현재 우리는 취약점 위주로 진행

분석 이유: 정적분석으로 잡아내기 위해 공격이 어떻게 이뤄지는지 파악하기 위해서 분석.

접근 방법들

* 방법 1.   
  대부분의 악성문서들의 매크로 기반은 parser를 통해서 스크립트를 추출해서 스크립트에 대해 악성 여부를 판단해서 결과를 준다.  
  ex) xls를 통해 vbs코드를 추출가능
* 방법 2.  
  실제적인 문서의 파싱을 하다 취약점이 발생  
  parser를 구해서 파싱을 하다가 문제가 있는 파일일 경우 취약점으로 판단해서 점수를 매긴다.

주된 공격 방법

Script를 기반으로한 공격

Ex) Pdf도Parsing을 하다보면 에러가 나면 의심을 한다.

이 에러에는 대부분 JavaScript Code가 있다.

해당 코드 역할이 Exploit 역할을 한다. 🡨 Parser들은 많다.

그러나 해당 JavaScript Code에는 obfuscation이 되어있는다.

이러한 obfuscation을 해제해 분석해서 보니 악성 공격을 한다면 악성 pdf로 결론을 내려 알림을 띄운다.

따라서 중요점은

1. Parser
2. Obfuscation이 된 JavaScript Code, VBS 등 코드에서 악성여부 판단.

Application Virtualization(전상현 멘토님관련)

실제적으로 문서의 Parser를 만들어서 컴포넌트를 다 떼서 보고 정확하게 이것이 정상적으로 Parsing이 되는지에 대해서 판단하고 안된다면 악성으로 규정한다 생각했음.

그러나 단순히 Parsing만 안된다해서 악성은 아니다.

🡪 우리의 Parser도 완벽하지 않고 구현하는 입장이다.

🡺 에러가 있을 수 있다.

따라서 Parsing 결과와 스코어링을 통해 악성여부의 심각도를 알려줘야 할 것 같다.

🡺 해당 부분은 성민이와 Brainstorming을 해야 한다.

취약점을 트리깅할 때 주로 많이 사용하는 Component를 정확하게 Parsing을 해낼 수 있어야 한다.

BoB 에서 누군가 한글 취약점을 찾기위해 만든 Parser가 있는 것 같다.

이를 베이스로 해서 제작할 수 있을 듯하다.

워드나 엑셀은 오픈 오피스가 있음

완벽하게 MS office를 지원하지 않지만 xlxs나 다양한 부분은 지원하기에 해당 부분에서 Parser 부분만 추출해서 Parser를 만들 수 있는지 파악해봐야 한다.

논문을 쓰기위해서는 기존 도구들의 문제점이 무엇인지 파악해야 한다.

기존의 엔진이 어떻게 되어있는지 파악해야 한다.

우리가 내세우는 것은 Unknown임.

🡪 해당 엔진 분석에 너무 시간을 쓰지 말자.

Ex) 기존은 해쉬보고 파싱하다 안되면 악성인지 본다 정도?

논문의 흐름은 그것보다 더 우리는 현재 유행하고 있는 문서를 분석해보니 Script가 많아서 Script가 100%에 대해서 문서가 꼬여있더라도 추출이 가능하다.

그래서 Script 추세를 보면 어려 조각으로 쪼개서 쉽게 분석하지 못하게 한다.

Obfuscation도 포함.

따라서 간단 obfuscation 디코딩 포함 자동 악성 분석으로 접근해도 될 듯하다.

우리는 Unknown이라는 것이 핵심 키워드이다.

따라서 새로운 악성코드를 만든다.

기존에 hash 값이 없는 것을 만들고 그것을 테스트 삼아서 기존 공개된 문서 악성코드를 돌려보니 못잡는다.

카스퍼스키, 메이저 백신들을 자원을 신청해서 구매해서 최신 DB상에서 새로운 공격 악성문서를 만들어 보고 이전은 탐지를 못하지만 우리는 가능하다 라고 주장해도 좋다.

문서의 우선순위를 잘 둬야 한다.

한글은 파서도 없기에 Parser를 새로 만드는 것에 시간이 오래 걸린다.

있는 것을 최대한 사용해서 우리의 아이디어를 넣자.

한글 Parser는 Github에 자료가 좀 있다.

Office는 OpenOffice에 코드가 나와있다.

Xlxs는 zip기반이므로 다 풀린다.

Doc, ppt 등등은 Parser를 만들기 어렵다.

주로 공격되는 문서를 파악하고 해당 문서에 특화된 것으로 전략적으로 해보자.

기존의 한글 탐지는 한글내부에 JavaScript 코드가 있는지 확인한다.

🡪 일반인이 JavaScript 코드를 사용하진 않기 때문이다.

최근 새로운 방법은 한글의 Puzzer를 통해 Parser를 만들어서 악성코드를 넣었다.

🡺 한글의 선택은 비중을 낮추는 것이 좋다.

Pdf Parser는 매우 많고

Pdf가 tagging위주로 되어있다.

< or Tag가 있다.

연구도 많이 됬고 자료도 되었다.

요즘에 pdf로 공격을 많이 한다.

고도화로 pdf 내에 Font Parser를 만들어서 Parsing이 제대로 안된다면 악성코드로 포함된 악성문서로 판단이 가능하다.

따라서 어디까지 해서 악성인지 판단할 것인지 정하자.

따라서 우리가 선택한 문서 + 해당 문서의 OLE 객체가 품을 수 있는 객체에 대한 Parser도 추가하면 완벽한 정적분석 엔진이다.

Font는 Font tools Parser가 있다.

Opensource.

1. 어디까지 정적분석을 통해서 취약점을 분석할지
2. 문서의 범위와 종류는?