用機器學習偵測 DDoS

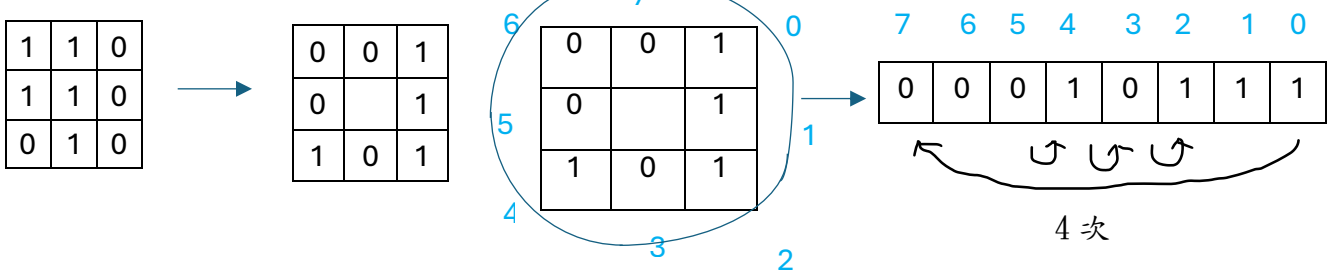
Using Machine Learning to detect DDoS

指導教授:洪昌鈺
專題成員:許桓瑞
開發工具:python
測試環境:window

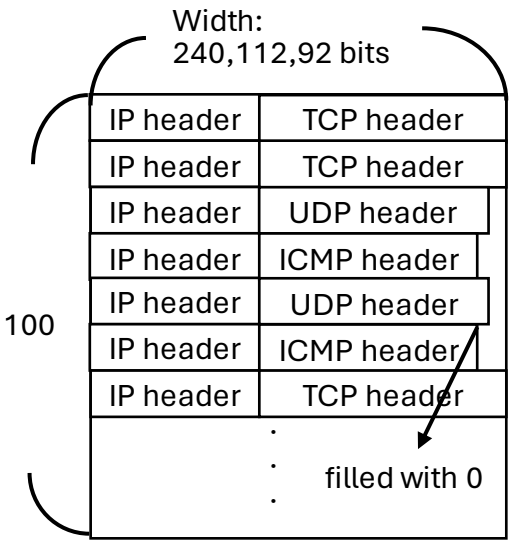
一、簡介

DDoS，分散式阻斷服務攻擊，攻擊者會對受害者的系統或網路進行大量訪問或請求，我針對的方向是流量攻擊(flood attack)，有 3 種，SYN flood，透過 TCP 的連線機制，攻擊者會送大量的 SYN 封包給受害者，受害者會逐一回應並等待接收 ACK 封包，但攻擊者不會發送 ACK 封包，導致受害者會有一段時間被占用；UDP flood，攻擊者會送大量 UDP 封包給受害者，讓受害者一直檢查並回應攻擊者，無暇顧及正常使用者；ICMP flood，攻擊方式與 UDP flood 相似，傳送大量 ICMP 封包，讓受害者不斷回應攻擊者。這次基於 CNN 的想法，將封包的 header 提取出來並拼成圖像(如圖(二)所示)，作為 model 的輸入。這次用了 Decision Tree model, SVM model, RandomForest model 和 CNN 去做比較，特徵的部分採用圖像的紋理特徵，採取紋理特徵的方式有 Gray-Level Co-Occurrence Matrix (GLCM), local binary pattern(LBP), Image Gradient，GLCM 是計算兩個像素值在一定距離且相對角度在 0, 45, 90, 135 度所出現次數的矩陣，接著會計算對比度、能量和熵，對比度是計算共生矩陣主對角線上的轉動慣量，體現出矩陣內的值如何分布的，能量是計算矩陣內各個元素的平方和，熵計算了整個圖像的複雜程度，熵越大，圖像越複雜，因為有 4 種角度，所以對比度和能量各會算出 4 個數值，而熵是對全部計算再取平均，所以只會輸出一個值，最後結合起來就會是長度 9 的特徵向量(計算方式如圖(三)所示)；LBP 是計算每個像素對於在一定範圍內其他像素的大小關係(>=:0,<:1)，再依照一定方向形成一個 2 進位的數字，把 0 接 1 和 1 接 0 的次數加總(要循環計算,計算方式如圖(一)所示)，所以會有 0~8 次，全部算完後，統計出現次數的個數，最後形成一個長度為 9 的特徵向量；Image Gradient 是計算整體的圖像強度，會計算每個像素的上下及左右差值，並對其平方和開根號算出 gradient，具體計算是利用 Sobel 濾波器來計算水平梯度和垂直梯度，再利用這兩個梯度算出角度並依 0 到π(不包括π)以$\pi/8$為間隔做直方圖，所以就有 7 個數值的輸出(詳細計算請看簡報)。我們有 10000 張圖像，每張圖像會產生一組特徵向量，GLCM, LBP, Image Gradient 分別會產生長度為 9,9,7 的特徵向量，並將這 3 種特徵向量結合成長度 25 的向量，最終會有 10000*25 的矩陣輸入 model 中，而輸出的部分，我稍微簡化一下，原本一張圖像的輸出是長度 3 的向量，分別代表是哪一種攻擊方式(SYN flood, UDP flood, ICMP flood)，但是長度 3 的輸出計算 confusion matrix 不太嚴謹，所以我把輸出改成長度 1，僅僅判斷是否有攻擊，有攻擊輸出 1，反之輸出 0。

圖(一)



圖(二)



圖(三)

0	0	1	1
0	1	0	1
0	0	0	1
1	0	0	0

P_{0° =

j \ i	0	1
0	5	6
1	6	1

P(i,j,d,θ)

P(0,0,1,0°)=5

0 和 0 在相對 0 度且距離為 1 出現 5 次

對比度公式:$\sum_i \sum_j (i-j)^2 * P(i,j)$

能量公式:$\sum_i \sum_j P(i,j)^2$

熵公式:$-\sum_i \sum_j P(i,j) \log P(i,j)$

for i=0 to 1 , j=0 to 1

contrast={$contrast_{0^\circ}$, $contrast_{45^\circ}$,$contrast_{90^\circ}$,$contrast_{135^\circ}$}

energy={$energy_{0^\circ}$, $energy_{45^\circ}$,$energy_{90^\circ}$,$energy_{135^\circ}$}

entropy=($entropy_{0^\circ}$+ $entropy_{45^\circ}$ + $entropy_{90^\circ}$ + $entropy_{135^\circ}$)/4

最後輸出 9 個數值

二、測試結果

從結果上可以發現，輸出長度為 3 的準確度都比較低，我認為這是因為除了要判斷攻擊外，還要識別出是哪一種攻擊方式，而輸出長度只有 1 的模型，只要有攻擊就可以輸出，所以事情不用做那麼多，準確度自然就比較高。與 CNN 的比較中，RandomForest 表現最佳，在只判斷是否有攻擊的情況下，準確度非常接近 CNN。

	CNN		RandomForest		Decision tree	
Output 長度為 1 (判斷是否有攻擊)	True positive	5965	True positive	5937	True positive	5897
	True negative	4031	True negative	4012	True negative	3948
	False positive	0	False positive	19	False positive	83
	False negative	4	False negative	32	False negative	72
	Accuracy	0.9996	Accuracy	0.9949	Accuracy	0.9845
	Precision	0.9996	Precision	0.9949	Precision	0.9845
	Recall	0.9996	Recall	0.9949	Recall	0.9845
	F1 Score	0.9996	F1 Score	0.9949	F1 Score	0.9845
Output 長度為 3 (要判斷哪種攻擊)	True positive	6949	True positive	6998	True positive	6987
	True negative	2983	True negative	2951	True negative	2933
	False positive	3	False positive	35	False positive	53
	False negative	65	False negative	16	False negative	27
	Accuracy	0.9932	Accuracy	0.9816	Accuracy	0.9679
	Precision	0.9995	Precision	0.9841	Precision	0.9655
	Recall	0.9907	Recall	0.9873	Recall	0.9789
	F1 Score	0.9951	F1 Score	0.9857	F1 Score	0.9722

Referenc:

方嘉祥,CNN based DDoS attack detection with raw packet data,國立成功大學資訊所碩士論文

https://blog.csdn.net/Jen_Hsueh/article/details/104754460

https://medium.com/@trapti.kalra/texture-analysis-with-deep-learning-for-improved-computer-vision-aa627c8bb133#b9ea

https://blog.csdn.net/zfjBIT/article/details/90638844