문서분류기법을 이용한 악성 스크립트 탐지

신대현[0] 위규범
아주대학교 정보통신전문대학원
{shigong[0], kbwee}@ajou.ac.kr

A Study On Malicious Script Detection using Text Categorization

Dae-Hyun Shin[0] Kyu-Bum Wee
Graduate School of Information and Communication, Ajou University

요 약

본 논문은 스크립트 호스트 모니터링을 통한 정보검색 기법인 유사도 알고리즘을 이용하는 악성 스크립트 탐지에 관한 연구이다. 스크립트 호스트의 모니터링을 통하여 스크립트가 실행되기 전에 스크립트를 가로채고 알려진 악성 스크립트와의 유사도를 비교하여 악성 여부를 판단한다. 소스기반의 빠른 탐지와 유사 변형의 악성 스크립트 탐지가 가능하며 악성행위의 종류를 사용자에게 보고할 수 있는 장점을 갖는다.

1. 서 론

2002년 1월 현재 비주얼 베이직 스크립트(VBS)와 자바 스크립트(JS)로 작성된 악성 스크립트는 1500개 이상이 발견 되었으며, 특히 VBS는 윈도우 98이후 운영체제에 기본적으로 탑재되었고 웹 브라우저뿐만 아니라 스크립트 호스트에 의해 독립적으로도 실행이 가능하다. VBS는 자기복제, 레지스트리의 생성, 수정, 삭제 그리고 네트워크 또는 MAPI를 이용한 메일 전송 등의 바이러스나 웜 제작에 필요한 기능들을 두루 갖추고 있어서 악성 스크립트 작성이 용이하고 그 기능이 강력하여 많은 피해를 입히고 있고 꾸준히 그 수가 증가하며 새로운 기법이 소개되고 있다[2,3,4].

그러나 이러한 스크립트들은 모두 wscript.exe라는 윈도우 기반 스크립트 호스트를 통해서 수행이 되기 때문에 스크립트 호스트의 모니터링을 통하여 실행되려는 VBS의 악성 여부를 판단하여 사용자에게 보고하고 그 실행을 중단할 수 있다.

따라서 본 논문은 스크립트 호스트 모니터링 기법과 악성 VBS의 악성행위 판단을 위한 유사도 알고리즘을 이용하여 악성 VBS를 탐지하고 그 실행을 차단할 수 있는 시스템을 제안한다.

제 2장에서는 악성스크립트의 기존 대응기법에 대해 소개한다. 재 3장에서는 시스템의 개요와 구조를 설명한다. 제4장에서는 시스템 구현에 사용된 정보검색 기법에 대해 설명한다. 제 5장에서는 실험 결과 및 분석을 제6장에서는 결론을 서술한다. 제 7장에서 향후 연구 방향을 서술한다.

2. 기존의 탐지기법

기존의 악성 스크립트에 대한 탐지방법은 이진 코드에 사용된 기법을 그대로 이용하거나 소스 코드 형태인 스크립트에 적합하도록 변형한 기법이 일반적으로 사용된다.

시그너쳐 인식: 특정 악성코드에만 존재하는 독특한 문자열을 검색하여 악성 여부를 진단하는 방법이며 현재에도 일반적으로 사용되는 기법이다[5]. 탐지 속도가 빠르며 악성임을 정확히 구별해내지만 알려지지 않은 스크립트에 대해서는 대응하지 못하며 시그너쳐의 추출이 문제점이 되고 있다.

휴리스틱 분석은 정적분석과 동적 분석으로 분류할 수 있다.

정적 분석기법은 대상 스크립트를 검사하여 악성행위에 관련된 함수나 메서드의 호출 회수를 근거로 하여 악성임을 판단하는 방법으로서[6], 알려지지 않은 악성 스크립트의 탐지에 사용될 수 있으며 비교적 빠른 속도와 높은 탐지율을 갖지만 긍정오류가 큰 것이 단점이다.

동적 분석기법은 에뮬레이션을 이용하는 방법으로서 스크립트의 실제 수행 중에 발생하는 시스템 콜 및 자원의 변화를 참조하여 정확한 탐지가 가능하지만 그 구현이 매우 어렵고 부하가 높은 것으로 알려져 있는 방법이다[7].

3. 시스템의 개요와 구조

3.1 시스템의 개요

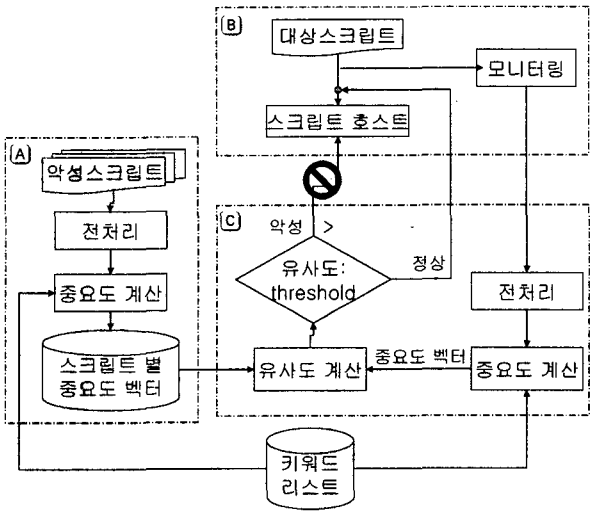
3.1.1 본 시스템은 실시간 스크립트 호스트의 모니터링을 통해 스크립트가 실행되기 전에 해당 스크립트의 악성 여부를 판단하여 실행을 차단하는 것을 목적으로 한다.

3.1.2 악성 여부의 판단은 보유하고 있는 악성 VBS의 샘플과 탐지 대상이 되는 VBS의 유사 정도를 근거로 한다.

3.1.3 암호화된 VBS는 탐지대상에서 제외한다. 암호

화된 VBS의 처리는 정적분석 기법과정에서 암호화 여부의 판단, 복호화 루틴의 추출 작업을 추가적으로 요구하며 본 논문에서는 다루지 않는다.

3.2 시스템의 구조



<그림 1> 시스템의 구조

본 시스템은 그 기능에 근거하여 Ⓐ,Ⓑ,Ⓒ의 세부분으로 구성되며 시나리오는 다음과 같다.

3.2.1 <그림 1>에서 Ⓐ는 수집된 악성 VBS로부터 중요도 벡터를 계산하는 부분이다. 악성 VBS는 전처리 과정을 통해 주석과 객체 변수가 아닌 변수의 선언 등 빈도수 계산에 불필요한 요소를 제거한다.
키워드 리스트를 참조하여 VBS에 포함되어있는 키워드의 빈도를 구하고 키워드별 중요도를 계산하여 저장한다.

3.2.2 Ⓑ는 스크립트 호스트를 모니터하는 부분으로 대상 VBS가 스크립트 호스트에 의해 실행되기 전에 가로채어 Ⓒ로 전달하게 되고, 악성여부 판단결과에 따라 대상 VBS의 흐름을 변경한다.

3.2.3 Ⓒ는 대상 VBS의 악성 여부를 판단하는 부분으로 전처리 과정을 거쳐 대상 VBS의 키워드별 중요도를 구하고 왼쪽부분에서 저장된 VBS별 중요도 벡터와 함께 유사도를 계산한다. 정상 VBS인 경우 대상 VBS가 실행되도록 스크립트 호스트에게 대상 VBS를 넘겨주고, 악성인 경우 사용자에게 보고하고 대상 VBS의 실행을 막는다.

4. 정보 검색

4.1 정보 검색 모델

정보 검색 모델에서 문서는 색인어라고 불리는 키워드 집합으로 표현되며 이러한 키워드는 문서 내용을 색인하고 요약하는데 사용된다. 본 논문의 실험에서는 키워드-중요도를 고차원 공간에 벡터로 표현하는 벡터 스페이스 모델을 적용하였다[8].

4.1.1 키워드-중요도 할당 기법으로 tf-idf(term frequency-inverse document frequency)기법을 사용하며, 다음 식(수식 1)을 사용하거나 이 식의 변형을(수식 2) 사용한다.

$$weight(i,j) = (tf_{i,j})log\left(\frac{N}{n_j}\right) \quad \text{<수식 1>}$$

$$weight(i,j) = \left(\frac{tf_{i,j}}{tot\_freq_i}\right)\log\left(\frac{N}{n_j}\right) \quad \text{<수식 2>}$$

weight(i,j):	i번째 문서에서 j번째 키워드가 갖는 중요도
i:	문서의 인덱스
j:	키워드의 인덱스
$tf_{i,j}$:	i번째 문서에서 j번째 키워드의 빈도
n_j:	j번째 키워드가 포함된 문서의 수
N:	전체 문서의 수
tot_freq_i:	i번째 문서에 나타나는 모든 키워드의 수

4.1.2 벡터 스페이스 모델에서 문서와 질의 문서간의 유사정도를 측정하기위한 방법으로 코사인계수를 이용한다<수식 3>. 이 값은 0에서 1사이의 값을 가지며, 1에 근접할수록 색인문서-질의문서 간의 유사성이 높다는 것을 나타낸다.

$$\cos(q,d) = \frac{\sum_{i=1}^{n} q_i d_i}{\sqrt{\sum_{i=1}^{n} q_i^2}\sqrt{\sum_{i=1}^{n} d_i^2}} \quad \text{<수식 3>}$$

q: 질의 문서
d: 색인 문서
n: 키워드의 수
q_i: 질의문서가 갖는 i번째 키워드의 중요도
d_i: 색인문서가 갖는 i번째 키워드의 중요도

5. 실험 결과 및 분석

5.1 dataset

구분	개수
정상 VBS	332
악성 VBS	72
미탐지 VBS	24
합계	428

<표 1> 실험에 사용된 dataset

실험에 사용된 data는 하우리와 안철수연구소의 바이러스 스캐너를 이용하여 정상과 악성으로 구분하였고, 악성 VBS에는 웜 생성기에 의한 악성VBS 8개를 포함하고 있다. 미탐지 VBS는 바이러스 스캔 결과 정상으로

판단되었지만 악성행위를 갖고 있는 VBS이다.

5.2 실험결과 및 분석

TP(True Positive): 악성을 악성으로 판단
TN(True Negative): 정상을 정상으로 판단
FP(False Positive): 정상을 악성으로 판단
FN(False Negative): 악성을 정상으로 판단

Detection rate = TP/(TP+FN)
FP rate = FP/(TN+FP)
Overall accuracy = (TP+TN)/(TP+TN+FP+FN)

5.2.1 실험

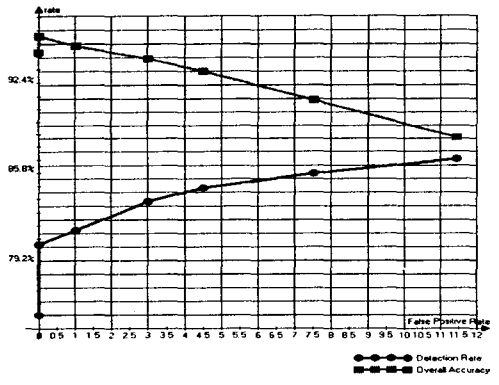
Threshold	TP	TN	FP	FN	Detection Rate	FP Rate	Overall Accuracy
0.70	83	294	38	13	86.45%	11.44%	88.08%
0.75	82	307	25	14	85.41%	7.53%	90.88%
0.80	81	317	15	15	84.37%	4.51%	92.99%
0.85	80	322	10	16	83.33%	3.01%	93.92%
0.90	78	328	4	18	81.25%	1.01%	94.85%
0.95	77	332	0	19	80.20%	0.00%	95.56%
1.00	72	332	0	24	75.00%	0.00%	94.39%

<표 2> 실험결과

실험은 유사도를 나타내는 지표로 코사인계수와 벡터간 거리, 키워드-중요도 할당 방법의 변형, 키워드 리스트에 포함된 키워드의 수와 악성에만 등장하는 혹은 정상에만 나타나는 키워드 등을 고려하여 수행되었다.

<표 2>실험결과는 코사인계수로 유사도를 계산하고, <수식 2>를 키워드-중요도 할당에 적용하였다. 그리고 악성VBS와 정상VBS에서 1회 이상의 빈도를 보인 키워드 중 악성행위에 오용될 수 있는 객체의 메서드와[9] 레지스트리키 그리고 내장함수로부터 193개를 선택하여 사용한 결과이다.

5.3 분석



<그림 2> 실험결과

악성VBS의 키워드만 리스트에 포함한 경우는 긍정 오류의 비율이 커지고, 반대의 경우에는 탐지율에 영향을 미쳤다. 특정 VBS 한두 개가 긍정오류의 대부분을 발생시키는 것으로 나타났다. 미탐지 VBS에 대해 평균 34%, 알려진 악성VBS에 대해서는 100%의 탐지율을 보였다. 실험결과 threshold는 0.94~0.97에서 95.56%의 가장 높은 정확도를 보였다.

6. 결론

본 논문에서는 유사도를 이용한 악성VBS의 탐지와 그 실행의 차단이 가능한 시스템을 제안하였다. 정확도의 극대화와 긍정오류의 최소화를 위해 키워드-중요도 할당 방법, 키워드 집합을 달리하여 실험을 수행하였고 95% 이상의 정확도를 얻었다.

7. 향후 연구 방향

실험 결과 정확도에 가장 큰 영향을 미치는 요소는 VBS의 내용을 색인하는 키워드의 리스트였다. 키워드의 선택방법과 그 검증에 대한 연구가[10,11] 필요하고, 분석결과 긍정오류의 대부분을 발생시키는 특정 스크립트에 대한 처리도 문제점으로 남아있다. 더욱 객관적인 성능 평가를 위해 더 큰 dataset에 대한 실험을 계획하고 있다.

8. 참고문헌

[1] 문제근, "IR 기반의 유사도 알고리즘을 이용한 침입탐지 기법 연구",아주대학교, 2002.
[2] CERT coordination center. http://www.cert.org
[3] 안철수 연구소, http://www.ahnlab.com
[4] 하우리, http://hauri.co.kr
[5] S. Kumar, E. H.Spafford, "A Generic Virus Scanner in C++", Purdue Unuiversity Technical Report CSD-TR-92-062, 1992.9.
[6] Symantec AntiVirus Research Center, "Understanding Heuristics", Symantec White Paper, 1998.
[7] B. Le Charlier, M Swimmer, A. Mounji, "Dynamic detection and classification of computer viruses using general behaviour patterns", Fifth Internal Virus Bulletin Conference, Boston, 1995.
[8] B. Yates, R. Neto, "Modern Information Retrieval", Addison-Wesley, 1999.
[9] M. Kennedy, "Script-Based Mobile Threats", Symantec White Paper, 2000.
[10] S. M. Weiss, et al., "Maximizing Text-Mining Performance" IEEE Intelligent System, 1999.
[11] Y. Yang, J. P. Pedersen, "A Comparative Study on Feature Selection in Text Categorization", The 14th International Conference on Machine Learning, 1997.