남성엽 멘토님이 해주신 말들

참어렵습니다.

단기간에 무언가 모든 문제를 해결할 수준으로 올라가기 힘들다.

취약점이라고 생각하면

Asec.ahnlab.com/en/28877/

우리가 보는 것이 문서 취약점이니 악성문서를 보는 것이 쉽다.

취약점까지는 난독화, 안티리버싱이 안들어가 있다.

취약점은 목표가 되는 문서를 여는 doc에서 발생하는 것이라 손을 댈수 없다.

동작해 다운로드를 받아 실행됨.

받아온 악성코드는 안티 디버깅, 리버싱하기 힘들다.

이부분은 덜어내고 생각하는 것 만으로 좋다.

문서 취약점에 포커스 둬야한다.

취약점 분석에 가장 중요한 것은

첫번째로 취약점을 돌 가상의 환경을 만들어라.

취약점이 돌지 않으면 어디서 동작하는지 파악하기 힘들다.

건덕지가 없다.

무조껀 취약점이 어디서 동작하는 지 확인하고 환경을 가상으로 만들고 테스트를 해야한다.

환경은?

보고서를 보면 취약점을 패치하면 CVE번호를 받음.

이 번호를 검색하면 MS에서 패치했다고 띄워줌.

아래에 보면 패치 전의 버전은 먹는다는 것을 암.

워드나 파워포인트 버전에 맞춰 환경을 구성해야 한다.

가상환경을 구축하면된다.

일종의 인큐베이팅임.

취약점을 잘동작하는 인큐베이터를 만들고 거기서 시작한다.

해당 취약점을 중심에 두고 살펴본다.

악성 문서파일도 구했으면 그것을 동작시킬텐대

그 문서를 동작하면서 네트워크 끊어두고 실행하여 취약점이 먹는지 확인한다.

실제 악성파일로 하는 것은 부담스럽다.

네트워크를 다끊어둔다.

스냅샷을 찍어둔다.

분석 전에 스냅샷을 찍어야 한다. 🡪 깨끗한 환경을 찍어야함.

일단 먼저 취약점에 대해 알아야한다.

도대체 어떤 취약점인지 알아야 분석이 가능하다.

정보를 다 뒤집어 봐야한다.

트위터도 찾아보기.

Xml에 링크 집어 두기인거 같음 (화면)

Docx를 확장지를 zip으로 만들면 풀 수 있다.

해당 취약점이 동작하는 xml이 어디있는지 알아야 한다.

그러나 대부분 힌트가 있다.

풀어서 검색해보기.

내가 만들 수 있나?

웹서버를 만들어서 내껄로 만들어서 zip으로 docx를 만들어야 한다.

🡪 해당 취약점은 논리 취약점이다.

로컬 웹서버를 만들고 여기로 돌려지는지 확인.

파이썬으로 만들기.

Vm에 nginx나 파이썬으로 만들어서 테스트.

서버에 계산기나 텍스트 파일을 올려두고 이후 internal download에 남을 것이다.

흔적이라 생각되는 파일을 올려두기.

취약점이 트리거 되는 환경을 제대로 분석해야 한다.

거기서부터 시작해야 한다.

취약점 자체에 대해서 딥하게 해야 한다.

CVE-2022-30190

워드의 매크로나 컴포넌트 기능을 사용해 다른곳에 접속하게 하는 것임.

수업시간에 했던 윈도우 iso관련해서 업데이트 안하고 하면 된다.

5월전까지 먹는 취약점이다.

샘플을 구해서 돌려 봐야한다.

어제 이야기했던 템플릿 인젝션에 해당하는 것에 집어넣은 것이다.

시작은 템플릿 인젝션인것 같음.

Msdt의 역할이 뭔지 확인해야 한다.

Sysinternal에 들어가면 동적 분석 도구가 많이 있다.

프로세스 익스플로러와 모니터를 주로 사용한다.

Msdt프로세스니 msdt프로세스에 필터를 걸어서 봐서 힌트를 알 수 있다.

악성코드가 들어오면 보고 CVE 를 봐서 조사함.

악성코드가 동작할 수 있는 가상환경이 설치되어있음.

시작으로 정적으로 디버거 붙여 확인

각종 모니터링 툴을 통해 프로세스 액티비티가 무엇인지 확인.

프로세스 모니터에 msdt가 생성되고 실행되는 것을 볼 수 있음.

프로세스 모니터는 악성코드가 동작할 때 외부로 표출되는 것을 통해 볼 수 있음.

운좋으면 이 취약한 문서에서 다운로드 받거나 다른 행위를 함.

우리는 취약점이 궁금하니 인자정보를 볼 것임.

속된 동적분석을 돌려봐야 한다.

1차적으로 모니터링 툴을 열고

스냅샷고 찍어야 함.

보통 안나오면 정적 분석을 한다.

Msdt가 실행할 때 인자가 무엇인지 보면 느낌이 온다.

워드자체에 권한이 낮으니 권한을 높일 방법이 필요해서 msdt를 사용했다.

Msdt가 높은 권한이고 word는 낮은 권한인데 낮은 권한을 가진 놈이 높은 권한을 가진이에게 쿼리를 날렸는데 검증을 안하고 실행.

권한상승을 목표로한 취약점이 먹힌것 같다.

파라미터러 커멘드를 줘야하지만 파라미터를 조작해서 거기에 악성 파일을 열거나 변경하면 파일을 높은 권한으로 연다.

파라미터가 어떻게 열리는지 봐야한다.

🡪 동적 분석을 해야 한다.

프로세스 모니터를 보자

작년 1월까지 패치된 상황이므로 vm에 워드만깔아서 보면 된다.

필터에서 프로세스 name에 msdt를 삽입.

Msdt활용은 워드를 까보면 된다.

워드에 없으면 템플릿 외부 링크 스크립트 안에 흔적이 있을 것이다.

논리 취약점으로 보인다.

Zip에서 Docx로 변경이 가능하다.

Docx계열인지 확인하다.

어제 이야기한 템플릿 인젝션이 핵심일 수 있다.

Rels 밑에 악성 url을 있는지 확인

폰트는 파이썬으로 파싱할 수 있음

폰트가 깨지면 의심을 한다. 🡪 폰트취약점을 노렸다 볼 수 있다.

문서 취약점에 중점을 두자

취약점에 관련된 문서가 있으면 열 필요 없다.

취약점까지는 가능하나 뒷부분은 난독화가 있어 다르게 해야 한다.

Msdt의 취약점이 뭔지.

권한상승으로 연거같은데.

템플릿 인젝션을 통한 파일을 열기위해 사용하는 지 모름.

리눅스 커멘드 인젝션 처럼 파라미터에 원하는 스크립트 경로를 집어넣어 실행하는지 확인

문서 파일도 뒤지는 것도 중요하지만

Msdt가 어떻게 동작하는지 확인.

CVE번호로 검색해서 많은 정보를 찾아보자.

공격그룹에 대해 볼 필요 없음

분석 보고서에서 트리거 되는 부분에 대해서 만 확인하기.

포커스를 맞추자.

IDA나 windbg가 사용가능하고 동적분석이 가능하면 취약점 정보를 빨리 캐치할 수 있는 연습이 필요하다.

쉽게 문제 푸는게 좋으니

서버의 앱가상화 앞단에서zip으로 풀 수 있음.

이런 것들이 있음.

패턴으로 제거할 수 있음.

템플릿 인젝션의 케이스 비중은?

공격이 템플릿 인젝션인지 확인해봐야한다.

공격이 많아지면 해당부분에 중점을 두면 된다.

2021-40444

취약점 중심으로 악성문서에서 취약점을 어떻게 활용하는지, 취약점의 목적은?

2022-30

은

40444는 외부의 스크립트 파일을 다운받아 http접속을 우회하기 위해서 사용함.

악성문서의 msdt의 흔적이나 어떻게 동작하는지 확인.