Modelo computacional para la detección automática de URLs maliciosas utilizando técnicas de machine learning

Nicolas R Enciso[1] and Jorge E Camargo[2]

Universidad Nacional de Colombia UNSecure Lab Research Group
jecamargom@unal.edu.co
UNSecure Lab Research Group nricardoe@unal.edu.co
www.unsecurelab.org

Resumen El presente trabajo, expone una metodología alterna de extracción de características léxicas a URLs, que permitan obtener los elementos más diferenciadores de cada uno de los grupos de ataques web más conocidos (malware, phishing, defacement, spam). El modelo propuesto parte de un dataset público que contiene más de 100.000 URLs entre benignas, y anomalas, las cuales son sometidas a métodos de visualización como forma de identificación (PCA, T-SNE), y a algoritmos de machine learning (Bayes Gaussiano, Random Forest, Regresión logística, SVM etc). Los resultados muestran una precisión de más del 85 % , con un esfuerzo computacional menor, comparativamente con los resultados de los autores del dataset.

Keywords: Spam · URL · Phishing · machine learning classifier · web attacks

1. Introducción

Con el explosivo crecimiento de las conexiones a internet, y con ella la cantidad de usuarios, los temas de seguridad se convierte en pieza clave para todos los actores de la red. Internet es un lugar plagado de peligros, como queda demostrado en el reporte del buscador Google, para finales de Octubre de 2019, habían eliminado más de 32000 sitios web por ser malociosos sólo en ese mes [1]. Adicionalmente a la gran cantidad diaria de sitios web maliciosos que son detectaddos, hay una gran cantidad que no logran ser dectectados y que generan grandes brechas de seguridad, es por ello que los ataques web más comunes, figuran como principales responsables en el top 10 de las vulnerabilidades más críticas de la web según OWASP [2].

De acuerdo a lo anterior, es crítico disponer de las herramientas que permitan ofrecer a los usuarios de la web, la posibilidad de navegar de forma segura, antes de ser víctimas de alguno de éstos ataques. Es por ello que el presente trabajo, se enfoca en ofrecer una alternativa computacionalmente eficiente y ligera, capaz de alertar sobre ataques con solo la URL.

2. Antecedentes

Tradicionalmete la forma en la que se ha abrodado la problemática de los ataques web relacionados a sitios web maliciosos, como lo son el spam, phishing, impersonalización (defacement) o malware, es a través de listas negras. Sin embargo, esta solución resulta quedarse corta, debido a que la tendencia de quienes usan estos ataques, es a cambiar muy continuamente de URL, lo que hace inserviles las listas. Hay autores que han intentado combinar lo mejor de las listas negras con caracterizaciones léxicas de las URL [3].

Otras aproximaciones al problema, incluyen hacer un análisis del contenido de la página web [4], sin ambargo, esta metodología resulta no ser conveniente, en la medida en que exige ingresar a explorar la pgaina a la que conduce la URL sospechosa, lo que genera riesgos de seguridad mayores. Finalmente, la tendencia es a usar métodos léxicos combinados con machine learning, dando flexibilidad, escalabilidad y menores riesgos a la hora de implementar, como se ha ido haciendo por ejemplo, combinando características propias de URLs de phishing con caraterísticas del dominio en donde están alojadas éstas direcciones, obteniendo efectividad mayor al 90 % [5]. Adicionalmente, se han combinado técnicas de caracterización descriptiva de las URL, junto con caracterizaciones léxicas estáticas para entrenar algoritmos de clasificación [6], incluso se han implementado clasificadores multiclase, que además de detectar una URL maliciosa, la clasifica en un grupo de ataque [9].

El estado del arte actual, indica que es en el machine elarning en donde se tiene los mejores resultados y la mayor flexibilidad a la hora de detectar y clasificar ataques web usando como insumo la URL.

3. Metodología

En el presente trabajo, se inició con un data set abierto, con más de 100000 URLs entre benignas y maliciosas, en donde se encuentras 4 tipos de ataque: spam, phishing, impersonalización (defacement) y malware. Los autores del dataset, adicionalmente, generaron un estudio de selección de mejores caracteríticas, que permitieran detectar estos ataques, con no más de 8 características, luego de realizar sobre ellos dos algoritmos de selección, obteniendo 4 datasets, uno por cada tipo de ataque [10].

Adicionalmente, los autores dan las URLs que usaron para extraer las características. Es con estas que se realiza el presente trabajo. Como primera medida, la selección de características se hizo conforme a los elementos más diferenciadores encontrados en cada tipo de ataque reflejados en la URL, y que fueran comunes entre ellos, destacando un trabajo hecho por [7] en donde se presentan características propias de ataque web más avanzados como SQL injectio o XSS.

Basado en ello, a cada URL se le extrajeron las siguientes características: Longitud, cantidad de caracteres sospechosos (@, /)[13], palabras clave de posibles referencias a SQL o XSS [11], [12], saltos de línea, métrica de divergencia de kullback para conocer qué tal similar es la URL con el idioma de inglés, y

la prueba de Smirnov, con la que se obtiene qué tan probable es que la URL provenga del inglés común.

Con las características extraídas, se prosiguió a realizar una reducción de la dimensionalidad, con el fin de obtener una visualización de los datos, y poder tener una aproximación, si la caracterización de las URLs fue lo suficientemente buena, como para poder diferenciar a simple vista los dos grupos (ataque, benigno). Para ello se implementó análisis de componentes principales (PCA) y reducción por t-SNE, obteniendo imágenes en 2D de cada grupo de datos.

Finalmente, se tomó cada dataset de cada tipo de ataque caracterizado, se le añadieron URLs provenientes del grupo de benignos, creando nuevos dataset, en relación 4 a 1, de URLs benignas vs URLs de ataque, con los cuales se entrenaron algoritmos de machine learning para clasificación binaria, los cuales fueron bayes, random forest, regresión logística, y SVM en 4 modadlidades, para tener una gráfica ROC, mostrando el performance de cada uno de ellos.

Todos los elementos anteriores se realizaron apoyados en el software SKlearn para python, así como numpy y pandas.

4. Resultados

Para la visualización de los datos caracterizados, se obtuvieron los siguientes resultados:

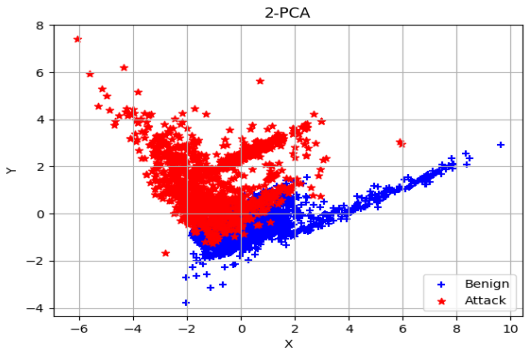

Figura 1. PCA para datos de phishing, en rojo puntos de phishing, en azul puntos de URLs benignas

Como se puede observar, se pueden identificar claramente grupos de un mismo tipo formando clusters, lo que permite deducir, que la caraterización aplicad aha sido existosa, en la medida en que es fácilmente diferenciable respecto a un área, el tipo de dato a observar.

Para el caso de los clasificadores de machine learning, se obtuvieron los siguientes resultados:

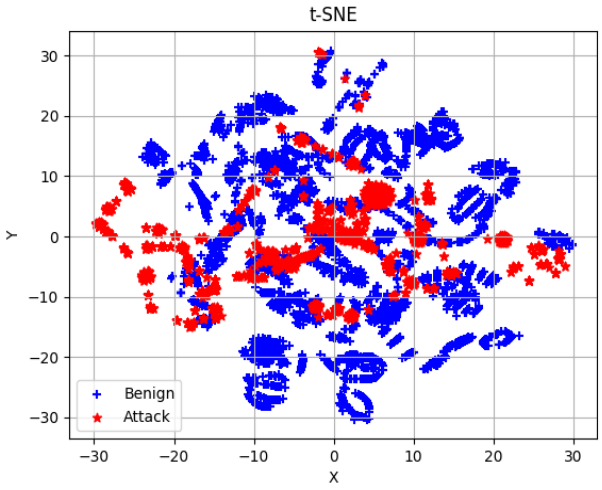

Figura 2. T-SNE para datos de malware, en rojo puntos de malware, en azul puntos de URLs benignas

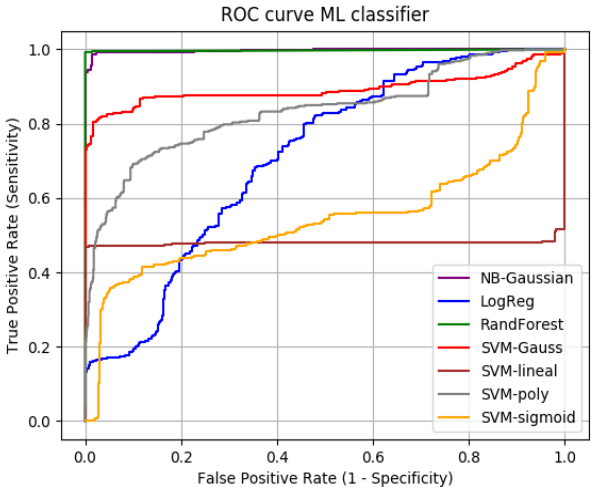

Figura 3. ROC gráfica de desempeño para calsificaicón de URLs de spam

Como podemos observar, random forest, bayes gaussiano y SVM gaussiano, obtuvieron precisiones maypres al 80 %, incluso para el caso de random forest, con precisión mayor al 95 %, tendencia que se mantuvo para clasificación de los otros grupos de ataque.

5. Conclusiones

Para el trabajo realizado, se puede concluir que es posible caracterizar de manera eficiente y sin la necesidad de más algoritmos de selección de características, los 4 tipos de ataque web trabajados acá, destacando que con un solo set de características, se pudo diferenciar con resultados destacables, a los 4, sin tener que hacer modificaciones para cada uno. Adicionalmente, se introdujo la visualización, que permite hacer una rápida revisión del estado de los datos, y poder hacer reflexiones al respecto, antes de correr algoritmos de machine elarning. COmo trabajo futuro, se pretende que se tomen las características expuestas para cada tipo dadas por los autores del dataset, y se combinen con los usados acá, en aras de obtener una precisión mayor, sin tener que correr algoritmos de selección adiocional que añaden carga computacional, a una tarea que por naturaleza, exige ser liviano.

Referencias

1. Google. (2019). Navegación segura: Software malicioso y suplantación de identidad (phishing). Octubre 2019, de Google Inc, https://transparencyreport.google.com/safe-browsing/overview
2. OWASP. (2017). Top 10 application security risks. Octubre 2019, de OWASP, https://www.owasp.org/index.php/Category:OWASP$_{T}op_{T}en_{P}roject$.
3. Ma, J., et al.: Identifying suspicious URLs: an application of large-scale online learning. In: Proceedings of the 26th Annual International Conference on Machine Learning. ACM (2009) 2016.
4. Thomas, K., et al.: Design and evaluation of a real-time URL spam filtering service. In: Proceeding of the IEEE Symposium on Security and Privacy (SP) (2011)
5. Chu, W., et al.: Protect sensitive sites from phishing attacks using features extractable from inaccessible phishing URLs. In: IEEE International Conference on Communications (ICC) (2013)
6. Lin, M.-S., et al.: Malicious URL filtering- a big data application. IEEE Interna- tional Conference on Big Data (2013)
7. Alejandro Correa Bahnsen, Eduardo Contreras Bohorquez, Sergio Villegas, Javier Vargas, Fabio A. González. (2017). Classifying Phishing URLs Using Recurrent Neural Networks. of Easy Solutions Research, MindLab Research Group National University of Colombia.
8. OWASP top ten project 2017, https://www.owasp.org. Last accessed 2 Jan 2019
9. Choi, H., Zhu, B.B., Lee, H.: Detecting malicious web links and identifying their attack types. In: Proceedings WebApps (2011)
10. Mohammad Saiful Islam Mamun, Mohammad Ahmad Rathore, Arash Habibi Lashkari, Natalia Stakhanova and Ali A. Ghorbani, "Detecting Malicious URLs Using Lexical Analysis", Network and System Security, Springer International Publishing, P467–482, 2016.

11. OWASP Cross-site Scripting (XSS), https://www.owasp.org/index.php/Cross-site_Scripting_(XSS). Last accessed 4 Jan 2019
12. OWASP Types of Cross-Site Scripting, https://www.owasp.org/index.php/Types_of_Cross-Site_Scripting. Last accessed 4 Jan 2019
13. OWASP CRLF Injection, https://www.owasp.org/index.php/CRLF_Injection. Last accessed 8 Jan 2019
14. OWASP Guide Project, https://www.owasp.org/index.php/OWASP_Guide_Project. Last accessed 8 Jan 2019
15. OWASP SQL injection, https://www.owasp.org/index.php/SQL_Injection. Last accessed 8 Jan 2019