişiselleştirmeyi kullanır.

Kimliğe bürünme saldırısı, genellikle güvenilir bir kaynaktan geliyormuş gibi görünen bir e-postayı içerir. Bazen e-posta saldırısı bir CEO, CFO veya başka bir üst düzey yöneticiden gelmiş gibi görünen bir mesajla başlayabilir - bu dolandırıcılıklara balina avı e-posta saldırıları da denir . Kimliğe bürünme saldırısı, güvenilir bir meslektaştan, üçüncü taraf bir satıcıdan veya diğer iyi bilinen İnternet markalarından geliyormuş gibi görünen bir mesajı da içerebilir.

1.2.3 Çöp Kutusunu Karıştırma

Çöp bidonlarından değerli bilgiler (adresler, e-postalar vb. içeren kağıtlar) toplamak.

1.2.4 Omuz Üstünden Gözlem

Kullanıcılar klavyede yazarken saldırganın makinelerinin arkalarında gözetlemesi.

1.2.5 Kimlik Avı

En popüler sosyal mühendislik saldırı türlerinden biri olan kimlik avı dolandırıcılıkları, kurbanlarda aciliyet, merak veya korku duygusu yaratmayı amaçlayan e-posta ve kısa mesaj kampanyalarıdır. Ardından, onları hassas bilgileri açığa çıkarmaya, kötü

amaçlı web sitelerinin bağlantılarını tıklamaya veya kötü amaçlı yazılım içeren ekleri açmaya teşvik eder.

Kimlik avı kampanyalarında tüm kullanıcılara aynı veya neredeyse aynı mesajlar gönderildiğinden, tehdit paylaşım platformlarına erişimi olan posta sunucuları için bunları tespit etmek ve engellemek çok daha kolaydır .

1.3 Sosyal Mühendislik Sızma Testi Adımları

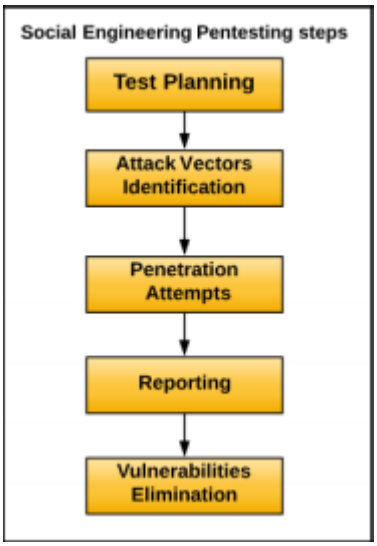
Penetrasyon testi, bir şirketin gerekli korumayı dağıtmak için güvenlik duruşunu değerlendirmek için siyah şapkalı bir hacker saldırısını simüle eder. Penetrasyon testi metodolojik bir süreçtir ve iyi tanımlanmış adımlardan geçer. Pek çok sızma testi türü vardır:

Beyaz kutu pentesti

Kara kutu pentesti

Gri kutu pentesti

Bir sosyal mühendislik sızma testi yapmak için aşağıdaki adımları izlemeniz gerekir:



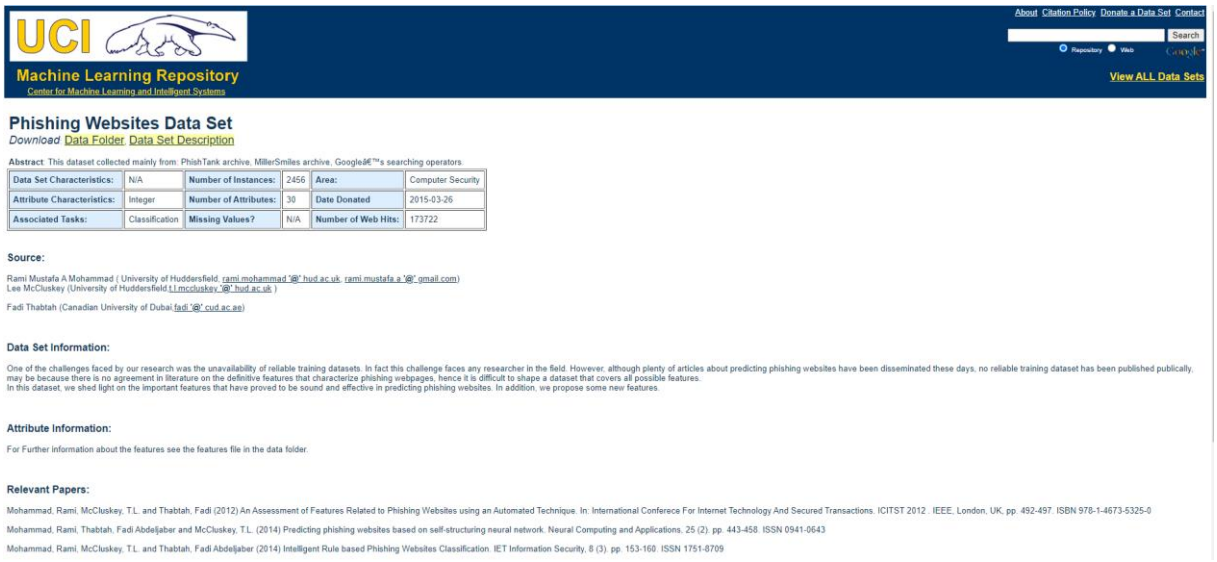
1.4 Makine Öğrenimi Modelleri Kullanarak Gerçek Zamanlı Kimlik Avı Saldırı Tespiti Oluşturma

Aşağıdaki iki yöntemi ele alacağız:

- Lojistik regresyon ile kimlik avı tespiti
- Karar ağaçlarıyla kimlik avı algılama

1.4.1 Lojistik regresyon ile kimlik avı tespiti

Lojistik regresyon algoritması ile sıfırdan bir phishing dedektörü oluşturacağız. Lojistik regresyon, binom tahminleri (iki sınıf) yapmak için kullanılan iyi bilinen bir istatistiksel tekniktir. Modelimiz için UCI Machine Learning Repository (Phishing Websites Data Set) kullanacağız. https://archive.ics.uci.edu/ml/datasets/Phishing+Websites



UCI Machine Learning Repository
Center for Machine Learning and Intelligent Systems

About Citation Policy Donate a Data Set Contact
Search
Repository Web
View ALL Data Sets

Phishing Websites Data Set

Download: Data Folder, Data Set Description

Abstract: This dataset collected mainly from: PhishTank archive, MillerSmiles archive, Google's searching operators.

Data Set Characteristics:	N/A	**Number of Instances:**	2456	**Area:**	Computer Security
Attribute Characteristics:	Integer	**Number of Attributes:**	30	**Date Donated**	2015-03-26
Associated Tasks:	Classification	**Missing Values?**	N/A	**Number of Web Hits:**	173722

Source:

Rami Mustafa A Mohammad (University of Huddersfield, rami.mohammad '@' hud.ac.uk, rami.mustafa a '@' gmail.com)
Lee McCluskey (University of Huddersfield,t.l.mccluskey '@' hud.ac.uk)

Fadi Thabtah (Canadian University of Dubai,fadi '@' cud.ac.ae)

Data Set Information:

One of the challenges faced by our research was the unavailability of reliable training datasets. In fact this challenge faces any researcher in the field. However, although plenty of articles about predicting phishing websites have been disseminated these days, no reliable training dataset has been published publically, may be because there is no agreement in literature on the definitive features that characterize phishing webpages, hence it is difficult to shape a dataset that covers all possible features.
In this dataset, we shed light on the important features that have proved to be sound and effective in predicting phishing websites. In addition, we propose some new features.

Attribute Information:

For Further information about the features see the features file in the data folder.

Relevant Papers:

Mohammad, Rami, McCluskey, T.L. and Thabtah, Fadi (2012) An Assessment of Features Related to Phishing Websites using an Automated Technique. In: International Conference For Internet Technology And Secured Transactions. ICITST 2012 . IEEE, London, UK, pp. 492-497. ISBN 978-1-4673-5325-0

Mohammad, Rami, Thabtah, Fadi Abdeljaber and McCluskey, T.L. (2014) Predicting phishing websites based on self-structuring neural network. Neural Computing and Applications, 25 (2). pp. 443-458. ISSN 0941-0643

Mohammad, Rami, McCluskey, T.L. and Thabtah, Fadi Abdeljaber (2014) Intelligent Rule based Phishing Websites Classification. IET Information Security, 8 (3). pp. 153-160. ISSN 1751-8709

Index of /ml/machine-learning-databases/00327

- Parent Directory
- .old.arff
- Phishing Websites Features.docx
- Training Dataset.arff

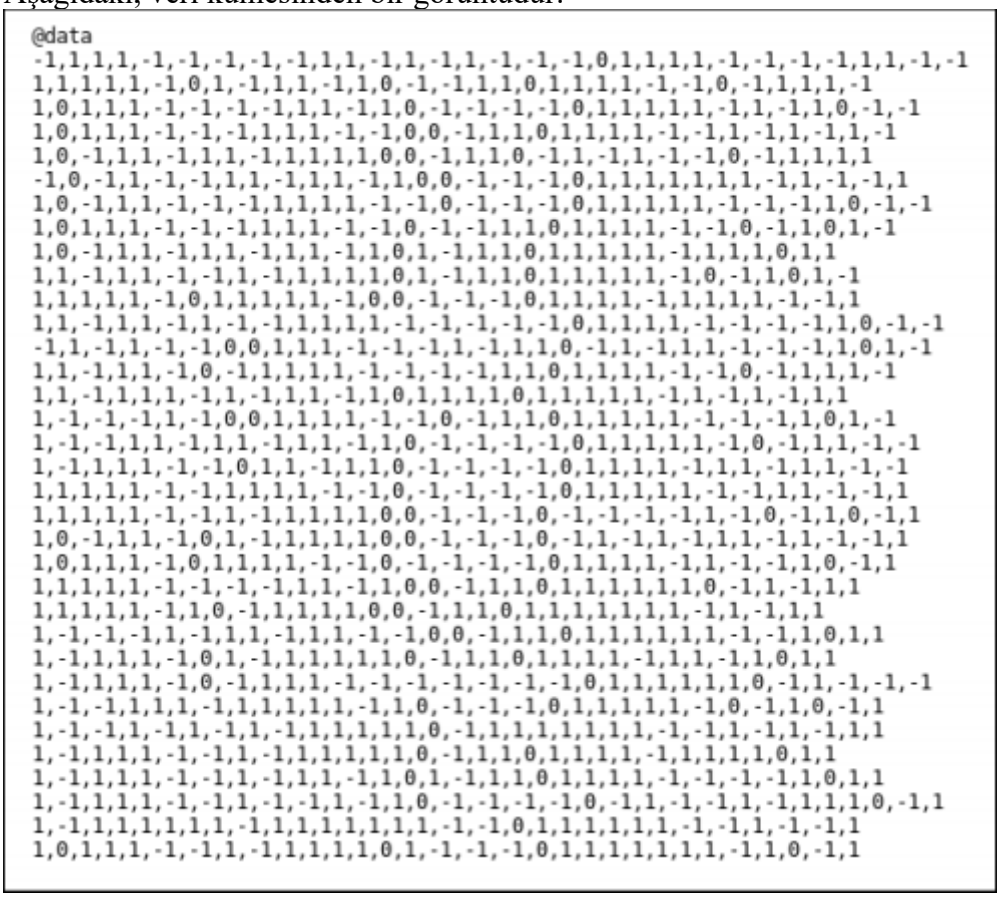
Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips SVN/1.7.14 Phusion_Passenger/4.0.53 mod_perl/2.0.11 Perl/v5.16.3 Server at archive.ics.uci.edu Port 443

Veri kümesi bir arff dosyasıdır.

```
@relation phishing

@attribute having_IP_Address  { -1,1 }
@attribute URL_Length   { 1,0,-1 }
@attribute Shortining_Service { 1,-1 }
@attribute having_At_Symbol   { 1,-1 }
@attribute double_slash_redirecting { -1,1 }
@attribute Prefix_Suffix  { -1,1 }
@attribute having_Sub_Domain  { -1,0,1 }
@attribute SSLfinal_State  { -1,1,0 }
@attribute Domain_registeration_length { -1,1 }
@attribute Favicon { 1,-1 }
@attribute port { 1,-1 }
@attribute HTTPS_token { -1,1 }
@attribute Request_URL  { 1,-1 }
@attribute URL_of_Anchor { -1,0,1 }
@attribute Links_in_tags { 1,-1,0 }
@attribute SFH  { -1,1,0 }
@attribute Submitting_to_email { -1,1 }
@attribute Abnormal_URL { -1,1 }
@attribute Redirect  { 0,1 }
@attribute on_mouseover  { 1,-1 }
@attribute RightClick  { 1,-1 }
@attribute popUpWidnow  { 1,-1 }
@attribute Iframe { 1,-1 }
@attribute age_of_domain  { -1,1 }
@attribute DNSRecord   { -1,1 }
@attribute web_traffic  { -1,0,1 }
@attribute Page_Rank { -1,1 }
@attribute Google_Index { 1,-1 }
@attribute Links_pointing_to_page { 1,0,-1 }
@attribute Statistical_report { -1,1 }
@attribute Result  { -1,1 }
```

Aşağıdaki, veri kümesinden bir görüntüdür:



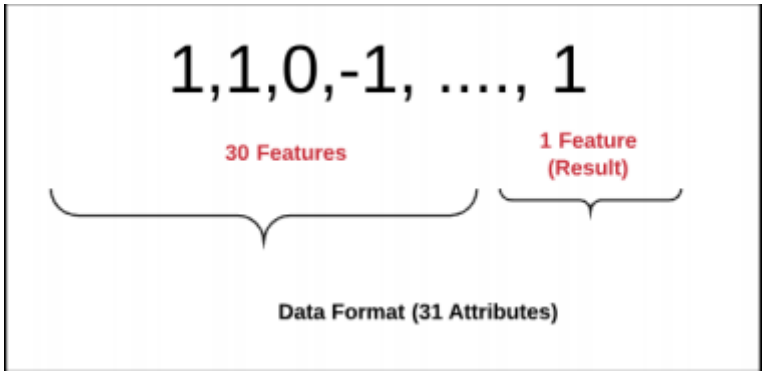
```
@data
-1,1,1,1,-1,-1,-1,-1,-1,1,1,-1,1,-1,1,-1,-1,-1,0,1,1,1,1,-1,-1,-1,-1,1,1,-1,-1
1,1,1,1,1,-1,0,1,-1,1,1,-1,1,1,0,-1,-1,1,1,0,1,1,1,1,-1,-1,0,-1,1,1,1,-1
1,0,1,1,1,-1,-1,-1,-1,1,1,-1,1,1,0,-1,-1,-1,-1,0,1,1,1,1,1,-1,1,-1,1,0,-1,-1
1,0,1,1,1,-1,-1,-1,1,1,1,-1,-1,0,0,-1,1,1,1,0,1,1,1,1,-1,-1,1,-1,1,-1,1,-1
1,0,-1,1,1,-1,1,1,-1,1,1,1,1,1,1,0,0,-1,1,1,1,0,-1,1,-1,1,-1,-1,0,-1,1,1,1,1
-1,0,-1,1,-1,-1,1,1,-1,1,1,1,-1,1,1,0,0,-1,-1,-1,0,1,1,1,1,1,1,1,-1,1,-1,-1,1
1,0,-1,1,1,-1,-1,-1,1,1,1,1,1,-1,-1,0,-1,-1,-1,0,1,1,1,1,1,-1,-1,-1,1,0,-1,-1
1,0,1,1,1,-1,-1,-1,1,1,1,1,-1,-1,0,-1,-1,1,1,0,1,1,1,1,-1,-1,0,-1,1,0,1,-1
1,0,-1,1,1,-1,1,1,-1,1,1,-1,1,0,1,-1,1,1,0,1,1,1,1,1,-1,1,1,1,0,1,1
1,1,-1,1,1,-1,-1,1,-1,1,1,1,1,1,0,1,-1,1,1,0,1,1,1,1,1,-1,0,-1,1,0,1,-1
1,1,1,1,1,-1,0,1,1,1,1,1,1,-1,0,0,-1,-1,-1,0,1,1,1,1,1,-1,1,1,1,1,1,-1,-1,1
1,1,-1,1,1,-1,1,1,-1,-1,1,1,1,1,1,-1,-1,-1,-1,-1,0,1,1,1,1,1,-1,-1,-1,1,1,0,-1,-1
-1,1,-1,1,1,-1,-1,0,0,1,1,1,1,-1,-1,-1,1,-1,1,1,1,0,-1,1,-1,1,1,1,-1,-1,-1,1,1,0,1,-1
1,1,-1,1,1,-1,0,-1,1,1,1,1,1,-1,-1,-1,-1,1,1,1,0,1,1,1,1,1,-1,-1,0,-1,1,1,1,1,-1
1,1,-1,1,1,1,-1,1,-1,1,1,-1,1,1,0,1,1,1,1,1,0,1,1,1,1,1,1,-1,1,-1,1,-1,1,1
1,-1,-1,-1,1,-1,0,0,1,1,1,1,1,-1,-1,0,-1,1,1,0,1,1,1,1,1,1,-1,-1,-1,1,0,1,-1
1,-1,-1,1,1,1,-1,1,1,-1,1,1,1,-1,1,1,0,-1,-1,-1,-1,0,1,1,1,1,1,1,-1,0,-1,1,1,1,-1,-1
1,-1,1,1,1,1,-1,-1,1,-1,1,1,1,1,1,0,-1,-1,-1,-1,-1,0,1,1,1,1,1,1,-1,1,-1,-1,1,1
1,1,1,1,1,1,-1,-1,-1,-1,1,1,1,-1,1,1,0,-1,1,1,0,1,1,1,1,1,1,1,0,-1,1,-1,1,1
1,1,1,1,1,1,-1,1,0,-1,1,1,1,1,1,0,0,-1,1,1,0,1,1,1,1,1,1,1,1,-1,1,-1,1,1
1,-1,-1,-1,1,-1,1,1,1,-1,1,1,1,-1,-1,0,0,-1,1,1,1,0,1,1,1,1,1,1,1,-1,-1,1,1,0,1,1
1,-1,1,1,1,1,-1,0,1,-1,1,1,1,1,1,1,1,0,-1,1,1,1,0,1,1,1,1,1,1,1,-1,1,-1,1,1,0,1,1
1,-1,-1,1,1,1,-1,0,-1,1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0,1,1,1,1,1,1,1,0,-1,1,-1,-1,-1
1,-1,-1,1,1,1,1,-1,1,1,1,1,1,1,-1,1,1,0,1,1,0,-1,-1,-1,-1,-1,1,1,1,1,1,1,1,0,-1,1,-1,1
1,-1,-1,1,-1,1,-1,1,-1,1,1,1,1,1,1,1,0,-1,1,1,1,1,1,1,1,1,-1,-1,1,-1,1,-1,1,1
1,-1,1,1,1,1,-1,-1,1,-1,1,1,1,1,1,1,1,0,-1,1,1,0,1,1,1,1,1,1,1,-1,-1,-1,-1,1,0,1,1
1,-1,1,1,1,1,-1,-1,1,-1,-1,1,-1,1,0,-1,-1,-1,-1,0,-1,1,-1,-1,1,-1,1,1,1,1,0,-1,1
1,-1,1,1,1,1,1,1,-1,1,1,1,1,1,1,1,1,-1,-1,0,1,1,1,1,1,1,1,-1,-1,1,-1,1,1
1,0,1,1,1,-1,-1,1,-1,1,1,1,1,1,0,1,-1,-1,-1,0,1,1,1,1,1,1,1,1,-1,1,0,-1,1
```

Daha iyi manipülasyon için veri setinin csv hali:

dataset.csv — Open with Google Sheets

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	-1	1	1	1	-1	-1	-1	-1	-1	1	1	-1	
2	1	1	1	1	1	-1	0	1	-1	1	1	-1	
3	1	0	1	1	1	-1	-1	-1	-1	1	1	-1	
4	1	0	1	1	1	-1	-1	-1	1	1	1	-1	-1
5	1	0	-1	1	1	-1	1	1	-1	1	1	1	1
6	-1	0	-1	1	-1	-1	1	1	-1	1	1	-1	1
7	1	0	-1	1	1	-1	-1	-1	1	1	1	1	-1
8	1	0	1	1	1	-1	-1	-1	1	1	1	-1	-1
9	1	0	-1	1	1	-1	1	1	-1	1	1	-1	1
10	1	1	-1	1	1	-1	-1	1	-1	1	1	1	1
11	1	1	1	1	1	-1	0	1	1	1	1	1	-1
12	1	1	-1	1	1	-1	1	-1	-1	1	1	1	1
13	-1	1	-1	1	-1	-1	0	0	1	1	1	-1	-1
14	1	1	-1	1	1	-1	0	-1	1	1	1	1	-1
15	1	1	-1	1	1	1	-1	1	-1	1	1	-1	1
16	1	-1	-1	-1	1	-1	0	0	1	1	1	1	-1
17	1	-1	-1	1	1	-1	1	1	-1	1	1	-1	1
18	1	-1	1	1	1	-1	-1	0	1	1	-1	1	1

Veri kümesinin her satırı şu biçimde temsil edilir.



UCI Machine Learning Repository (Phishing Websites Data Set) https://archive.ics.uci.edu/ml/datasets/Phishing+Websites:

```
In [1]: #the required libraries
        import numpy as np
        from sklearn import *
        from sklearn.linear_model import LogisticRegression
        from sklearn.metrics import accuracy_score

In [2]: #Load the data
        db = np.genfromtxt('dataset.csv', delimiter=',', dtype=np.int32)

In [3]: #identify the inputs and the outputs
        inputs = db[:,:-1]
        outputs = db[:, -1]

In [4]: #Dividing the dataset into training data (x) and testing data (y)
        x_inputs = inputs[:1000]
        x_outputs = outputs[:1000]
        y_inputs = inputs[1000:]
        y_outputs = outputs[1000:]

In [5]: #Creating the scikit-learn logistic regression classifier:
        classifier = LogisticRegression()

In [6]: #Training the classifier
        classifier.fit(x_inputs, x_outputs)
Out[6]: LogisticRegression()

In [7]: #Making Prediction
        predictions = classifier.predict(y_inputs)

In [8]: #Accuracy of Phishing Detector Model
        accuracy = 100 * accuracy_score(y_outputs, predictions)

In [9]: print("The accuracy of your Logistic Regression on testing data is :"+str(accuracy))
        The accuracy of your Logistic Regression on testing data is :85.66882148184982

In [ ]:
```

Modelin doğruluğu yaklaşık %85'tir. Modelin 100'den 85'i kimlik avı URL'si tespit ettiğinden bu iyi bir doğruluktur.

1.4.2 Karar ağaçlarıyla kimlik avı algılama

İkinci modeli oluşturmak için aynı makine öğrenimi kitaplıklarını kullanacağız, bu yüzden onları tekrar içe aktarmaya gerek yok. Ancak, karar ağacı sınıflandırıcısını sklearn'den içe aktaracağız:

147 lines (147 sloc) | 2.91 KB Raw Blame

```
In [1]: #the required libraries
        import numpy as np
        from sklearn import tree
        from sklearn.metrics import accuracy_score

In [2]: #Load the data
        db = np.genfromtxt('dataset.csv', delimiter=',', dtype = np.int32)

In [3]: #identfy the inputs and the outputs
        inputs = db[:,:-1]
        outputs = db[:,-1]

In [4]: #Dividing the dataset into training data (x) and testing data (y)
        x_inputs= inputs[:2000]
        x_outputs= outputs[:2000]
        y_inputs= inputs[2000:]
        y_outputs= outputs[2000:]

In [5]: #Creating the DecisionTreeClassifier()
        #scikit-learn classifier
        classifier = tree.DecisionTreeClassifier()

In [6]: #Training the classifier
        classifier.fit(x_inputs,x_outputs)
Out[6]: DecisionTreeClassifier()

In [7]: #Making Prediction
        prediction=classifier.predict(y_inputs)

In [8]: #Accuracy of Phishing Detector Model
        accuracy = 100 * accuracy_score(y_outputs, prediction)

In [9]: print("The accuracy of your Decision Tree on testing data is :" + str(accuracy))
        The accuracy of your Decision Tree on testing data is :90.35891772501381

In [ ]:
```

İkinci modelin doğruluğu yaklaşık %90,4'tür.

2 Sonuç

Kimlik avı, saldırganın e-posta veya diğer iletişim kanallarında saygın bir kişi olarak göndererek oturum açma kimlik bilgileri veya hesap bilgileri gibi hassas bilgileri öğrenmeye çalıştığı bir dolandırıcılık biçimidir.

Kurban, bilinen bir kişi veya kuruluş tarafından gönderilmiş gibi görünen bir mesaj alır. Mesaj, kullanıcının bilgisayarını hedefleyen kötü amaçlı yazılımlar içerir.

Kimlik avı saldırganlar için çok popülerdir. Çünkü birinin meşru görünen kötü niyetli bir bağlantıya tıklaması için kandırmak, bilgisayarın savunma sistemlerini kırmaya çalışmaktan daha kolaydır.

Bu makalede, phishing domain özelliklerini, bu domainleri tespit etmenin neden önemli olduğunu ve makine öğrenimi teknikleri kullanılarak nasıl tespit edilebileceğini açıkladık. Sonuç olarak iki farklı proje oluşturarak kimlik avı girişimlerini tespit etmeyi öğrendik. Son teknoloji Python makine öğrenimi kitaplıkları sayesinde iki farklı makine öğrenimi tekniğini kullanarak bir kimlik avı detektörünün nasıl geliştirileceğini keşfettik.

3 Kaynakça

Mastering Machine Learning for Penetration Testing – Chiheb Chebbi
Machine Learning for Hackers – Drew Conway & Jhon Myles White
Hacker's Guide to Machine Learning with Python – Venelin Valkov