**Kali Linux Tutorial Mission 2**

두 번째 미션이다. Target 시스템을 VMWare에 load하고 가장 기본적인 주소와 open port에 대해 검색을 하였다. Ping scan(-sn)을 통하여 VM 이름과 일치하는 주소를 가진 host를 찾을 수 있었으며 주소의 경우 192.168.219.67의 주소를 가지고 있었다.

다음으로는 nmap을 사용하여 port scan을 진행하였다. Port scan의 범위는 기본값으로 진행하였으며 tcp 8080 port만 열려있었다. 처음에는 열려있는 port가 너무 적길래 범위를 1부터 65535까지 지정하고 다시 하였음에도 결과에 변화는 없었다.

우선 http://192.168.219.67:8080으로 접속하였을 경우 간단한 웹 페이지를 만날 수 있었다. 기본 화면의 경우 간단한 화면 문구가 있었고 페이지 중앙 부분에 Powered by Struts라고 흔히 볼 수 있는 웹 제공자 문구가 있었는데 해당 문구에 마우스를 올릴 경우 하이퍼링크로 연결된 다른 페이지를 볼 수 있었다.

nessus를 사용해서 취약점을 확인한 결과 apache tomcat이 동작 중이었으며 tomcat의 기본 파일(eg. Manager) 등이 존재한다고 하였다.

Nmap Script Engine : 스크립트를 통해 취약점이 exploit 가능한지 파악

기본적으로 /*usr/share/nmap/scripts* 경로에 있음.

Powered by struts라는 문구를 아까 볼 수 있었는데 이를 토대로 apache struts의 취약점을 이용한 공격을 하는 것으로 보임. 취약점이 꽤 있는 것으로 보이는데 이 중 CVE-2017-5638을 이용하여 진행하였다.

Nmap script engine을 사용할 때 사용할 스크립트의 경우 --script=[file name]을 통해서 지정할 수 있고 스크립트 실행 시 사용할 arguments의 경우 --script-args [argument name]=[value]를 통해서 지정할 수 있다. script의 argument 목록의 경우 <https://nmap.org/nsedoc/> 페이지에서 확인하고자 하는 스크립트 명에 해당하는 항목을 클릭하면 관련된 자세한 내용과 사용 가능한 arguments를 확인할 수 있다.

Metasploit에서 해당 CVE에 해당하는 exploit module을 찾을 수 있었다. 찾는 방법의 경우 search cve:[CVE 뒤 4자리] 를 통해서 쉽게 검색할 수 있다. Exploit의 과정은 CVE-2017-5638 문서에 별도로 정리하였다. 여튼 reverse shell을 취득할 수 있었는데 운이 좋게도 바로 root 권한을 취득할 수 있었다.

Root 권한을 취득하는 데에는 성공하였지만 항상 shell 획득을 위해서는 exploit을 해야하므로(영상에서는 post-exploitation 이라고 하였음) 어딘가 마음에 들지 않나보다. 아무튼 netstat 명령어를 통해서 사용 가능한 port를 확인하는데 이전 nmap scan에서는 확인할 수 없던 mysql과 ssh 서비스가 동작 중인 것을 확인할 수 있었다.

Nmap scan을 통해서 확인할 수 없던 이유는 local에서만 접속 가능하도록 binding 되어 있기 때문으로 보인다. mysql의 경우 127.0.0.1:3306으로 설정되어 있었다. 이는 원격에서는 접속할 수 없다는 의미이다.

이 강의에서는 파일을 수정하기 위해 interactive shell을 취득하려고 하는 것으로 보인다. 물론 cat과 sed를 이용하면 바꿀 수 있을 것으로 보이기도 하지만.. 암튼 reverse shell cheat sheet라고 검색해서 python script를 이용해서 취득하는 과정을 보여줬는데 그럼에도 파일 수정은 불가능하였다. Cheat sheet의 경우 다음 링크를 참조하였다.

(“[https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md](https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology and Resources/Reverse Shell Cheatsheet.md)”)

참고로 python script의 경우 python3로 바꿔주어야 정상적으로 실행이 가능했다. 이를 확인하는 방법은 python -V를 입력했을 때 버전 정보가 출력되면 수정할 필요가 없고 python3 -V로 입력해야 버전 정보가 출력되면 python3로 바꿔주어야 한다.

그럼에도 수정이 불가능하였기에 이 강의에서는 target server의 sshd\_config 내용을 복사해서 이를 attacker host에서 임의로 수정 후 다시 이 파일을 target server에서 다운로드 하는 식으로 진행하였다. 파일을 다시 다운로드 할 때는 attacker에서 web server를 동작시켜 wget 명령어로 가져왔다. 물론 실제로 공격할 때 이 방법을 썼다가는 그대로 자신의 정보를 노출시켜주지 않을까 싶다. 그래서 sed 명령어를 사용하는 법을 정리할 겸 해당 방법으로 진행해보았다. sed의 다양한 사용 방법의 경우 별도 문서로 작성하였다.

현재 sshd\_config 파일에서 수정할 부분은 PermitRootLogin의 prohibit-password를 yes로 바꿔주어야 했다. 이는 ssh를 사용하여 root로 로그인할 수 있도록 설정하기 위함으로 보였다. 따라서 아래 명령을 통해 값을 바꿔주고 서비스를 재시작해주었다.

sed -i “s/prohibit-password/yes/” /etc/ssh/sshd\_config

참고로 명령어 실행 과정에 있어서 sed: -e expression #1, char 53: unterminated `s' command

와 같은 에러 메시지를 확인할 수 있었는데 해당 오류는 s/대상 문자열/바꿀 문자열/[옵션] 의 구성으로 이루어져 있는데 옵션을 지정하지 않더라도 마지막 / 문자를 포함하지 않아서 발생했었다. 다른 해결 방법으로는 / 문자를 | 문자로 대체하여도 해결 가능하다고 하였다. 물론 일차적으로 위 양식이 맞는다는 가정 하에서..

root 계정으로 로그인이 가능하게 해주었는데도 nmap을 사용해서 port scan을 했을 때 filtered라고 표시되었다. 즉, SSH 설정에 있어서 무언가 있거나 혹은 방화벽 설정에 따른 결과로 보인다. 해당 문제를 해결하기 이전에 root 계정의 password도 알아내보기로 하였다. 물론 passwd 명령어를 사용하면 변경할 수야 있지만 그러면 침입한 흔적이 남게 될 수 밖에 없다. 그러나 현재 시스템에서는 root 계정이 lock으로 설정되어 있었던 지라 shadow 파일에서 패스워드 hash 값을 확인할 수 없었다. 기존 hash 부분이 !로 표시되어 있었으며 이와 관련해서는 아래 링크를 참조했다.

https://hotpotato.tistory.com/329

ssh의 경우 bind address가 0.0.0.0 인 것을 확인할 수 있었기 때문에 방화벽을 먼저 확인했다. 방화벽 규칙을 확인할 때 iptables로 찾으려 했었는데 우분투여서 그런지 ufw status로 확인해야 했었다.

강의 영상과 다른 점은 ufw status 명령을 통해 port 22번에 관한 내용을 찾을 수 없었다는 것이다. 찾아본 결과 ufw의 경우 기본적으로 허용되지 않은 모든 패킷을 차단하기 때문에 별도로 차단하지 않아도 filtered로 표시되는 것 같다. ufw에 관한 내용의 경우 아래 링크를 참고 했으며 ssh에 대해서 허용으로 바꿔준 뒤 다시 scan하니 open으로 되는 것을 확인할 수 있었다.

https://webdir.tistory.com/206

아쉽게도 root password의 경우 그냥 passwd 명령을 통해 직접 바꿔주었다. 만약 꼭 root 계정을 건들지 않아야 한다면 새로운 user를 생성 후에 그 user에 권한을 부여해서 해도 될 것 같긴하지만.. 어쨋든 정말 복잡한 세상인 것 같다.

ssh를 통해 완벽한 interactive shell로 접속할 수 있게 되었다. 그 다음으로는 database에 접속하고자 하는 것으로 보였다. mysql의 경우 127.0.0.1:3306으로 binding 되어 있기 때문에 현재로썬 local에서만 접속할 수 있다. 로그인 시도의 경우 역시 root 계정으로 시도하려고 하였으나, mysql의 password는 모르기 때문에 진행이 불가능했다.

강의에서는 같은 비밀번호를 여러 곳에서 사용할 경우 한 계정의 비밀번호만 취득해도 다른 계정으로 로그인 시도를 할 수 있는 점을 이용하려고 한 것 같다. Passwd 파일을 확인해 보았을 때 로그인 가능하고 비밀번호가 존재하는 계정은 webdev 계정이었으며 해당 계정에 대해 john the ripper를 통해 비밀번호를 알아냈다. John the ripper 사용의 경우 문서로도 정리했고 아래 링크도 잘 설명되어 있었다.

https://epicarts.tistory.com/56

그렇게 알아낸 비밀번호로 root 계정으로도 시도해보고 혹시나 싶어서 webdev로도 시도해봤는데 로그인이 되지 않았다. 그래서 강의를 보니 여기서는 hydra를 사용하여 crack하려고 하는 것 같았다. 여기서 중요한 말이 하나 있었는데 지금 과정의 경우 악의적인 목적으로 cracking하는 관점이기 때문에 hydra를 target system에 직접적으로 설치하지만 만약 ethical hacking일 경우 절대 무슨일이 있어도 이러면 안된다는 것이였다.

hydra의 사용법에 대해서는 따로 문서로 정리해야 할 것 같다. 핵심적인 옵션을 살펴보자면 -l의 경우 login할 username을 지정하며 -P의 경우 무차별 대입할 password를 file로부터 읽어들인다. 마지막으로는 서비스의 종류와 주소를 적어주면 된다. 처음에 시도하였을 때 강의처럼 결과가 안나와서 뭐가 문제인가 싶었는데 Dictionary file에서 문장 끝에 있던 공백 한 칸 때문에 제대로 대입이 안됐었다. 결과적으로는 공백을 모두 지우고 나니 성공했다.

데이터를 조회하는 거에 있어서는 mysql을 사용해본 적이 있기에 어렵지 않았으나 파일로 추출하는 과정은 새로이 알게 되었다. Dump 어쩌구 해서 mysqldump를 사용하는건가 싶었는데 그냥 명령줄 뒤에 into outfile [filename]을 통해서 file로 redirection했다.

영상에서는 파일로 추출한 뒤에 이를 apache tomcat의 web root directory에 올려서 다운받았는데 이 부분의 경우 scp 혹은 위에서 봤던 mysqldump 방법을 사용해보아도 좋을 것 같다.

추가적으로 into outfile을 사용할 때 강의와는 다른 오류를 볼 수 있었는데 그 오류는 다음과 같았다.

ERROR 1290 (HY000): The MySQL server is running with the --secure-file-priv option so it cannot execute this statement

해당 오류를 찾아본 결과 secure-file-priv 옵션으로 동작 중일 경우 지정된 위치에만 파일을 저장할 수 있다고 하였다. 지정된 위치의 경우 아래 명령어로 확인할 수 있다고 하였다.

Show variables like “secure\_file\_priv”;

위 명령어로 확인하였을 때 출력된 경로로 수정하여 시도하자 정상적으로 성공할 수 있었다. Mysqldump 명령어를 사용하여 export하는 방법의 경우 다음과 같았다. DB를 export하는 방법과 data를 export하는 방법이 달랐는데 먼저 DB를 export하는 방법의 경우 다음과 같았다.

Mysqldump -h <host> -u <user> -p <DB Name> > filename.sql

참고한 사이트의 경우 아래와 같으며 해당 링크에서 data를 export하는 법과 db, data import 방법도 기술되어 있다.

https://blog.opid.kr/370

이를 ssh와 연계하면 접속과 동시에 DB 정보를 local machine으로 export하는 것 또한 가능하다. ssh를 사용하여 로그인할 때 “”를 사용해서 접속 후 사용할 명령어를 지정할 수 있는데 export하는 과정의 명령어를 double quote에 입력하는 것이다. 여기서 한 가지 주의해야 할 점이 있는데 redirection 또한 묶어서 넣으면 local machine에 저장되지 않기 때문에 아래와 같은 형태로 고쳐주어야 한다.

Ssh <user>@<host> “mysqldump -h <host> -u <user> -p<password> <DB Name>” > filename.sql

-p 옵션 뒤에 password를 입력하지 않을 경우 ssh 로그인 시 암호와 mysql 접속 시 암호 총 두번을 입력해야 하기에 위와 같이 사용해주었다. 이렇게 사용할 경우 warning 경고가 뜨는데 동작에는 이상이 없으므로 신경쓰지 않아도 된다.