주요정보통신기반시설

기술적 취약점 분석·평가 프로젝트

IaC 기반 하이브리드 클라우드 인프라 및 보안 솔루션 구축 1기 한덕관

목차

1.계정 관리

U-01 (상)	1. 계정관리 > 1.1 root 계정 원격접속 제한
취약점 개요	
점검내용	■ 시스템 정책에 root 계정의 원격터미널 접속차단 설정이 적용되어 있는지 점검
점검목적	■ 관리자계정 탈취로 인한 시스템 장악을 방지하기 위해 외부 비인가자의 root 계정 접근 시도를 원천적으로 차단하기 위함
보안위협	■ root 계정은 운영체제의 모든기능을 설정 및 변경이 가능하여(프로세스, 커널변경 등) root 계정을 탈취하여 외부에서 원격을 이용한 시스템 장악 및 각종 공격으로(무작위 대입 공격) 인한 root 계정 사용 불가 위협
참고	※ **root 계정:** 여러 사용자가 사용하는 컴퓨터에서 모든 기능을 관리할 수 있는 총괄권한을 가진 유일한 특별 계정. 유닉스 시스템의 루트(root)는 시스템 관리자인 운용 관리자(Super User)로서 윈도우의 Administrator 보다 높은 System 계정에 해당하며, 사용자 계정을 생성하거나 소프트웨어를 설치하고, 환경 및 설정을 변경하거나 시스템의 동작을 감시 및 제어할 수 있음 ※ **무작위 대입 공격(Brute Force Attack):** 특정한 암호를 풀기 위해 가능한 모든 값을 대입하는 공격 방법 ※ **사전 대입 공격(Dictionary Attack):** 사전에 있는 단어를 입력하여 암호를 알아내거나 암호를 해독하는 데 사용되는 컴퓨터 공격 방법

```
---

- name: Remove Vulnerability U-01
  hosts: all
  tasks:
    - name: Remove Package - telnet-server
      ansible.builtin.package:
        name: telnet-server
        state: absent

    - name: Replace File - /etc/ssh/sshd_config
      ansible.builtin.replace:
        path: /etc/ssh/sshd_config
        regexp: '^PermitRootLogin\s+yes\s*$'
        replace: 'PermitRootLogin prohibit-password'
      notify:
        - Restart Service

  handlers:
    - name: Restart Service
      ansible.builtin.service:
        name: sshd
        state: restarted
```

1.계정 관리

U-02 (상)	1. 계정관리 > 1.2 패스워드 복잡성 설정
취약점 개요	
점검내용	■ 시스템 정책에 사용자 계정(root 및 일반계정 모두 해당) 패스워드 복잡성 관련 설정이 되어 있는지 점검
점검목적	■ 패스워드 복잡성 관련 정책이 설정되어 있는지 점검하여 비인가자의 공격(무작위 대입 공격, 사전 대입 공격 등)에 대비가 되어 있는지 확인하기 위함
보안위협	■ 복잡성 설정이 되어있지 않은 패스워드는 사회공학적인 유추가 가능 할 수 있으며 암호화된 패스워드 해시값을 무작위 대입공격, 사전대입 공격 등으로 단시간에 패스워드 크랙이 가능함
참고	※ **패스워드 복잡성:** 사용자 패스워드 설정 시 영문(대문자, 소문자), 숫자, 특수문자가 혼합된 일정 길이 이상으로 패스워드를 설정하는 방법

```
---

- name: Remove Vulnerability U-02
  hosts: all
  vars:
    attrs:
      - name: minlen
        value: 8
      - name: ucredit
        value: -1
      - name: lcredit
        value: -1
      - name: dcredit
        value: -1
      - name: ocredit
        value: -1
      - name: difok
        value: 4
  tasks:
    - name: Replace File - /etc/security/pwquality.conf
      ansible.builtin.replace:
        path: /etc/security/pwquality.conf
        regexp: '^{{ item.name }}\s*\=\s*\-?\d+\s*$'
        replace: ''
      loop: "{{ vars.attrs }}"

    - name:
      ansible.builtin.blockinfile:
        path: /etc/security/pwquality.conf
        block: |
          {{ item.name }} = {{ item.value }}
        marker: ""
        create: yes
```

1.계정 관리

U-03 (상)	1. 계정관리 > 1.3 계정 잠금 임계값 설정
취약점 개요	
점검내용	▌사용자 계정 로그인 실패 시 계정잠금 임계값이 설정되어 있는지 점검
점검목적	▌계정탈취 목적의 무작위 대입 공격 시 해당 계정을 잠금하여 인증 요청에 응답하는 리소스 낭비를 차단하고 대입 공격으로 인한 비밀번호 노출 공격을 무력화하기 위함
보안위협	▌패스워드 탈취 공격(무작위 대입 공격, 사전 대입 공격, 추측 공격 등)의 인증 요청에 대해 설정된 패스워드와 일치 할 때까지 지속적으로 응답하여 해당 계정의 패스워드가 유출 될 수 있음
참고	※ 사용자 로그인 실패 임계 값: 시스템에 로그인 시 몇 번의 로그인 실패에 로그인을 차단할 것인지 결정하는 값

```
---

- name: Check Vulnerability U-03
  hosts: all
  become: yes

  vars:
    faillock_deny_attempts: 5
    faillock_unlock_time: 300
    faillock_fail_interval: 900
    faillock_root_unlock_time: 60

  tasks:
    - name: pam_faillock.so
      ansible.builtin.blockinfile:
        path: /etc/pam.d/system-auth
        marker: "# {mark} ANSIBLE MANAGED BLOCK - pam_faillock preauth"
        block: |
          auth        required      pam_faillock.so preauth audit deny={{ faillock_deny_attempts }} unlock_time={{ faillock_unlock_time }}
fail_interval={{ faillock_fail_interval }}
        insertbefore: "auth        sufficient    pam_unix.so"

    - name: pam_faillock.so
      ansible.builtin.blockinfile:
        path: /etc/pam.d/system-auth
        marker: "# {mark} ANSIBLE MANAGED BLOCK - pam_faillock authfail"
        block: |
          auth        [default=die] pam_faillock.so authfail audit deny={{ faillock_deny_attempts }} unlock_time={{ faillock_unlock_time }}
fail_interval={{ faillock_fail_interval }} root_unlock_time={{ faillock_root_unlock_time }}
        insertafter: "auth        sufficient    pam_unix.so"

    - name: pam_faillock.so
      ansible.builtin.blockinfile:
```

1.계정 관리

U-04 (상)	1. 계정관리 > 1.4 패스워드 파일 보호
취약점 개요	
점검내용	■ 시스템의 사용자 계정(root, 일반계정) 정보가 저장된 파일(예 /etc/passwd, /etc/shadow)에 사용자 계정 패스워드가 암호화되어 저장되어 있는지 점검
점검목적	■ 일부 오래된 시스템의 경우 /etc/passwd 파일에 패스워드가 평문으로 저장되므로 사용자 계정 패스워드가 암호화되어 저장되어 있는지 점검하여 비인가자의 패스워드 파일 접근 시에도 사용자 계정 패스워드가 안전하게 관리되고 있는지 확인하기 위함
보안위협	■ 사용자 계정 패스워드가 저장된 파일이 유출 또는 탈취 시 평문으로 저장된 패스워드 정보가 노출될 수 있음
참고	※ 관련 점검 항목 : U-07(상), U-08(상)

```
---


- name: Check Vulnerability U-04
  hosts: all
  tasks:
    - name: Read File - /etc/passwd
      ansible.builtin.slurp:
        src: /etc/passwd
      register: passwd


    - name: Decode conetnt & Regex
      ansible.builtin.set_fact:
        pwunconv: >-
          {{
            ((passwd.content | b64decode) | regex_search('^([\w-]+)[:](?!x[:]).+?[:]', '\0', multiline = true))
          }}


    - name: Execute Command - pwconv
      ansible.builtin.command:
        cmd: /usr/sbin/pwconv
      when: 'pwunconv != ""'
```

1.계정 관리

U-44 (중)	1. 계정관리 > 1.5 root 이외의 UID가 '0' 금지
취약점 개요	
점검내용	■ 사용자 계정 정보가 저장된 파일(예 /etc/passwd)에 root(UID=0) 계정과 동일한 UID(User Identification)를 가진 계정이 존재하는지 점검
점검목적	■ root 계정과 동일한 UID가 존재하는지 점검하여 root권한이 일반 사용자 계정이나 비인가자의 접근 위협에 안전하게 보호되고 있는지 확인하기 위함
보안위협	■ root 계정과 동일 UID가 설정되어 있는 일반사용자 계정도 root 권한을 부여받아 관리자가 실행 할 수 있는 모든 작업이 가능함(서비스 시작, 중지, 재부팅, root 권한 파일 편집 등) ■ root와 동일한 UID를 사용하므로 사용자 감사 추적 시 어려움이 발생함
참고	※ UID(User Identification): 여러 명의 사용자가 동시에 사용하는 시스템에서 사용자가 자신을 대표하기 위해 쓰는 이름

```
---

- name: Check Vulnerability U-44
  hosts: all
  tasks:
    - name: Read File - /etc/passwd
      ansible.builtin.slurp:
        src: /etc/passwd
      register: passwd

    - name: Decode conetnt & Regex
      ansible.builtin.set_fact:
        uid_0: >-
          {{
            ((passwd.content | b64decode) | regex_findall('^(?!root)([\w-]+)[:][^:]*[:]0[:]', '\1'))
          }}

    - name: Delete User - UID 0
      ansible.builtin.user:
        name: "{{ item }}"
        state: absent
        force: true
      loop: "{{ uid_0 }}"
```

1.계정 관리

U-45 (하)	1. 계정관리 > 1.6 root 계정 su 제한
취약점 개요	
점검내용	■ su 명령어 사용을 허용하는 사용자를 지정한 그룹이 설정되어 있는지 점검
점검목적	■ su 관련 그룹만 su 명령어 사용 권한이 부여되어 있는지 점검하여 su 그룹에 포함되지 않은 일반 사용자의 su 명령 사용을 원천적으로 차단하는지 확인하기 위함
보안위협	■ 무분별한 사용자 변경으로 타 사용자 소유의 파일을 변경 할 수 있으며 root 계정으로 변경하는 경우 관리자 권한을 획득 할 수 있음
참고	.

```
---

- name: Check Vulnerability U-45
  hosts: all
  become: false
  tasks:
    command: whoami
    register: result
  - debug:
      var: result.stdout
```

1.계정 관리

U-46 (중)	1. 계정관리 > 1.7 패스워드 최소 길이 설정
취약점 개요	
점검내용	■ 시스템 정책에 패스워드 최소(8자 이상) 길이 설정이 적용되어 있는 점검
점검목적	■ 패스워드 최소 길이 설정이 적용되어 있는지 점검하여 짧은(8자 미만) 패스워드 길이로 발생하는 취약점을 이용한 공격(무작위 대입 공격, 사전 대입 공격 등)에 대한 대비(사용자 패스워드 유출)가 되어 있는지 확인하기 위함
보안위협	■ 패스워드 문자열이 짧은 경우 유추가 가능 할 수 있으며 암호화된 패스워드 해시값을 무작위 대입공격, 사전대입 공격 등으로 단시간에 패스워드 크랙이 가능함
참고	※ 패스워드 최소길이를 8자리 이상으로 설정하여도 특수문자, 대소문자, 숫자를 혼합하여 사용하여함

```
---

- name: Check Vulnerability U-46
  hosts: your_servers
  become: yes

  vars:
    min_password_length: 8
    pam_config_file: /etc/pam.d/common-password

  tasks:
    - name:
      ansible.builtin.lineinfile:
        path: "{{ pam_config_file }}"
        regexp: '^(password\s+requisite\s+pam_pwquality\.so.*)(minlen=\d+)(\s.*)?$'
        replace: '\1minlen={{ min_password_length }}\3'
        state: present
        backup: yes

    - name:
      ansible.builtin.lineinfile:
        path: "{{ pam_config_file }}"
        regexp: '^(password\s+requisite\s+pam_pwquality\.so)(?!.*minlen=\d+)(.*)$'
        replace: '\1 minlen={{ min_password_length }}\2'
        state: present
        backup: yes
```

1.계정 관리

U-47 (중)	1. 계정관리 > 1.8 패스워드 최대 사용기간 설정
취약점 개요	
점검내용	■ 시스템 정책에 패스워드 최대(90일 이하) 사용기간 설정이 적용되어 있는지 점검
점검목적	■ 패스워드 최대 사용 기간 설정이 적용되어 있는지 점검하여 시스템 정책에서 사용자 계정의 장기간 패스워드 사용을 방지하고 있는지 확인하기 위함
보안위협	■ 패스워드 최대 사용기간을 설정하지 않은 경우 비인가자의 각종 공격(무작위 대입 공격, 사전 대입 공격 등)을 시도할 수 있는 기간 제한이 없으므로 공격자 입장에서는 장기적인 공격을 시행할 수 있어 시행한 기간에 비례하여 사용자 패스워드가 유출될 수 있는 확률이 증가함
참고	-

```yaml
---

- name: Check Vulnerability U-47
  hosts: all
  become: yes

  vars:
    max_password_age_days: 90

  tasks:
    - name: /etc/login.defs
      ansible.builtin.lineinfile:
        path: /etc/login.defs
        regexp: '^PASS_MAX_DAYS\s+\d+'
        line: 'PASS_MAX_DAYS {{ max_password_age_days }}'
        state: present
        backup: yes
```

1.계정 관리

U-48 (중)	1. 계정관리 > 1.9 패스워드 최소 사용기간 설정
취약점 개요	
점검내용	■ 시스템 정책에 패스워드 최소 사용기간 설정이 적용되어 있는지 점검
점검목적	■ 사용자가 자주 패스워드를 변경할 수 없도록 하고 관련 설정(최근 암호 기억)과 함께 시스템에 적용하여 패스워드 변경 전에 사용했던 패스워드를 재사용 할 수 없도록 방지하는지 확인하기 위함
보안위협	■ ※ 최소 사용기간이 설정되어 있지 않아 반복적으로 즉시 변경이 가능한 경우 이전 패스워드 기억 횟수를 설정하여도 반복적으로 즉시 변경하여 이전 패스워드로 설정이 가능함
참고	※ **최근 암호 기억**: 사용자가 현재 암호 또는 최근에 사용했던 암호와 동일한 새 암호를 만드는 것을 방지하는 설정. 예를 들어 값 1은 마지막 암호만 기억한다는 의미이며 값 5는 이전 암호 5개를 기억한다는 의미임

```yaml
---

- name: Check Vulnerability U-48
  hosts: all
  become: yes

  vars:
    min_password_min_days: 1

  tasks:
    - name: /etc/login.defs
      ansible.builtin.lineinfile:
        path: /etc/login.defs
        regexp: '^PASS_MIN_DAYS\s+\d+'
        line: 'PASS_MIN_DAYS {{ min_password_min_days }}'
        state: present
        backup: yes
```

1.계정 관리

U-49 (하)	1. 계정관리 > 1.10 불필요한 계정 제거
취약점 개요	
점검내용	■ 시스템 계정 중 불필요한 계정(퇴직, 전직, 휴직 등의 이유로 사용하지 않는 계정 및 장기적으로 사용하지 않는 계정 등)이 존재하는지 점검
점검목적	■ 불필요한 계정이 존재하는지 점검하여 관리되지 않은 계정에 의한 침입에 대비하는지 확인하기 위함
보안위협	■ 로그인이 가능하고 현재 사용하지 않는 불필요한 계정은 사용중인 계정보다 상대적으로 관리가 취약하여 공격자의 목표가 되어 계정이 탈취될 수 있음 ※ 퇴직, 전직, 휴직 등의 사유발생시 즉시 권한을 회수
참고	※ **Default 계정**: OS나 Package 설치 시 기본적으로 생성되는 계정(예 lp, uucp, nuucp 등) ※ 불필요한 default 계정 삭제 시 업무 영향도 파악 후 삭제 권고

```yaml
---

- name: Check Vulnerability U-49
  hosts: all
  become: yes

  vars:
    unnecessary_users:
      - lp
      - uucp
      - nuucp
      - games
      - sync
  tasks:
    - name:
      ansible.builtin.user:
        name: "{{ item }}"
        state: absent
        remove: yes
        force: no

      loop: "{{ unnecessary_users }}"
      ignore_errors: yes
```

1.계정 관리

U-50 (하)	1. 계정관리 > 1.11 관리자 그룹에 최소한의 계정 포함
취약점 개요	
점검내용	■ 시스템 관리자 그룹에 최소한(root 계정과 시스템 관리에 허용된 계정)의 계정만 존재하는지 점검
점검목적	■ 관리자 그룹에 최소한의 계정만 존재하는지 점검하여 불필요하게 권한이 남용되고 있는지 확인하기 위함
보안위협	■ 시스템을 관리하는 root 계정이 속한 그룹은 시스템 운영 파일에 대한 접근 권한이 부여되어 있으므로 해당 관리자 그룹에 속한 계정이 비인가자에게 유출될 경우 관리자 권한으로 시스템에 접근하여 계정 정보 유출, 환경설정 파일 및 디렉터리 변조 등의 위협이 존재함
참고	-

```
---

- name: Check Vulnerability U-50
  hosts: all
  become: yes

  vars:
    allowed_admin_users:
      - ansible_admin

  tasks:
    - name:
      ansible.builtin.set_fact:
        admin_group_name: "{{ 'wheel' if ansible_os_family == 'RedHat' else 'sudo' }}"

    - name:
      ansible.builtin.command: "getent group {{ admin_group_name }}"
      register: current_admin_group_info
      changed_when: false
      failed_when: current_admin_group_info.rc != 0 and 'no such group' not in
current_admin_group_info.stderr

    - name:
      ansible.builtin.set_fact:
        current_admin_members_list: "{{ current_admin_group_info.stdout.split(':')[3].split(',') |
default([]) }}"
      when: current_admin_group_info.stdout is defined and current_admin_group_info.stdout.split(':') |
length > 3 and current_admin_group_info.stdout.split(':')[3] is defined and
current_admin_group_info.stdout.split(':')[3] != ''

    - name:
      ansible.builtin.command: "gpasswd -d {{ item }} {{ admin_group_name }}"
      loop: "{{ current_admin_members_list }}"
```

1.계정 관리

U-51 (하)	1. 계정관리 > 1.12 계정이 존재하지 않는 GID 금지
취약점 개요	
점검내용	■ 그룹(예 /etc/group) 설정 파일에 불필요한 그룹(계정이 존재하지 않고 시스템 관리나 운용에 사용되지 않는 그룹, 계정이 존재하고 시스템 관리나 운용에 사용되지 않는 그룹 등)이 존재하는지 점검
점검목적	■ 시스템에 불필요한 그룹이 존재하는지 점검하여 불필요한 그룹의 소유권으로 설정되어 있는 파일의 노출에 의해 발생할 수 있는 위험에 대한 대비가 되어 있는지 확인하기 위함
보안위협	■ 계정이 존재하지 않는 그룹은 현재 사용되고 있는 그룹이 아닌 불필요한 그룹으로 삭제 조치가 필요함.
참고	※ **GID(Group Identification):** 다수의 사용자가 특정 개체를 공유할 수 있게 연계시키는 특정 그룹의 이름으로 주로 계정처리 목적으로 사용되며, 한 사용자는 여러 개의 GID를 가질 수 있음. ※ /etc/group 파일과 /etc/passwd 파일을 비교하여 점검하기를 권고함

```
- name:
    ansible.builtin.shell: |
      find {{ item }} -xdev -nogroup -print 2>/dev/null
    loop: "{{ search_paths }}"
    register: nogid_files_raw
    changed_when: false
    failed_when: false

  - name:
    ansible.builtin.set_fact:
      found_nogid_files: "{{ (found_nogid_files | default([])) +
item.stdout_lines }}"
    loop: "{{ nogid_files_raw.results }}"
    loop_control:
      label: "결과 처리 중: {{ item.item }}"
    when: item.stdout_lines is defined and item.stdout_lines | length > 0

  - name:
    ansible.builtin.set_fact:
      found_nogid_files: "{{ found_nogid_files | unique }}"

  - name:
    ansible.builtin.file:
```

```
---

- name: Check Vulnerability U-51
  hosts: your_servers
  become: yes

  vars:
    target_group_for_nogid_files: "nogroup"
    search_paths:
      - /
      - /var
      - /opt
      - /usr
      - /etc
      - /home

  tasks:
    - name:
      ansible.builtin.getent:
        database: group
        key: "{{ target_group_for_nogid_files }}"
      register: nogroup_check
      ignore_errors: true

    - name:
      ansible.builtin.set_fact:
        final_target_group: "{{ target_group_for_nogid_files }}"
      when: nogroup_check.rc == 0

    - name:
      ansible.builtin.set_fact:
        final_target_group: "root"
      when: nogroup_check.rc != 0
```

1.계정 관리

U-52 (중)	1. 계정관리 > 1.13 동일한 UID 금지
취약점 개요	
점검내용	■ /etc/passwd 파일 내 UID가 동일한 사용자 계정 존재 여부 점검
점검목적	■ UID가 동일한 사용자 계정을 점검함으로써 타 사용자 계정 소유의 파일 및 디렉터리로의 악의적 접근 예방 및 침해사고 시 명확한 감사추적을 목적으로 함
보안위협	■ 중복된 UID가 존재할 경우 시스템은 동일한 사용자로 인식하여 소유자의 권한이 중복되어 불필요한 권한이 부여되며 시스템 로그를 이용한 감사 추적 시 사용자가 구분되지 않음 (권한 할당은 그룹권한을 이용하여 운영)
참고	※ **UID**(User Identification): 여러 명의 사용자가 동시에 사용하는 시스템에서 사용자가 자신을 대표하기 위해 사용되는 식별 번호 ※ 패스워드 파일 수정 변경 및 신규 사용자 추가 시 UID가 동일한 계정이 존재하는지 확인해야 함(계정생성, UID 변경은 passwd 파일을 직접 편집 금지, 명령어를 이용하여 수정)

```
---
- name: Check Vulnerability U-52
  hosts: all
  become: yes

  tasks:
    - name:
      ansible.builtin.command: getent passwd
      register: passwd_output
      changed_when: false

    - name:
      ansible.builtin.set_fact:
        users_by_uid: |
          {% set user_map = {} %}
          {% for line in passwd_output.stdout_lines %}
            {% set parts = line.split(':') %}
            {% if parts | length >= 3 %}
              {% set username = parts[0] %}
              {% set uid = parts[2] | int %}
              {% if uid not in user_map %}
                {% set _ = user_map.update({uid: []}) %}
              {% endif %}
              {% set _ = user_map[uid].append(username) %}
```

```
        {% endif %}
      {% endfor %}
      {{ user_map }}

- name:
  ansible.builtin.set_fact:
    duplicate_uid_entries: |
      {% set duplicates = [] %}
      {% for uid, users in users_by_uid.items() %}
        {% if users | length > 1 %}
          {% set _ = duplicates.append({'uid': uid, 'users': users}) %}
        {% endif %}
      {% endfor %}
      {{ duplicates }}

- name:
  ansible.builtin.debug:
    msg: "중복 UID가 발견되었습니다 : {{ duplicate_uid_entries }}"
  when: duplicate_uid_entries | length > 0

- name:
  ansible.builtin.debug:
    msg: "중복 UID가 발견되지 않았습니다."
```

1.계정 관리

U-53 (하)	1. 계정관리 > 1.14 사용자 shell 점검
취약점 개요	
점검내용	■ 로그인이 불필요한 계정(adm, sys, daemon 등)에 쉘 부여 여부 점검
점검목적	■ 로그인이 불필요한 계정에 쉘 설정을 제거하여, 로그인이 필요하지 않은 계정을 통한 시스템 명령어를 실행하지 못하게 하기 위함
보안위협	■ 로그인이 불필요한 계정은 일반적으로 OS 설치 시 기본적으로 생성되는 계정으로 쉘이 설정되어 있을 경우, 공격자는 기본 계정들을 이용하여 시스템에 명령어를 실행 할 수 있음
참고	※ **쉘(Shell):** 대화형 사용자 인터페이스로써, 운영체제(OS) 가장 외곽계층에 존재하여 사용자의 명령어를 이해하고 실행함

```
- name:
    ansible.builtin.stat:
      path: "{{ item.shell }}"
    register: shell_check_results
    loop: "{{ users_to_check }}"
    loop_control:
      label: "{{ item.username }}"

  - name:
    ansible.builtin.debug:
      msg: |
        - 사용자: {{ item.item.username }}
        - 지정된 Shell: {{ item.item.shell }}
        - Shell 존재 여부: {{ '존재함' if item.stat.exists else '존재하지
않음' }}
        - 표준 Shell 여부: {{ '표준' if item.item.shell in standard_shells else
'비표준' }}
    loop: "{{ shell_check_results.results }}"
    when:
      - not item.stat.exists or item.item.shell not in standard_shells
```

```
---

- name: Check Vulnerability U-53
  hosts: all
  become: yes

  vars:
    standard_shells:
      - /bin/bash
      - /bin/sh
      - /bin/zsh
      - /bin/ksh
      - /bin/dash
      - /sbin/nologin
      - /bin/false

  tasks:
    - name:
      ansible.builtin.command: getent passwd
      register: passwd_output
      changed_when: false

    - name:
      ansible.builtin.set_fact:
        users_to_check: |
          {% set users = [] %}
          {% for line in passwd_output.stdout_lines %}
            {% set parts = line.split(':') %}
            {% if parts | length >= 7 and parts[2] | int >= 1000 %}
              {% set username = parts[0] %}
              {% set shell = parts[6] %}
              {% if shell is defined and shell != '' %}
                {% set _ = users.append({'username': username, 'shell': shell}) %}
```

1.계정 관리

U-54 (하)	1. 계정관리 > 1.15 Session Timeout 설정
취약점 개요	
점검내용	■ 사용자 쉘에 대한 환경설정 파일에서 session timeout 설정 여부 점검
점검목적	■ 사용자의 고의 또는 실수로 시스템에 계정이 접속된 상태로 방치됨을 차단하기 위함
보안위협	■ Session timeout 값이 설정되지 않은 경우 유휴 시간 내 비인가자의 시스템 접근으로 인해 불필요한 내부 정보의 노출 위험이 존재함
참고	※ **session:** 프로세스들 사이에 통신을 수행하기 위해서 메시지 교환을 통해 서로를 인식한 이후부터 통신을 마칠 때까지의 시간

```
      backup: yes
    notify:
      - sshd service restart

  - name:
    ansible.builtin.blockinfile:
      path: /etc/profile.d/session_timeout.sh
      block: |
        TMOUT={{ session_timeout_seconds }}
        readonly TMOUT
        export TMOUT
      create: true
      mode: '0644'
      backup: yes

handlers:
  - name:
    ansible.builtin.service:
      name: sshd
      state: restarted
```

```
---

- name: Check Vulnerability U-54
  hosts: all
  become: yes

  vars:
    session_timeout_seconds: 300

  tasks:
    - name:
      ansible.builtin.lineinfile:
        path: /etc/ssh/sshd_config
        regexp: '^#?ClientAliveInterval\s+\d+'
        line: 'ClientAliveInterval {{ session_timeout_seconds }}'
        state: present
        backup: yes
      notify:
        - sshd restart service

    - name:
      ansible.builtin.lineinfile:
        path: /etc/ssh/sshd_config
        regexp: '^#?ClientAliveCountMax\s+\d+'
        line: 'ClientAliveCountMax 0'
        state: present
```

1.계정 관리

U-54 (하)	1. 계정관리 > 1.15 Session Timeout 설정
취약점 개요	
점검내용	■ 사용자 쉘에 대한 환경설정 파일에서 session timeout 설정 여부 점검
점검목적	■ 사용자의 고의 또는 실수로 시스템에 계정이 접속된 상태로 방치됨을 차단하기 위함
보안위협	■ Session timeout 값이 설정되지 않은 경우 유휴 시간 내 비인가자의 시스템 접근으로 인해 불필요한 내부 정보의 노출 위험이 존재함
참고	※ **session:** 프로세스들 사이에 통신을 수행하기 위해서 메시지 교환을 통해 서로를 인식한 이후부터 통신을 마칠 때까지의 시간

```
      backup: yes
    notify:
      - sshd service restart

  - name:
    ansible.builtin.blockinfile:
      path: /etc/profile.d/session_timeout.sh
      block: |
        TMOUT={{ session_timeout_seconds }}
        readonly TMOUT
        export TMOUT
      create: true
      mode: '0644'
      backup: yes

handlers:
  - name:
    ansible.builtin.service:
      name: sshd
      state: restarted
```

```
---

- name: Check Vulnerability U-54
  hosts: all
  become: yes

  vars:
    session_timeout_seconds: 300

  tasks:
    - name:
      ansible.builtin.lineinfile:
        path: /etc/ssh/sshd_config
        regexp: '^#?ClientAliveInterval\s+\d+'
        line: 'ClientAliveInterval {{ session_timeout_seconds }}'
        state: present
        backup: yes
      notify:
        - sshd restart service

    - name:
      ansible.builtin.lineinfile:
        path: /etc/ssh/sshd_config
        regexp: '^#?ClientAliveCountMax\s+\d+'
        line: 'ClientAliveCountMax 0'
        state: present
```

2.파일 및 디렉터리 관리

U-05 (상)	2. 파일 및 디렉토리 관리 > 2.1 root홈, 패스 디렉터리 권한 및 패스 설정
취약점 개요	
점검내용	■ root 계정의 PATH 환경변수에 "."이(마침표) 포함되어 있는지 점검
점검목적	■ 비인가자가 불법적으로 생성한 디렉터리 및 명령어를 우선으로 실행되지 않도록 설정하기 위해 환경변수 점검이 필요함
보안위협	■ root 계정의 PATH(환경변수)에 정상적인 관리자 명령어(예: ls, mv, cp등)의 디렉터리 경로 보다 현재 디렉터리를 지칭하는 "." 표시가 우선하면 현재 디렉터리에 변조된 명령어를 삽입하여 관리자 명령어 입력 시 악의적인 기능이 실행 될 수 있음
참고	※ **환경변수:** 프로세스가 컴퓨터에서 동작하는 방식에 영향을 미치는 동적인 값들의 집합으로 Path 환경변수는 실행파일을 찾는 경로에 대한 변수임

```yaml
---

- name: Check Vulnerability U-05
  hosts: all
  tasks:
    - name: Read File - /root/.bash_profile
      ansible.builtin.slurp:
        src: /root/.bash_profile
      register: profile

    - name: Decode conetnt & Regex
      ansible.builtin.set_fact:
        PATH: >-
          {{
            ((profile.content | b64decode) | regex_search('PATH[=]([\w\/\$:]+)*[.][:]?([\w\/\$:]+)*', '\0'))
          }}

    - name: Replace File - /root/.bash_profile
      ansible.builtin.replace:
        path: /root/.bash_profile
        regexp: 'PATH[=]([\w\/\$:]+)*[.][:]?([\w\/\$:]+)*'
        replace: 'PATH=\1\2'
      when: PATH | length > 0
```

2.파일 및 디렉터리 관리

U-06 (상)	2. 파일 및 디렉토리 관리 > 2.2 파일 및 디렉터리 소유자 설정
취약점 개요	
점검내용	▮ 소유자 불분명한 파일이나 디렉터리가 존재하는지 여부를 점검
점검목적	▮ 소유자가 존재하지 않는 파일 및 디렉터리를 삭제 및 관리하여 임의의 사용자가 해당파일을 열람, 수정하는 행위를 사전에 차단하기 위함
보안위협	▮ 소유자가 존재하지 않는 파일의 UID와 동일한 값으로 특정계정의 UID값을 변경하면 해당 파일의 소유자가 되어 모든 작업이 가능함
참고	※ 소유자가 존재하지 않는 파일 및 디렉터리는 퇴직자의 자료이거나 관리 소홀로 인해 생긴 파일인 경우 또는 해킹으로 인한 공격자가 만들어 놓은 악의적인 파일인 경우가 있음

```
---

- hosts: linux
  tasks:
     - name: Check Vulnerability U-06
       copy:
         src: /path/to/local/config_file.conf
         dest: /path/to/remote/config_file.conf
         owner: myuser
         group: mygroup
         mode: '0644'
```

2.파일 및 디렉터리 관리

U-07 (상)	2. 파일 및 디렉토리 관리 > 2.3 /etc/passwd 파일 소유자 및 권한 설정
취약점 개요	
점검내용	■ /etc/passwd 파일 권한 적절성 점검
점검목적	■ /etc/passwd 파일의 임의적인 변경을 차단하기 위함을 통해 비인가자가 권한 상승하는 것을 막기 위함
보안위협	■ 관리자(root) 외 사용자가 "/etc/passwd" 파일의 사용자 정보를 변조하여 shell 변경, 사용자 추가/삭제 등 root를 포함한 사용자 권한 획득 가능
참고	※ **/etc/passwd:** 사용자의 ID, 패스워드, UID, GID, 홈 디렉터리, 쉘 정보를 담고 있는 파일

```
---

- name: Check File/Directory Stat - /etc/passwd
  hosts: all
  tasks:
    - name: Gather File/Directory Stat
      ansible.builtin.stat:
        path: /etc/passwd
      register: file_stat

- name: Report - Check Vulnerability
  hosts: master.hdk.ko.mega
  tasks:
    - name: Delimiter
      ansible.builtin.blockinfile:
        path: /home/hdk/ansible/report_{{ ansible_date_time.date }}.txt
        block: "---------- Vulnerability U-07 ----------"
        marker: ""
        create: yes

    - name: Write Report
      ansible.builtin.blockinfile:
        path: /home/hdk/ansible/report_{{ ansible_date_time.date }}.txt
        block: |
          hostname: {{ hostvars[item].ansible_facts.fqdn }}
          U-07 취 약 : {{ hostvars[item].file_stat.stat.pw_name != "root"
```

```
                or hostvars[item].file_stat.stat.mode != "0644" }}
      - 소 유 주 : {{ hostvars[item].file_stat.stat.pw_name }}
      - 권    한 : {{ hostvars[item].file_stat.stat.mode }}
  marker: ""
  create: yes
when: 'hostvars[item].file_stat.stat.pw_name != "root"
       or hostvars[item].file_stat.stat.mode != "0644"'
loop: "{{ hostvars.keys() }}"
```

2.파일 및 디렉터리 관리

U-08 (상)	2. 파일 및 디렉토리 관리 > 2.4 /etc/shadow 파일 소유자 및 권한 설정
취약점 개요	
점검내용	■ /etc/shadow 파일 권한 적절성 점검
점검목적	■ /etc/shadow 파일을 관리자만 제어할 수 있게 하여 비인가자들의 접근을 차단하도록 shadow 파일 소유자 및 권한을 관리해야함
보안위협	■ shadow파일은 패스워드를 암호화하여 저장하는 파일이며 해당 파일의 암호화된 해쉬값을 복호화하여(크래킹) 비밀번호를 탈취할 수 있음
참고	※ **/etc/shadow**: 시스템에 등록된 모든 계정의 패스워드를 암호화된 형태로 저장 및 관리하고 있는 파일

```
    create: yes

- name: Write Report
  ansible.builtin.blockinfile:
    path: /home/hdk/ansible/report_{{ ansible_date_time.date }}.txt
    block: |
      hostname: {{ hostvars[item].ansible_facts.fqdn }}
      U-08 취약 : {{ hostvars[item].file_stat.stat.pw_name != refer_file.pw_name
                  or hostvars[item].file_stat.stat.mode != refer_file.mode }}
        - 소유주 : {{ hostvars[item].file_stat.stat.pw_name }}
        - 권  한 : {{ hostvars[item].file_stat.stat.mode }}
    marker: ""
    create: yes
  when: 'hostvars[item].file_stat.stat.pw_name != refer_file.pw_name
        or hostvars[item].file_stat.stat.mode != refer_file.mode'
  loop: "{{ hostvars.keys() }}"
```

```
---

- name: Check File/Directory Stat - {{ refer_file.path }}
  hosts: all
  vars_files:
    - files_stat.yaml
  vars:
    refer_file: "{{ shadow }}"
  tasks:
    - name: Gather File/Directory Stat
      ansible.builtin.stat:
        path: "{{ refer_file.path }}"
      register: file_stat

- name: Report - Check Vulnerability
  hosts: master.hdk.ko.mega
  vars_files:
    - files_stat.yaml
  vars:
    refer_file: "{{ shadow }}"
  tasks:
    - name: Delimiter
      ansible.builtin.blockinfile:
        path: /home/hdk/ansible/report_{{ ansible_date_time.date }}.txt
        block: "---------- Vulnerability U-08 ----------"
        marker: ""
```

2.파일 및 디렉터리 관리

U-09 (상)	2. 파일 및 디렉토리 관리 > 2.5 /etc/hosts 파일 소유자 및 권한 설정
취약점 개요	
점검내용	■ /etc/hosts 파일의 권한 적절성 점검
점검목적	■ /etc/hosts 파일을 관리자만 제어할 수 있게 하여 비인가자들의 임의적인 파일 변조를 방지하기 위함
보안위협	■ hosts 파일에 비인가자 쓰기 권한이 부여된 경우, 공격자는 hosts파일에 악의적인 시스템을 등록하여, 이를 통해 정상적인 DNS를 우회하여 악성사이트로의 접속을 유도하는 파밍(Pharming) 공격 등에 악용될 수 있음 ■ hosts파일에 소유자외 쓰기 권한이 부여된 경우, 일반사용자 권한으로 hosts 파일에 변조된 IP주소를 등록하여 정상적인 DNS를 방해하고 악성사이트로의 접속을 유도하는 파밍(Pharming) 공격 등에 악용될 수 있음
참고	※ **/etc/hosts:** IP 주소와 호스트네임을 매핑하는 파일. 일반적으로 인터넷 통신 시 주소를 찾기 위해 도메인 네임 서비스(DNS)보다 hosts 파일을 먼저 참조함. hosts 파일은 문자열 주소로부터 IP 주소를 수신받는 DNS 서버와는 달리, 파일 내에 직접 문자열 주소와 IP 주소를 매칭하여 기록하며, DNS 서버 접근 이전에 확인하여 해당 문자열 주소가 목록에 존재할 시 그 문자열 주소에 해당하는 IP 주소로 연결함 ※ **파밍(Pharming):** 사용자의 DNS 또는 hosts 파일을 변조함으로써 정상적인 사이트로 오인하여 접속하도록 유도한 뒤 개인정보를 훔치는 새로운 컴퓨터 범죄 수법

```
    create: yes

- name: Write Report
  ansible.builtin.blockinfile:
    path: /home/hdk/ansible/report_{{ ansible_date_time.date }}.txt
    block: |
      호스트명 : {{ hostvars[item].ansible_facts.fqdn }}
      U-09 취약 : {{ hostvars[item].file_stat.stat.pw_name != refer_file.pw_name
                  or hostvars[item].file_stat.stat.mode != refer_file.mode }}
        - 소유주 : {{ hostvars[item].file_stat.stat.pw_name }}
        - 권  한 : {{ hostvars[item].file_stat.stat.mode }}
    marker: ""
    create: yes
  when: 'hostvars[item].file_stat.stat.pw_name != refer_file.pw_name
        or hostvars[item].file_stat.stat.mode != refer_file.mode'
  loop: "{{ hostvars.keys() }}"
```

```
---

- name: Check File/Directory Stat - {{ refer_file.path }}
  hosts: all
  vars_files:
    - files_stat.yaml
  vars:
    refer_file: "{{ hosts }}"
  tasks:
    - name: Gather File/Directory Stat
      ansible.builtin.stat:
        path: "{{ refer_file.path }}"
      register: file_stat

- name: Report - Check Vulnerability
  hosts: master.hdk.ko.mega
  vars_files:
    - files_stat.yaml
  vars:
    refer_file: "{{ hosts }}"
  tasks:
    - name: Delimiter
      ansible.builtin.blockinfile:
        path: /home/hdk/ansible/report_{{ ansible_date_time.date }}.txt
        block: "---------- Vulnerability U-09 ----------"
        marker: ""
```

2.파일 및 디렉터리 관리

U-10 (상)	2. 파일 및 디렉토리 관리 > 2.6 /etc/(x)inetd.conf 파일 소유자 및 권한 설정
취약점 개요	
점검내용	■ /etc/(x)inetd.conf 파일 권한 적절성 점검
점검목적	■ /etc/(x)inetd.conf 파일을 관리자만 제어할 수 있게 하여 비인가자들의 임의적인 파일 변조를 방지하기 위함
보안위협	■ (x)inetd.conf 파일에 소유자외 쓰기 권한이 부여된 경우, 일반사용자 권한으로 (x)inetd.conf 파일에 등록된 서비스를 변조하거나 악의적인 프로그램(서비스)를 등록할 수 있음
참고	※ (x)inetd (슈퍼데몬) : 자주 사용하지 않는 서비스가 상시 실행되어 메모리를 점유하는 것을 방지하기 위해 (x)inetd(슈퍼대몬)에 자주 사용하지 않는 서비스를 등록하여 요청이 있을시에만 해당 서비스를 실행하고 요청이 끝나면 서비스를 종료하는 역할 수행

```
---

- name: Check Vulnerability U-10
  hosts: all
  become: true
  tasks:
    - name: Ensure /etc/(x)inetd.conf exists (or create if not)
      file:
        path: /etc/xinetd.conf
        state: touch
        mode: '0600'
        owner: root
        group: root

    - name: Ensure /etc/(x)inetd.conf has correct ownership and permissions
      file:
        path: /etc/xinetd.conf
        owner: root
        group: root
        mode: '0600'
      notify:
        - restart xinetd

  handlers:
    - name: restart xinetd
      service:
        name: xinetd
        state: restarted
```

2.파일 및 디렉터리 관리

U-11 (상)	2. 파일 및 디렉토리 관리 > 2.7 /etc/syslog.conf 파일 소유자 및 권한 설정
취약점 개요	
점검내용	■ /etc/syslog.conf 파일 권한 적절성 점검
점검목적	■ /etc/syslog.conf 파일의 권한 적절성을 점검하여, 관리자 외 비인가자의 임의적인 syslog.conf 파일 변조를 방지하기 위함
보안위협	■ syslog.conf 파일의 설정내용을 참조하여 로그의 저장위치가 노출되며 로그을 기록하지 않도록 설정하거나 대량의 로그를 기록하게 하여 시스템 과부하를 유도할 수 있음
참고	※ **/etc/syslog.conf** : syslogd 데몬 실행시 참조되는 설정파일로 시스템 로그 기록의 종류, 위치 및 Level을 설정할 수 있음

```
---

- name: Check Vulnerability U-11
  hosts: all
  become: true

  tasks:
    - name: Change owner of /etc/syslog.conf to root
      file:
        path: /etc/syslog.conf
        owner: root
        group: root
        mode: '0640'
```

2.파일 및 디렉터리 관리

U-12 (상)	2. 파일 및 디렉토리 관리 > 2.8 /etc/services 파일 소유자 및 권한 설정
취약점 개요	
점검내용	■ /etc/services 파일 권한 적절성 점검
점검목적	■ /etc/services 파일을 관리자만 제어할 수 있게 하여 비인가자들의 임의적인 파일 변조를 방지하기 위함
보안위협	■ services 파일의 접근권한이 적절하지 않을 경우 비인가 사용자가 운영 포트 번호를 변경하여 정상적인 서비스를 제한하거나, 허용되지 않은 포트를 오픈하여 악성 서비스를 의도적으로 실행할 수 있음
참고	※ **/etc/services** : 서비스 관리를 위해 사용되는 파일. 해당 파일에 서버에서 사용하는 모든 포트(port)들에 대해 정의되어 있으며, 필요시 서비스 기본사용 포트를 변경하여 네트워크 서비스를 운용할 수 있음

```
---

- name: Check Vulnerability U-12
  hosts: all
  become: true
  tasks:
    - name: Set /etc/services ownership and permissions
      ansible.builtin.file:
        path: /etc/services
        owner: root
        group: root
        mode: "0644"
```

2.파일 및 디렉터리 관리

U-13 (상)	2. 파일 및 디렉토리 관리 > 2.9 SUID, SGID, 설정 파일점검
취약점 개요	
점검내용	■ 불필요하거나 악의적인 파일에 SUID, SGID 설정 여부 점검
점검목적	■ 불필요한 SUID, SGID 설정 제거로 악의적인 사용자의 권한상승을 방지하기 위함
보안위협	■ SUID, SGID 파일의 접근권한이 적절하지 않을 경우 SUID, SGID 설정된 파일로 특정 명령어를 실행하여 root 권한 획득 가능함
참고	※ **SUID:** 설정된 파일 실행 시, 특정 작업 수행을 위하여 일시적으로 파일 소유자의 권한을 얻게 됨 ※ **SGID:** 설정된 파일 실행 시, 특정 작업 수행을 위하여 일시적으로 파일 소유 그룹의 권한을 얻게 됨 ※ 불필요한 SUID/SGID 목록: 부록 참고

```
---

- name: Check Vulnerability U-13
  hosts: all
  vars_files:
    - reference_files.yaml
  tasks:
    - name: Search SetUID Files
      ansible.builtin.shell:
        cmd: "find / -xdev -perm -4000 -type f -print"
      register: setuid_files

    - name: Search SetGID Files
      ansible.builtin.shell:
        cmd: "find / -xdev -perm -2000 -type f -print"
      register: setgid_files

    - name: Delete Files - Extra SetUID
      ansible.builtin.file:
        name: "{{ item }}"
        state: absent
      loop: "{{ setuid_files.stdout_lines | difference(setuid) }}"

    - name: Delete Files - Extra SetGID
      ansible.builtin.file:
        name: "{{ item }}"
        state: absent
      loop: "{{ setgid_files.stdout_lines | difference(setgid) }}"
```

2.파일 및 디렉터리 관리

U-14 (상)	2. 파일 및 디렉토리 관리 > 2.10 사용자, 시스템 시작파일 및 환경파일 소유자 및 권한 설정
취약점 개요	
점검내용	■ 홈 디렉터리 내의 환경변수 파일에 대한 소유자 및 접근권한이 관리자 또는 해당 계정으로 설정되어 있는지 점검
점검목적	■ 비인가자의 환경변수 조작으로 인한 보안 위험을 방지하기 위함
보안위협	■ 홈 디렉터리 내의 사용자 파일 및 사용자별 시스템 시작파일 등과 같은 환경변수 파일의 접근권한 설정이 적절하지 않을 경우 비인가자가 환경변수 파일을 변조하여 정상 사용중인 사용자의 서비스가 제한 될 수 있음
참고	※ **환경변수 파일 종류:** ".profile", ".kshrc", ".cshrc", ".bashrc", ".bash_profile", ".login", ".exrc", ".netrc" 등

```
---

- name: Remove Vulnerability U-14
  file:
    path: /path/to/file
    owner: user1
    group: group1
    mode: '0750'
```

2.파일 및 디렉터리 관리

U-15 (상)	2. 파일 및 디렉토리 관리 > 2.11 world writable 파일 점검
취약점 개요	
점검내용	■ 불필요한 world writable 파일 존재 여부 점검
점검목적	■ world writable 파일을 이용한 시스템 접근 및 악의적인 코드 실행을 방지하기 위함
보안위협	■ 시스템 파일과 같은 중요 파일에 world writable 설정이 될 경우, 일반사용자 및 비인가된 사용자가 해당 파일을 임의로 수정, 삭제가 가능함
참고	※ **world writable 파일:** 파일의 내용을 소유자나 그룹 외 모든 사용자에 대해 쓰기가 허용된 파일 (예 : **rwxrwxrwx** root root <파일명>)

```
---

- name: Check Vulnerability U-15
  stat:
    path: /path/to/file
  register: file_stat

- name: Print file permissions
  debug:
    msg: "File: {{ file_stat.path }}, Permissions: {{ file_stat.stat.mode }}"
```

2.파일 및 디렉터리 관리

U-16 (상)	2. 파일 및 디렉토리 관리 > 2.12 /dev에 존재하지 않는 device 파일 점검
취약점 개요	
점검내용	■ 존재하지 않는 device 파일 존재 여부 점검
점검목적	■ 실제 존재하지 않는 디바이스를 찾아 제거함으로써 root 파일 시스템 손상 및 다운 등의 문제를 방지하기 위함
보안위협	■ 공격자는 rootkit 설정파일들을 서버 관리자가 쉽게 발견하지 못하도록 /dev에 device 파일인 것처럼 위장하는 수법을 많이 사용함
참고	※ **/dev 디렉터리:** 논리적 장치 파일을 담고 있는 /dev 디렉터리는 /devices 디렉터리에 있는 물리적 장치 파일에 대한 심볼릭 링크임. 예를 들어 rmt0를 rmto로 잘못 입력한 경우 rmto 파일이 새로 생성되는 것과 같이 디바이스 이름 입력 오류 시 root 파일 시스템이 에러를 일으킬 때까지 /dev 디렉터리에 계속해서 파일을 생성함 ※ /dev 디렉터리 내 불필요한 device 파일이 존재할 시 삭제 권고

```
---

- name: Check Vulnerability U-16
  hosts: all
  gather_facts: false
  tasks:
    - name: Get list of files in /dev
      find:
        paths: /dev
        file_type: file
      register: dev_files

    - name: Define expected device files
      set_fact:
        expected_devices:
          - "sda"
          - "sdb"
          - "sda1"
          - "sdb1

    - name: Find missing device files
      set_fact:
        missing_devices: "{{ expected_devices | difference(dev_files.files | map(attribute='path') | map('basename'))
}}"
```

```
    - name: Print missing device files
      debug:
        msg: "Missing device files: {{ missing_devices }}"
```

2.파일 및 디렉터리 관리

U-17 (상)	2. 파일 및 디렉토리 관리 > 2.13 $HOME/.rhosts, hosts.equiv 사용 금지
취약점 개요	
점검내용	■ /etc/hosts.equiv 파일 및 .rhosts 파일 사용자를 root 또는, 해당 계정으로 설정한 뒤 권한을 600으로 설정하고 해당파일 설정에 '+' 설정(모든 호스트 허용)이 포함되지 않도록 설정되어 있는지 점검
점검목적	■ 'r' command 사용을 통한 원격 접속은 인증 없이 관리자 원격접속이 가능하므로 서비스 포트를 차단해야 함
보안위협	■ rlogin, rsh 등과 같은 'r' command의 보안 설정이 적용되지 않은 경우, 원격지의 공격자가 관리자 권한으로 목표 시스템상의 임의의 명령을 수행시킬 수 있으며, 명령어 원격 실행을 통해 중요 정보 유출 및 시스템 장애를 유발시킬 수 있음. 또한 공격자 백도어 등으로도 활용될 수 있음 ■ r-command(rlogin, rsh등) 서비스의 접근통제에 관련된 파일로 권한설정을 미 적용한 경우 r-command 서비스 사용 권한을 임의로 등록하여 무단 사용이 가능함
참고	※ **'r'command:** 인증 없이 관리자의 원격접속을 가능하게 하는 명령어들로 rsh(remsh), rlogin, rexec 등이 있으며, 포트번호 512,513,514 (TCP)를 사용함

```
---

- name: Check Vulnerability U-17
  file:
    path: "{{ ansible_user_dir }}/.rhosts"
    owner: "{{ ansible_user_id }}"
    group: "{{ ansible_user_id }}"
    mode: "600"
  when: rhosts_file_stat.stat.exists

- name: Set permissions for hosts.equiv file
  file:
    path: /etc/hosts.equiv
    owner: root
    group: root
    mode: "600"
  when: hosts_equiv_file_stat.stat.exists
```

2.파일 및 디렉터리 관리

U-18 (상)	2. 파일 및 디렉토리 관리 > 2.14 접속 IP 및 포트 제한
취약점 개요	
점검내용	■ 허용할 호스트에 대한 접속 IP 주소 제한 및 포트 제한 설정 여부 점검
점검목적	■ 허용한 호스트만 서비스를 사용하게 하여 서비스 취약점을 이용한 외부자 공격을 방지하기 위함
보안위협	■ 허용할 호스트에 대한 IP 및 포트제한이 적용되지 않은 경우, Telnet, FTP같은 보안에 취약한 네트워크 서비스를 통하여 불법적인 접근 및 시스템 침해 사고가 발생할 수 있음
참고	• 접속 IP 및 포트제한 애플리케이션 종류 예시 ※ **TCP Wrapper:** 네트워크 서비스에 관련한 트래픽을 제어하고 모니터링 할 수 있는 UNIX 기반의 방화벽 툴 ※ **IPFilter:** 유닉스 계열에서 사용하는 공개형 방화벽 프로그램으로써 Packet Filter로 시스템 및 네트워크 보안에 아주 강력한 기능을 보유한 프로그램 ※ **IPtables:** 리눅스 커널 방화벽이 제공하는 테이블들과 그것을 저장하는 체인, 규칙들을 구성할 수 있게 해주는 응용프로그램

```
---

- name: Check Vulnerability U-18
  firewalld:
    service: ssh
    rich_rule:
    permanent: yes
    state: enabled
  become: true
- name: Remove default SSH rule
  firewalld:
    service: ssh
    permanent: yes
    state: disabled
  become: true
```

☺봐주셔서 감사합니다.☺