Detección de Tráfico Botnet con Redes Neuronales MLP

Universidad Técnica Federico Santa María
TEL547 2018-2
Esteban Jara 201330002-2, Edgard Abarcas 201430004-2, Mario Marín 201321048-1

***Abstract*— En este documento se explicará en detalle el proyecto elegido para la asignatura TEL-547, su desarrollo y resultados obtenidos, además de el trabajo futuro necesario para poder terminar de forma exitosa este proyecto.**

I. INTRODUCCIÓN

La amenaza de Malware o programas malignos en el mundo actual es un problema que administradores de redes de computadores no pueden ignorar, el gran problema es que existe una gran facilidad de descargar Malware desde internet, incluso si el usuario no tiene intenciones maliciosas. Como administrador de una red de computadores (por ejemplo en un laboratorio de computadores, o computadores de una empresa) puedes ser una tarea logísticamente compleja al no poder monitorizar cada equipo de forma individual.

II. DESCRIPCIÓN DEL PROYECTO

Se ha visto la posibilidad de utilizar redes neuronales para poder detectar Malware por medio del monitoreo de el trafico TCP/IP, para esto hemos encontrado un dataset el cual contiene capturas de red de una red de computadores infectadas con una botnet.
Con este dataset se espera entrenar una red neuronal MLP que pueda detectar cual trafico de la red corresponde a trafico de botnet, con esto se podría discernir que equipos podrían estar infectados y requieren de una revision del técnico encargado.

III. OBJETIVOS

- Detectar trafico perteneciente a una botnet.
- Detectar IP de la cual proviene el trafico malicioso.
- Levantar una alerta al administrador de los equipos.

IV. DESCRIPCIÓN DEL DATASET

Los datasets utilizados pertenecen a la organización *Stratosphere IPS* [2] , específicamente, se utilizaron los datasets que fueron usados en el paper[1] que dio inicio a la organización Stratosphere.
Los datasets contienen:

- **Botnet exe**: El dataset entrega un ejecutable de el Malware botnet utilizado en la captura, estos archivos nunca fueron inspeccionados.
- **Archivos pcap**: Estos archivos contienen los paquetes capturados en la red, estas capturas fueron editadas para mantener la privacidad de los usuarios involucrados, dejando solo disponible el encabezado de los paquetes.
- **Archivos binetflow**: Estos archivos son los resultantes de procesar los archivos pcap con el programa Argus [3], estos archivos son mas ligeros que los pcap dado que resumen la información, además estos archivos fueron etiquetados por el equipo investigador.

IV-A. Archivos binetflow

Estos archivos estan estructurados como un archivo csv, los cuales contienen:

1. **Start Time**: tiempo de inicio de la transmisión.
2. **Dur**: Duración de la transmisión.
3. **Proto**: Protocolo de transición de datos.
4. **SrcAddr**: Dirección IP de origen.
5. **Sport**: Puerto origen.
6. **Dir**: Dirección de la comunicación (direccional, bidireccional,desconocido, etc)
7. **DstAddr**: Dirección IP destino.
8. **Dport**: Puerto destino.
9. **State**: Desconocido.
10. **sTos**: Desconocido.
11. **dTos**: Desconocido.
12. **TotPkts**: Cantidad de paquetes transmitidos.
13. **TotBytes**: Cantidad de Bytes transmitidos.
14. **SrcBytes**: Cantidad de Bytes transmitidos desde el origen.
15. **Label**: Etiqueta que indica si el trafico es Malware o background (no Malware).

Dichos archivos, se ven de la siguiente forma:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	StartTime	Dur	Proto	SrcAddr	Sport	Dir	DstAddr	Dport	State	sTos	dTos	TotPkts	TotByte	SrcByt	Label			
2	2011/08/15 16:43:28.078942	0.000000	tcp	114.33.245.44	6881	?>	147.32.84.118	1567	RA_	0	0	1	60	60	flow=Background			
3	2011/08/15 16:43:32.283576	13.431.962	tcp	212.93.105.52	49237	->	147.32.84.229	80	SRA_SA	0	0	6	388	208	flow=Background-TCP-Established			
4	2011/08/15 16:43:32.456441	13.350.228	tcp	212.93.105.52	14906	->	147.32.84.229	13363	SRA_SA	0	0	6	388	208	flow=Background-TCP-Established			
5	2011/08/15 16:43:32.850648	13.010.090	tcp	212.93.105.52	60349	->	147.32.84.229	443	SRA_SA	0	0	6	388	208	flow=Background-TCP-Established			
6	2011/08/15 16:45:09.305002	20.990.047	tcp	115.127.24.116	3198	->	147.32.84.229	443	SR_SA	0	0	5	308	122	flow=Background-TCP-Established			
7	2011/08/15 16:45:27.991372	12.542.819	tcp	115.127.24.116	3196	->	147.32.84.229	13363	SR_SA	0	0	5	308	122	flow=Background-TCP-Established			
8	2011/08/15 16:45:30.255050	13.308.726	tcp	115.127.24.116	3198	->	147.32.84.229	443	SR_SA	0	0	5	308	122	flow=Background-TCP-Established			
9	2011/08/15 16:45:57.756664	1.413.248	tcp	77.52.60.161	3767	->	147.32.84.118	6881	S_RA	0	0	4	244	124	flow=Background-TCP-Attempt			
10	2011/08/15 16:47:50.502720	5.147.244	tcp	77.52.60.161	3823	->	147.32.84.118	6881	S_RA	0	0	4	244	124	flow=Background-TCP-Attempt			
11	2011/08/15 16:58:38.284888	2.326.073	tcp	147.32.84.165	1358	->	65.55.40.23	443	FSPA_FSP	0	0	151	133909	6932	flow=From-Botnet-V46-TCP-Established			
12	2011/08/15 16:49:35.840430	6.603.677	tcp	148.100.181.73	50950	->	147.32.84.118	6881	S_RA	0	0	4	252	132	flow=Background-TCP-Attempt			
13	2011/08/15 16:50:08.824635	1.385.300	tcp	77.52.60.161	3895	->	147.32.84.118	6881	S_RA	0	0	4	244	124	flow=Background-TCP-Attempt			
14	2011/08/15 16:50:07.719410	2.867.043	tcp	41.102.134.171	3215	->	147.32.84.229	13363	SR_SA	0	0	3	184	122	flow=Background-TCP-Established			
15	2011/08/15 16:50:09.325122	2.967.635	tcp	41.102.134.171	3218	->	147.32.84.229	443	SR_SA	0	0	3	184	122	flow=Background-TCP-Established			
16	2011/08/15 16:50:10.927382	2.979.128	tcp	41.102.134.171	3220	->	147.32.84.229	80	SR_SA	0	0	3	184	122	flow=Background-TCP-Established			
17	2011/08/15 16:50:10.584643	6.135.476	tcp	41.102.134.171	3215	->	147.32.84.229	13363	SR_SA	0	0	3	184	122	flow=Background-TCP-Established			
18	2011/08/15 16:50:12.292757	6.035.348	tcp	41.102.134.171	3218	->	147.32.84.229	443	SR_SA	0	0	3	184	122	flow=Background-TCP-Established			
19	2011/08/15 16:50:13.900510	6.033.936	tcp	41.102.134.171	3220	->	147.32.84.229	80	SR_SA	0	0	3	184	122	flow=Background-TCP-Established			
20	2011/08/15 16:51:56.205098	11.069.827	tcp	115.127.24.116	3436	->	147.32.84.229	443	SR_SA	0	0	5	308	122	flow=Background-TCP-Established			
21	2011/08/15 16:51:57.676975	10.752.169	tcp	115.127.24.116	3439	->	147.32.84.229	80	SR_SA	0	0	5	308	122	flow=Background-TCP-Established			
22	2011/08/15 16:52:04.205268	20.540.148	tcp	115.127.24.116	3433	->	147.32.84.229	13363	SR_SA	0	0	5	308	122	flow=Background-TCP-Established			

Estructura de los archivos binetflow.

Los archivos binetflow fueron editados para poder ser utilizados de forma adecuada a la red MLP, esto se hizo por medio del código desarrollado por el equipo, *label-editor-V2.1.py*, hay que tener en cuenta que los archivos binetflow originales estaban muy desbalanceados en cuanto a trafico malware y tráfico backgorund. En general, más del 93 % de los datos son background, esto generó un problema, dado que la red MLP no lograba determinar las características que hacían de un paquete de trafico background, distinto a un paquete de tipo malware, por lo tanto, se decidió eliminar

el 95 % de los datos background. El archivo resultante de el procesamiento tiene una estructura csv de la siguiente forma:

1. **Dur**: No fue normalizado.
2. **Proto**: Fue convertido a etiquetas arbitrarias.
3. **SrcAddr**: Los primeros 2 octetos fueron eliminados y se genero un numero uniendo los 2 últimos octetos.
4. **Sport**: Los puertos estandarizados se encontraban escritos de forma hexadecimal, fueron convertidos a numero decimales.
5. **Dir**: Fue convertido a etiquetas arbitrarias.
6. **DstAddr**: Los primeros 2 octetos fueron eliminados y se genero un numero uniendo los 2 últimos octetos.
7. **Dport**: Los puertos estandarizados se encontraban escritos de forma hexadecimal, fueron convertidos a numero decimales.
8. **TotPkts**: Fueron divididos por 65535 para reducir su tamaño de forma significativa.
9. **TotBytes**: Fueron divididos por 65535 para reducir su tamaño de forma significativa.
10. **SrcBytes**: Fueron divididos por 65535 para reducir su tamaño de forma significativa.
11. **Label**: Las etiquetas fueron simplificadas a 1 (Malware) y 0 (background).

Dichos archivos se ven de la siguiente forma:

Dur	Proto	SrcAddr	Sport	Dir	DstAddr	Dport	TotPkts	TotBytes	SrcBytes	Label
0.287716	1	234164	2322	1	87191	80	6.103608758678569e-05	0.003723201342793927	0.0018921187151903563	0
0.146579	1	234163	3783	1	8724	80	4.5777065690089265e-05	0.002777141985198749	0.0018616006713969636	0
3.449038	1	80112	56949	1	86110	443	6.103608758678569e-05	0.003662165255207412	0.0018310826276035706	0
0.145543	1	234171	2442	1	8692	80	4.5777065690089265e-05	0.002777141985198749	0.0018616006713969636	0
1.21961	1	20318	12767	1	84118	6881	6.103608758678569e-05	0.003723201342793927	0.0018921187151903563	0
0.282098	1	234162	2889	1	86195	80	6.103608758678569e-05	0.003723201342793927	0.0018921187151903563	0
771.29071	1	86134	49187	1	2551	80	0.0001525902189669642 2	0.021286335545891 51	0.015213244831006332	0
0.926612	1	110148	4489	1	84118	6881	6.103608758678569e-05	0.003723201342793927	0.0018921187151903563	0
3599.997803	2	13398	16200	4	86125	35248	6.244403753719387	3166.970611123827	2787.169436179141	0
0.001742	1	253133	443	3	86107	51505	7.629510948348211e-05	0.009246967269398032	0.006942854962969687 15	0
941.105042	1	10217	51013	3	85112	22	0.08990615701533532	16.063538567177844	4.993484397650111	0
3230.669189	2	99166	500	1	86165	500	9.155413138017853e-05	0.0324101625085832	0.0324101625085832	0
33.469635	1	232215	443	3	8459	41457	0.0001983672846570534 8	0.023712520027466 24	0.016433966582742047	0
3597.063965	2	8013	514	1	84138	2990	0.27704280155642025	240.88378728923476	240.88378728923476	0
0.000336	2	84138	55592	4	809	53	3.051804379339284 4e-05	0.003265430685893034 4	0.001235980773632 41	0
0.000391	2	84138	46688	4	809	53	3.051804379339284 4e-05	0.003265430685893034 4	0.001235980773632 41	0
0.031266	2	8459	55627	4	809	53	3.051804379339284 4e-05	0.004119935912108034	0.001098649576562 1424	0
0.000631	2	781	18431	4	86165	443	3.051804379339284 4e-05	0.001953154802777142	0.000915541313801 7853	0
3599.334229	2	70198	21850	4	8526	54145	0.039444571602960 25	2.615793087663081	1.853910124361028 5	0
16.801308	1	85124	51690	2	52196	80	6.103608758678569e-05	0.003662165255207412	0.003662165255207412	0
0.330835	1	8459	54671	1	11490	80	0.000137331197070267 8	0.023483634690915793	0.010666056305790799	0

Estructura de los archivos editados.

V. DESCRIPCIÓN DE LA RED MLP

Para implementar la red neuronal MLP, se utilizó la biblioteca *sklearn* [4].Además, para poder almacenar los datos de manera eficiente dentro del programa se utilizó la biblioteca *pandas* [5].

Las características de la red MLP utilizada son:

- Puede ser considerada una ”shallow network” dado que solo tiene una capa intermedia de 100 neuronas.
- Posee la función de optimización sgd (Stochastic Gradient Descent).
- Tiene una razón de entrenamiento adaptativa, lo cual significa que esta disminuye con las iteraciones de entrenamiento.
- Ocupa una función de activación Tanh.

Esta configuración de la red, sumado a los datasets previamente descritos y los respectivos cambios que se les hicieron, arrojaron los mejores resultados. Para esto, cabe destacar que se utilizó, aproximadamente, un 70 % de los datasets para entrenar la red, y el otro 30 % de los datasets, fue usado en labores de testeo.

VI. RESULTADOS OBTENIDOS

En primera instancia se obtuvieron resultados que, si bien, tenían un alto grado de exactitud, mostraban que la red no lograba identificar el tráfico Botnet. En esta red se utilizó la función de activación Relu, además de eso, la cantidad de tráfico Botnet era mucho más baja que la de tráfico Background, los resultados bajo dichas condiciones se ilustran en la siguiente imagen:

```
--------------------------------------------------
test with dataset: testing/v2/out_52
correct 0 : 99070
correct 1 : 0
incorect 0 : 8164
incorect 1 : 0
Final points given by mlp.score : 0.92386743010612 3
--------------------------------------------------
test with dataset: testing/v2/out_54
correct 0 : 1884954
correct 1 : 0
incorect 0 : 40003
incorect 1 : 0
Final points given by mlp.score : 0.9792187565748222
--------------------------------------------------
test with dataset: testing/v2/out 53
correct 0 : 323286
correct 1 : 0
incorect 0 : 2168
incorect 1 : 0
Final points given by mlp.score : 0.9933385363215692
```

Resultados obtenidos con la versión 2 de la Red Neuronal.

Posterior a eso se procedió a utilizar como función de activación tanh, además de eso se modificaron los datasets, se eliminó tráfico Background, aproximadamente un 95 % de estos datos, con el fin de disminuir la razón que existía entre el tráfico Botnet y el Normal, así como se describió anteriormente. Con esta configuración se obtuvo los siguientes resultados:

```
test with dataset: testing/v2.1/out_52
correct 0 : 4835
correct 1 : 8162
incorect 0 : 2
incorect 1 : 135
Final points given by mlp.score : 0.9895690574082534
--------------------------------------------------
test with dataset: testing/v2.1/out_50
correct 0 : 93229
correct 1 : 165186
incorect 0 : 19801
incorect 1 : 2086
Final points given by mlp.score : 0.9219163616385184
--------------------------------------------------
test with dataset: testing/v2.1/out_53
correct 0 : 16009
correct 1 : 575
incorect 0 : 1593
incorect 1 : 336
Final points given by mlp.score : 0.8958029492788852
```

Resultados obtenidos con la versión 2 de la Red Neuronal.

VII. TRABAJO FUTURO

De acuerdo a los resultados obtenidos con el entrenamiento de la red, y dado que es posible almacenar dicha red, ya entrenada. A futuro sería deseable implementar algún tipo de software que, en tiempo real, sea capaz de detectar, con un alto grado de precisión, las posibles amenazas, de manera que sea posible alertar al administrador de la red y así prevenir algún tipo de ataque. Ligado a esto, tal vez sería bueno, implementar algún tipo de interfaz que permita replicar este proyecto, ya sea utilizando la red entrenada o nuestros códigos para a través nuevos datasets generar una nueva red neuronal, de manera más intuitiva y no solamente por comando como se realizó en este proyecto.

Si bien se logró detectar tráfico Malware, no se logró detectar las direcciones IP asociadas a dicho Malware, como trabajo

futuro y complementario se esperaría asociar direcciones IP a tráfico Malware, de esta forma, se podría bloquear todo tipo de tráfico proveniente de dicha IP, en caso de que la persona lo considere necesario. Esto sería un gran avance en el desarrollo futuro del proyecto.

APÉNDICE

REFERENCES

[1] An empirical comparison of botnet detection methods" Sebastian Garcia, Martin Grill, Jan Stiborek and Alejandro Zunino. Computers and Security Journal, Elsevier. 2014. Vol 45, pp 100-123. http://dx.doi.org/10.1016/j.cose.2014.05.011.

[2] https://www.stratosphereips.org/datasets-ctu13

[3] https://qosient.com/argus/

[4] https://scikit-learn.org/stable/index.html

[5] https://pandas.pydata.org/

ANEXO

- **Repositorio GitHub**: https://github.com/mario-marin/TEL-354_Proyecto