|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | KITRI 모의해킹 28기 | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | 작성: | | |  | |
|  | 최종프로젝트 매뉴얼 양식 | | | | | | |  |
|  | | OS Password Cracking | | | |  | | |
|  | |  | | | |  | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |
|  | | |  | |  | | | |

|  |
| --- |
| 1. Dictionary Attack |
| - 미리 생성된 pw사전(사용자 정보 등을 활용해 예상되는 비밀번호를 추려낸 데이터)를 이용해 패스워드를 알아내는 기법 |
|  |
| ① nmap으로 공격하려는 pc의 포트가 열려있는지 확인  root@localhost:~# nmap –p 21 –sV 192.168.0.75 |
|  |
| ② 계정 생성  root@localhost:~# useradd goseungwon  root@localhost:~# passwd goseungwon |
|  |
| ③ 사전파일 생성  root@localhost:~# cupp -I  >공격 대상 유저의 개인 정보 입력  >그 외 옵션들은 n으로 설정.  root@localhost:~# vi goseungwon.txt |
|  |
| ④만든 사전 파일로 hydra로 패스워드 크래킹  root@localhost:~# hydra –l goseungwon –P goseungwon.txt <ftp://192.168.0.75>  >hydra를 통해 사전에 제작한 pw사전을 활용한 사전대입공격 시 성공한 패스워드와 정보가 출력됨 |
|  |
| 보안 대책 |
| - 패스워드 설정에 있어, 개인정보 등을 활용하는 것을 지양한다.  - 패스워드를 가능한 한 길고 복잡하게 설정한다. |

|  |
| --- |
| 2. BRUTE FORCE |
| -Brute Force공격(무차별 대입 공격)은 패스워드에 모든 값을 대입해 찾아내는 기법으로써 성공확률은 100%지만 패스워드가 복잡해질수록 공격시간이 길어지고, 비용이 많이 들 수 있다. |
| ① 테스트 계정 생성  [root@CS7 home]# useradd goseungwon2  [root@CS7 home]# passwd goseungwon2 |
| >비밀번호는 테스트 목적 상 숫자 4자리로 생성. |
|  |
| ② hydra로 무차별 대입공격  root@localhost:~# hydra –l goseungwon2 –x 4:4:123456789 ftp://192.168.0.75 |
|  |
| 보안 대책 |
| - 패스워드를 가능한 한 길고 복잡하게 설정한다.  - 주기적으로 패스워드를 변경한다. |
|  |
|  |
|  |
|  |
| 3. Hybrid Attack |
| -앞서서 진행 했던 사전 대입 공격과 무차별 대입 공격을 혼합한 방식의 공격이다.  -문자열에 숫자나 심볼을 추가하여 패스워드에 대입함.  -공격 대상의 개인정보 등을 알고 있을 경우 매우 효율적인 공격이다. |
| ① 사전 파일 생성  root@localhost:~# cupp -i  > 하단의 옵션중 end of word에 y해주면 됨.(뒤쪽에 특수문자 기호 붙이는지, 랜덤숫자 붙이는지에 대한 옵션) |
|  |
| root@localhost:~# vi goseungwon.txt  >특수문자 등 문자열 뒤에 추가가 된 것을 볼 수 있다. |
|  |
| ②만든 사전 파일로 hydra로 패스워드 크래킹  root@localhost:~# hydra –l goseungwon –P goseungwon.txt <ftp://192.168.0.75>  >Dictionary attack처럼 만든 사전 파일으로 패스워드 크래깅하면 됨. |
|  |
| 보안대책 |
| -패스워드를 길고 복잡하게 설정하면 됨. |

|  |
| --- |
| 4)Rainbow Table |
| -하나의 패스워드에서 시작해 특정한 변환 함수로 변환된 형태의 여러 패스워드를 생성한다. 그리고 변환된 패스워드의 해시를 고리처럼 연결해 일정 수의 패스워드와 해시로 이뤄진 체인을 무수히 만들어 놓은 테이블을 말한다.  -해시를 크랙하는 속도를 희생해 룩업테이블을 작게만든다. 그래서 같은 공간에 더 많은 해시를 저장하고 뚫기 쉬워진다.  -존재하는 8자리까지의 모든 md5 해시를 크랙할 수 있다. |
| ① Shadow파일 탈취  root@localhost:~# gedit /etc/shadow  >공격 대상 PC 의 Shadow파일을 가져왔다고 가정하기 위해 정보를 가져옴. |
|  |
| ② 해쉬파일 해석  root@localhost:~# john centShadow.txt  >공격대상의 shadow정보를 복사해서 centShadow.txt.에 넣음 |
|  |
| ③공격 대상 PC에 ftp로 접속  root@localhost:~#ftp 192.168.0.75  >centShadow.txt에 담겨있는 Shadow정보를 복호화하여 사용자의 암호를 알아낸다. |
|  |
| 보안 대책 |
| -Salt라는 랜덤한 문자열을 원래 패스워드 앞에 붙여 레인보우 테이블을 이용한 패스워드 크래킹을 못하도록 막는다.  -암호화 알고리즘을 최신것으로 바꾼다.  **암호화 알고리즘 변경 방법 :** |
|  |
|  |
|  |
| ① 현재 암호화 알고리즘이 뭐인지 확인  [root@c7 ~]# authconfig --test | hashing  [root@c7 ~]# authconfig --test | password  [root@c7 ~]# tail /etc/login.defs |
|  |
| ② 테스트할 유저 추가  [root@c7 ~]# useradd test2  [root@c7 ~]# passwd test2  ③ 암호화 알고리즘 변경  [root@c7 ~]# authconfig –passalge=sha256 –update  > $6에서 $5로 변했다는 것은 알고리즘이 변경된게 적용이 되었다는 것이다. |