(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년06월26일
(11) 등록번호 10-2679203
(24) 등록일자 2024년06월24일

(51) 국제특허분류(Int. Cl.)
G06F 21/57 (2013.01) *G06F 21/55* (2013.01)
(52) CPC특허분류
G06F 21/577 (2013.01)
G06F 21/552 (2013.01)
(21) 출원번호 10-2022-0021182
(22) 출원일자 2022년02월18일
심사청구일자 2022년02월18일
(65) 공개번호 10-2023-0124189
(43) 공개일자 2023년08월25일
(56) 선행기술조사문헌
KR100918370 B1*
KR102176324 B1*
KR102186127 B1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
중부대학교 산학협력단
충청남도 금산군 추부면 대학로 201
(72) 발명자
조서연
서울특별시 서대문구 독립문공원길 17
김미란
경기도 파주시 운정로 11 (상지석동, 서부주택)
(뒷면에 계속)
(74) 대리인
특허법인(유)남아이피그룹, 특허법인 남앤남

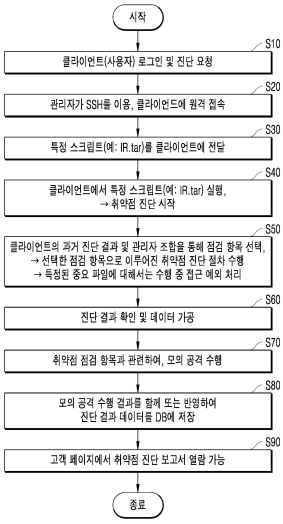
전체 청구항 수 : 총 4 항 심사관 : 구대성

(54) 발명의 명칭 **진단서버에 의해 제공되는 취약점 진단 방법**

(57) 요 약

본 발명은, 클라이언트의 과거 진단 결과 및 원격지의 관리자에 따라 선택되는 점검 항목으로 이루어진 동적 취약점 진단 절차를 수행하는 방식, 진단 결과 기반의 모의 공격을 수행하는 방식, 취약점 진단 절차 중에 중요 시스템파일에 접근되는 일 이 없도록 예외 처리하는 방식 실현을 통해, 새로운 방식의 취약점 진단 기술을 구현/수행함으로써, 취약점 진단 절차를 수행함에 있어서 진단 결과의 신뢰도를 높이고 진단 과정에서의 불필요한 부하 및 비용을 절감할 수 있는 방안을 제안하고 있다.

대 표 도 - 도2



(72) 발명자

김예리

경기도 파주시 광탄면 보광로553번길 92-1

김성한

서울특별시 은평구 갈현로47길 9 (갈현동, MS아트빌)

이윤지

경기도 고양시 일산서구 경의로 867 (덕이동, 철산아파트)

양환석

경기도 고양시 일산서구 후곡로 35 (일산동, 후곡마을5단지아파트)

명 세 서

청구범위

청구항 1

진단서버에 의해 제공되는 취약점 진단 방법에 있어서,

진단을 요청하는 클라이언트로, 원격지의 관리자가 웹기반 터미널 에뮬레이터 기반의 접속 서비스(SSH)를 이용하여 접속되도록 하는 접속 단계;

상기 관리자에 의한 특정 스크립트를 상기 클라이언트로 전달하여, 상기 클라이언트에서 상기 특정 스크립트를 실행하도록 하는 실행 단계;

상기 특정 스크립트 실행을 통해, 상기 클라이언트에서 취약점 진단 절차가 수행되게 하며 기 특정된 중요 파일에 대해서는 상기 취약점 진단 절차 수행 중에 접근되는 일이 없도록 예외 처리하는 절차 수행 단계;

상기 취약점 진단 절차에 따라 생성되는 진단 결과 데이터가 웹에서 표시 가능한 형태로 가공된 후 데이터 베이스(DB)에 저장되도록 하는 저장 단계를 포함하며;

상기 특정된 중요 파일은,

루트 파일시스템(root filesystem) 생성 시점에 생성된 것으로 식별되는 파일로서,

I-노드로 식별되는 위치에 블록 단위로 파일을 저장하는 데이터 구조로부터, 블록 비트맵 및 I-노드 비트맵에서 파일 데이터가 가장 먼저 저장되는 가장 작은 블록 넘버 및 I-노드 넘버 중 적어도 하나의 생성시간을 확인하고,

블록 비트맵 및 I-노드 비트맵으로부터, 상기 확인한 생성시간과 같은 생성시간을 가지며 기 설정된 임계 블록 넘버 및 I-노드 넘버 중 적어도 하나 보다 작은 넘버를 가지는 것으로 확인되는 파일인 것을 특징으로 하는 취약점 진단 방법.

청구항 2

제 1 항에 있어서,

상기 취약점 진단 절차에는 기 설정된 다수의 점검 항목이 포함되며,

상기 절차 수행 단계는,

상기 다수의 점검 항목 중에서, 상기 DB에서 조회되는 상기 클라이언트의 과거 진단 결과 데이터 및 상기 클라이언트의 진단 요청을 처리하는 원격지의 관리자 중 적어도 하나에 따라 선택되는 점검 항목으로 이루어지는 취약점 진단 절차가 수행되도록 하는 것을 특징으로 하는 취약점 진단 방법.

청구항 3

제 1 항에 있어서,

상기 취약점 진단 절차에 따른 진단 결과를 근거로, 상기 클라이언트에서 취약점으로 확인되는 특정 점검 항목과 관련하여 기 설정된 모의 공격을 수행하는 단계를 더 포함하며;

상기 저장 단계는,

상기 DB에 저장되는 진단 결과 데이터와 상기 모의 공격 수행 결과가 함께 저장되도록 하거나, 상기 DB에 저장되는 진단 결과 데이터에 상기 모의 공격 수행 결과를 반영 및 처리한 후 저장되도록 하는 것을 특징으로 하는 취약점 진단 방법.

청구항 4

제 1 항에 있어서,

상기 클라이언트가 로그인을 통해 상기 진단서버에 접속하면, 상기 DB 내 저장 데이터를 기반으로 상기 클라이언트의 고객 페이지에서 적어도 하나 이상의 진단 결과 데이터에 따른 취약점 진단 보고서를 열람할 수 있게 하는 열람 단계를 더 포함하는 것을 특징으로 하는 취약점 진단 방법.

청구항 5

삭제

발명의 설명

기 술 분 야

[0001] 본 발명은 악성 공격으로부터 보안이 취약할 수 있는 보안 취약점을 점검/진단하는 기술에 관한 것이다.

배 경 기 술

[0002] 악성 행위를 수행하여 공격을 감행하는 악성코드의 종류는 더욱 다양해지고 있으며, 인터넷 사용인구의 증가와 함께 이러한 악성 공격에 의한 보안 사고(해킹 및 침해 사고) 역시 빈번히 일어나고 있다.

[0003] 이에, 안전한 컴퓨팅 환경을 위해, 다양한 방식의 악성코드 진단 방안 및 보안 취약점 점검 방안들이 등장하였다.

[0004] 보안 취약점 점검이란, 정보 시스템에 불법적인 사용자의 접근을 허용할 수 있는 위협이나 점검 대상 시스템에서 관리하는 중요 데이터의 유출, 변조, 삭제에 대한 위협이 발생할 수 있는 사항들에 대하여 점검하는 것으로, 점검 대상 시스템에 보안 취약점이 존재하고 있는지에 대한 점검 작업을 수행한 후 점검대상시스템의 보안수준을 분석하는 것을 지칭한다.

[0005] 기존의 보안 취약점 점검 방안은, 점검 대상이 되는 클라이언트(예: PC, 네트워크 장비 등)를 선택하고 기 정해진 진단 기준에 따른 점검을 수행하여, 클라이언트에 보안 취약점이 존재하는지 여부를 점검하는 방식이다.

[0006] 이와 같은 기존의 보안 취약점 점검 방안은, 진단 기준에 따른 점검을 수행하여 보안 취약점 존재 여부의 점검 결과를 얻을 뿐, 그 점검 결과의 신뢰도에 대한 검증이 이루어지지 않고 있다.

[0007] 이에, 기존 방식으로는, 보안 취약점 점검이 원격지의 관리자에 의해 진행될 경우 관리자에 따라 그 점검 결과의 정확도 및 정당성이 달라질 수 있는 한계, 또는 단순히 진단 기준에 따른 점검 결과에 대해 보다 강력한 정확도가 필요한 상황이나 이를 만족시키기 어려운 한계가 있다.

[0008] 이 외에도, 기존 방식으로는, 보안 취약점 점검 과정 중에 접근해선 안 되는 중요 시스템파일에 접근하게 되는 경우가 발생할 경우 이를 예외 처리하지 못하여, 점검 결과의 신뢰도가 낮아질 뿐 아니라 클라이언트 내 시스템에 치명적인 상황을 야기시키는 문제까지도 발생할 수 있다.

[0009] 이 밖에도, 기존 방식으로는, 동일 클라이언트에 대해 매번 동일하게 보안 취약점 점검을 수행하므로, 보안 취약점 점검 과정에서의 불필요한 부하 및 비용이 소요되는 한계도 있을 수 있다.

[0010] 이에, 본 발명에서는, 클라이언트(예: PC, 네트워크 장비 등)에 대한 보안 취약점 점검을 수행함에 있어, 점검 결과의 신뢰도를 높이고 더 나아가 점검 과정에서의 불필요한 부하 및 비용을 절감할 수 있는 방식의 취약점 점검(진단) 기술을 제안하고자 한다.

발명의 내용

해결하려는 과제

[0011] 본 발명은 상기한 사정을 감안하여 창출된 것으로서, 본 발명에서 해결하고자 하는 과제는, 클라이언트(예: PC, 네트워크 장비 등)에 대한 보안 취약점 점검을 수행함에 있어, 점검 결과의 신뢰도를 높이고 더 나아가 점검 과정에서의 불필요한 부하 및 비용을 절감할 수 있는 새로운 방식의 취약점 점검(진단) 기술을 제공하는데 있다.

과제의 해결 수단

[0012] 상기 목적을 달성하기 위한 본 발명의 일 관점에 따른 진단서버에 의해 제공되는 취약점 진단 방법은, 진단을 요청하는 클라이언트로, 원격지의 관리자가 웹기반 터미널 에뮬레이터 기반의 접속 서비스(SSH)를 이용하여 접

속되도록 하는 접속 단계; 상기 관리자에 의한 특정 스크립트를 상기 클라이언트로 전달하여, 상기 클라이언트에서 상기 특정 스크립트를 실행하도록 하는 실행 단계; 상기 특정 스크립트 실행을 통해, 상기 클라이언트에서 취약점 진단 절차가 수행되게 하는 절차 수행 단계; 상기 취약점 진단 절차에 따라 생성되는 진단 결과 데이터가 웹에서 표시 가능한 형태로 가공된 후 데이터 베이스(DB)에 저장되도록 하는 저장 단계; 및 상기 클라이언트가 로그인을 통해 상기 진단서버에 접속하면, 상기 DB 내 저장 데이터를 기반으로 상기 클라이언트의 고객 페이지에서 적어도 하나 이상의 진단 결과 데이터에 따른 취약점 진단 보고서를 열람할 수 있게 하는 열람 단계를 포함한다.

[0013] 구체적으로, 상기 취약점 진단 절차에는 기 설정된 다수의 점검 항목이 포함되며, 상기 절차 수행 단계는, 상기 다수의 점검 항목 중에서, 상기 DB에서 조회되는 상기 클라이언트의 과거 진단 결과 데이터 및 상기 클라이언트의 진단 요청을 처리하는 원격지의 관리자 중 적어도 하나에 따라 선택되는 점검 항목으로 이루어지는 취약점 진단 절차가 수행되도록 할 수 있다.

[0014] 구체적으로, 상기 취약점 진단 절차에 따른 진단 결과를 근거로, 상기 클라이언트에서 취약점으로 확인되는 특정 점검 항목과 관련하여 기 설정된 모의 공격을 수행하는 단계를 더 포함하며; 상기 저장 단계는, 상기 DB에 저장되는 진단 결과 데이터와 상기 모의 공격 수행 결과가 함께 저장되도록 하거나, 상기 DB에 저장되는 진단 결과 데이터에 상기 모의 공격 수행 결과를 반영 및 처리한 후 저장되도록 할 수 있다.

[0015] 구체적으로, 상기 절차 수행 단계는, 기 특정된 중요 파일에 대해서는 상기 취약점 진단 절차 수행 중에 접근되는 일이 없도록 예외 처리하며, 상기 특정된 중요 파일은, 상기 클라이언트의 파일시스템의 데이터 구조로부터 확인되는 파일 생성시점 및 파일 위치 중 적어도 하나를 근거로, 루트 파일시스템(root filesystem) 생성 시점에 생성된 것으로 식별되는 파일일 수 있다.

[0016] 구체적으로, 상기 특정된 중요 파일은, 노드로 식별되는 위치에 블록 단위로 파일을 저장하는 데이터 구조로부터, 블록 및 I-노드 비트맵에서 파일 데이터가 가장 먼저 저장되는 가장 작은 블록 및 I-노드 넘버 중 적어도 하나의 생성시간을 확인하고, 블록 및 I-노드 비트맵으로부터, 상기 확인한 생성시간과 같은 생성시간을 가지며 기 설정된 임계 블록 및 I-노드 넘버 중 적어도 하나 보다 작은 넘버를 가지는 파일을 확인하여, 상기 루트 파일시스템 생성 시점에 생성된 파일로 식별될 수 있다.

발명의 효과

[0017] 본 발명의 실시 예들에 따르면, 클라이언트의 과거 진단 결과 및 원격지의 관리자에 의한 경우 관리자에 따라 선택되는 점검 항목으로 취약점 진단 절차를 수행하는 방식, 진단 결과 기반의 모의 공격을 수행하는 방식, 취약점 진단 절차 중에 중요 시스템파일에 접근되는 일 이 없도록 예외 처리하는 방식 실현을 통해, 새로운 방식의 취약점 진단 기술을 구현해 내고 있다.

[0018] 이로써, 본 발명에 따르면, 보안 취약점을 진단하는데 있어, 진단 결과의 신뢰도를 높이고 진단 과정에서의 불필요한 부하 및 비용을 절감할 수 있는 효과를 도출할 수 있다.

도면의 간단한 설명

[0019] 도 1은 본 발명의 일 실시 예에 따른 취약점 진단 시스템의 구조를 보여주는 도면이다.

도 2는 본 발명의 일 실시 예에 따른 취약점 진단 방법이 동작하는 흐름을 보여주는 흐름도이다.

도 3 및 도 4는 본 발명의 취약점 진단 방법에 의해 얻을 수 있는 진단 결과 데이터(가공 전/후)를 보여주는 예시도이다.

발명을 실시하기 위한 구체적인 내용

[0020] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시 예를 가질 수 있는 바, 특정 실시 예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다. 각 도면을 설명하면서 유사한 참조부호를 유사한 구성요소에 대해 사용하였다.

[0021] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있

다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다.

[0022] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0023] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가지고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 가지는 의미와 일치하는 의미를 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0024] 이하, 첨부된 도면을 참조하여 본 발명에 대하여 설명한다.

[0025] 본 발명은, 악성 공격으로부터 보안이 취약할 수 있는 보안 취약점을 점검/진단하는 기술에 관한 것이다.

[0026] 악성 행위를 수행하여 공격을 감행하는 악성코드의 종류는 더욱 다양해지고 있으며, 인터넷 사용인구의 증가와 함께 이러한 악성 공격에 의한 보안 사고(해킹 및 침해 사고) 역시 빈번히 일어나고 있다.

[0027] 이에, 안전한 컴퓨팅 환경을 위해, 다양한 방식의 악성코드 진단 방안 및 보안 취약점 점검 방안들이 등장하였다.

[0028] 보안 취약점 점검이란, 정보 시스템에 불법적인 사용자의 접근을 허용할 수 있는 위협이나 점검 대상 시스템에서 관리하는 중요 데이터의 유출, 변조, 삭제에 대한 위협이 발생할 수 있는 사항들에 대하여 점검하는 것으로, 점검 대상 시스템에 보안 취약점이 존재하고 있는지에 대한 점검 작업을 수행한 후 점검대상시스템의 보안수준을 분석하는 것을 지칭한다.

[0029] 기존의 보안 취약점 점검 방안은, 점검 대상이 되는 클라이언트(예: PC, 네트워크 장비 등)를 선택하고 기 정해진 진단 기준에 따른 점검을 수행하여, 클라이언트에 보안 취약점이 존재하는지 여부를 점검하는 방식이다.

[0030] 이와 같은 기존의 보안 취약점 점검 방안은, 진단 기준에 따른 점검을 수행하여 보안 취약점 존재 여부의 점검 결과를 얻을 뿐, 그 점검 결과의 신뢰도에 대한 검증이 이루어지지 않고 있다.

[0031] 이에, 기존 방식으로는, 보안 취약점 점검이 원격지의 관리자에 의해 진행될 경우 관리자에 따라 그 점검 결과의 정확도 및 정당성이 달라질 수 있는 한계, 또는 단순히 진단 기준에 따른 점검 결과에 대해 보다 강력한 정확도가 필요한 상황이나 이를 만족시키기 어려운 한계가 있다.

[0032] 이 외에도, 기존 방식으로는, 보안 취약점 점검 과정 중에 접근해선 안 되는 중요 시스템파일에 접근하게 되는 경우가 발생할 경우 이를 예외 처리하지 못하여, 점검 결과의 신뢰도가 낮아질 뿐 아니라 클라이언트 내 시스템에 치명적인 상황을 야기시키는 문제까지도 발생할 수 있다.

[0033] 이 밖에도, 기존 방식으로는, 동일 클라이언트에 대해 매번 동일하게 보안 취약점 점검을 수행하므로, 보안 취약점 점검 과정에서의 불필요한 부하 및 비용이 소요되는 한계도 있을 수 있다.

[0034] 이에, 본 발명에서는, 클라이언트(예: PC, 네트워크 장비 등)에 대한 보안 취약점 점검을 수행함에 있어, 점검 결과의 신뢰도를 높이고 더 나아가 점검 과정에서의 불필요한 부하 및 비용을 절감할 수 있는 방식의 취약점 점검(진단) 기술을 제안하고자 한다.

[0035] 더 정확히 설명하면, 본 발명에서는, 클라이언트(예: PC, 네트워크 장비 등)에 대한 보안 취약점 점검을 수행함에 있어, 점검 결과의 신뢰도를 높이고 점검 과정에서의 불필요한 부하 및 비용을 절감할 수 있는 구체화된 기술 구성을 제안/실현함으로써, 새로운 방식의 취약점 진단 기술을 구현하고자 한다.

[0036] 먼저, 도 1을 참조하여, 본 발명에서 제안하는 취약점 진단 시스템에 대하여 설명하겠다.

[0037] 도 1에 도시된 바와 같이, 본 발명에 따른 취약점 진단 시스템은, 점검(진단) 대상이 되는 클라이언트(예: PC, 네트워크 장비 등)에 해당될 수 있는 사용자, 원격지에서 취약점 진단 절차를 진행할 수 있는 관리자, 그리고 본 발명의 취약점 진단 방안을 관장하는 진단서버(100) 및 DB를 포함하여 구성될 수 있다.

[0038] 즉, 본 발명의 취약점 진단 시스템은, 도 1에 도시된 각 구성들 간의 유기적인 통신 및 동작을 기반으로, 클라

이언트(예: PC, 네트워크 장비 등)에 대한 취약점 진단을 수행함에 있어, 진단 결과의 신뢰도를 높이고 진단 과정에서의 불필요한 부하 및 비용을 절감할 수 있는 구체화된 기술 구성을 실현할 수 있다.

[0039] 이하에서는, 도 1을 참조하여, 본 발명의 취약점 진단 시스템(취약점 진단 방법)에 의한 취약점 진단 시나리오의 일 예를 조금 더 구체적으로 설명하겠다.

[0040] 본 발명에서 진단서버(100)는 취약점 진단 방안을 관장하는 서버로서, 일명 IR(Inform Relief) System이라 칭할 수 있다.

[0041] 본 발명에서는, 진단서버(100)에서 제공하는 웹 페이지에 사용자가 회원 가입 후 로그인을 한 후, 사용자는 자신의 PC(이하, 클라이언트라 함)에 대한 정보를 입력하고 취약점 진단을 요청할 수 있다(1).

[0042] 이렇게 되면, 본 발명에서는, 원격지의 관리자가 사용자의 정보(이하, 클라이언트 정보라 함)을 토대로, 기반 터미널 에뮬레이터 기반의 접속 서비스(Secure Shell, SSH)를 이용하여 해당 클라이언트에 접속하게 된다(2).

[0043] 이후, 본 발명에서 원격지의 관리자는 cmd에서 관리자 컴퓨터의 바탕화면에 있던 특정 스크립트(이하, IR.tar라 함)를, scp 명령어를 사용하여 사용자의 PC 즉 진단 대상의 클라이언트에 전달한다(2).

[0044] 본 발명에서 사용자의 PC 즉 진단 대상의 클라이언트에서는, 특정 스크립트 즉 IR.tar의 압출을 풀고 실행하면 클라이언트에서 취약점 진단 절차가 시작될 수 있다. 즉, 클라이언트에서 IR.tar 실행을 통해 취약점 진단 절차가 수행되는 것이다.

[0045] 그리고 본 발명에서는, 클라이언트에서 취약점 진단 절차가 끝나면, 취약점 진단 절차에 따른 진단 결과 데이터가 생성된다. 예를 들면, 진단 결과 데이터는, 날짜 + IP주소.txt 형태로서, 모든 결과가 들어있는 txt 파일로 생성될 수 있다.

[0046] 아울러, 웹에서 사용자(고객)가 볼 수 있는 형태로 데이터를 가공해야 하기 때문에, 본 발명에서 클라이언트에서는, 취약점 진단 절차를 통해 생성된 진단 결과 데이터(예: 날짜 + IP주소.txt 파일)를 가공시켜 줄 "./final_DB.sh"라는 스크립트를 실행시켜 진단 결과 데이터(예: 날짜 + IP주소.txt 파일)를 웹에서 표시 가능한 형태로 가공한 후, mysql 명령어를 이용하여 DB에 저장하며 진단 완료를 처리할 수 있다(3).

[0047] 이후, 본 발명에서는, 사용자(고객)가 로그인을 통해 진단서버(100)에 접속하면, 사용자(고객)의 마이 페이지에서 앞서 진행한 취약점 진단 절차의 진단 결과 데이터에 따른 취약점 진단 보고서를 열람할 수 있다(4).

[0048] 이렇듯, 본 발명에서는, 진단 대상의 클라이언트(예: PC, 네트워크 장비 등)에 취약점 진단 Agent를 설치하지 않고, 원격지의 관리자(또는 매니저)에 의한 명령어나 스크립트 전송을 통해 취약점 진단 절차가 진행(시작, 수행, 가공, 완료)되는 방식을 기본 구성으로 한다.

[0049] 즉, 본 발명에서는, 사용자(고객)가 웹 서비스(예: 마이 페이지)를 통해서, 관리자에 의해 진행된 취약점 진단에 따른 취약점 진단 보고서(예: 발견된 취약점들 및 상세 원인 분석부터 해결방법까지)를 열람할 수 있게 한다.

[0050] 이로 인해, 본 발명의 취약점 진단 시스템에 따르면, 보안상의 위협을 급감시킬 수 있고, 관리자가 효율적으로 클라이언트 시스템을 안전한 상태로 유지 시킬 수 있다. 이 외에도 본 발명에서는, 진단서버(100, IR System)가 갖는 부가적인 그래픽 인터페이스를 통한 편리한 사용성과 사용자가 보기 편하도록 만들어진 진단 결과 제공 등의 효과를 기대할 수 있다.

[0051] 그리고, 본 발명의 취약점 진단 시스템 특히 진단서버(100, IR System) 도입 시, 주요기반시설, 금융기반 시설 등 취약점 점검/진단과 관련된 법적 보안 규제 준수를 만족하게 되고, 주기적인 취약점 점검/진단을 통한 보안 사고에 대한 사전예방을 강화할 수 있다.

[0052] 더 나아가, 본 발명에서는, 전술한 바와 같은 클라이언트(예: PC, 네트워크 장비 등)에 대한 보안 취약점 진단을 수행함에 있어, 진단 결과의 신뢰도를 높이고 진단 과정에서의 불필요한 부하 및 비용을 절감할 수 있는 구체화된 기술 구성을 실현하는데 특징이 있다.

[0053] 이하에서는, 본 발명에서 실현하는 특징 기술 구성에 대해, 도 2를 참조하여 설명하겠다.

[0054] 도 2에 도시된 바와 같이, 본 발명의 일 실시 예에 따른 취약점 진단 방법에서는, 사용자가 자신의 PC(이하, 클라이언트라 함)를 통해 진단서버(100)에 접속 및 로그인한 후 PC 즉 클라이언트에 대해 취약점 진단에서 요구하는 정보를 입력 및 취약점 진단을 요청할 수 있다(S10).

[0055] 이렇게 되면, 본 발명의 일 실시 예에 따른 취약점 진단 방법에서 진단서버(100)는, 전술과 같이 취약점 진단을 요청하는 클라이언트로, 원격지의 관리자가 웹기반 터미널 에뮬레이터 기반의 접속 서비스를 이용하여 접속되도록 한다(S20).

[0056] 본 발명에서는, 리눅스(Linux) 운영체제를 기반으로 구체적인 실시 예들을 설명하고 있다.

[0057] 즉, 본 발명의 일 실시 예에 따른 취약점 진단 방법에서 진단서버(100)는, 접속 및 로그인한 관리자가 취약점 진단을 요청하는 클라이언트의 정보를 토대로 웹기반 터미널 에뮬레이터(예: shell in a box)를 띄워 리눅스 OS에서 지원하는 접속 서비스(SSH) 기반의 접속을 시도하면, 해당 클라이언트에 접속되도록 할 수 있다.

[0058] 물론, 본 발명에서는, 리눅스가 아닌 다른 운영체제 예컨대 윈도의 OS의 경우라면, 윈도우 OS에서 지원하는 접속 서비스(예: Xshell, PuTTY, iTerm 등)를 이용하여 원격지의 관리자가 클라이언트에 접속되도록 할 수도 있다.

[0059] 다만 이하 설명에서는, 설명의 편의 상 계속해서 리눅스 OS를 기반으로 하여 실시 예를 설명하겠다.

[0060] 본 발명의 일 실시 예에 따른 취약점 진단 방법에서 진단서버(100)는, 원격지의 관리자가 클라이언트에 접속되면, 관리자에 의한 특정 스크립트를 해당클라이언트로 전달하여, 해당 클라이언트에서 특정 스크립트를 실행할 수 있게 한다(S30).

[0061] 즉, 본 발명의 일 실시 예에 따른 취약점 진단 방법에서 진단서버(100)는, 클라이언트에 접속된 원격의 관리자(컴퓨터) 바탕화면에 있던 특정 스크립트(이하, IR.tar라 함)를 scp 명령어를 사용하여 전송하면, 특정 스크립트 즉 IR.tar를 진단 대상의 클라이언트 즉 관리자가 접속된 상태의 클라이언트로 전달할 수 있다.

[0062] 이에, 본 발명에서는, 특정 스크립트 즉 IR.tar를 전달받은 클라이언트에서, IR.tar의 압출을 풀고 실행하면 클라이언트에서 취약점 진단 절차가 시작될 수 있다(S40).

[0063] 즉, 본 발명의 일 실시 예에 따른 취약점 진단 방법에서 진단서버(100)는, 클라이언트에서 IR.tar 실행을 통해 취약점 진단 절차가 수행되도록 함으로써, 클라이언트에 보안 취약점이 존재하는지 여부가 점검/진단되도록 한다(S50).

[0064] 이때, 본 발명에서는, 취약점 진달 절차에 따른 진단 결과의 신뢰도를 향상시키고 불필요한 부하 및 비용 소요를 최소화하기 위한 다음의 기술 구성들을 제안한다.

[0065] 구체적인 실시 예를 설명하면, 본 발명의 경우, 취약점 진단 절차에는 기 설정된 다수의 점검 항목이 포함되는 것으로 정의한다.

[0066] 이에 구체적인 실시 예에 따르면, 본 발명의 일 실시 예에 따른 취약점 진단 방법에서 진단서버(100)는, IR.tar 실행을 통해 클라이언트에서 취약점 진단 절차가 수행되도록 함에 있어, 취약점 진단 절차에 기 설정된 다수의 점검 항목 중에서, DB에서 조회되는 해당 클라이언트의 과거 진단 결과 데이터 및 해당 클라이언트의 진단 요청을 처리하는 원격지의 관리자 중 적어도 하나에 따라 선택되는 점검 항목으로 이루어지는 취약점 진단 절차가 수행되도록 할 수 있다(S50).

[0067] 예를 들어, 점검 항목 선택 및 선택된 점검 항목으로 이루어진 동적 취약점 진단 절차가 수행되도록 하는 주체는, 관리자 측일 수도 있고, 클라이언트에서 실행되는 특정 스크립트(예: IR.tar)이거나 진단서버(100)일 수 있다.

[0068] 구체적인 일 예를 들면, 진단서버(100)는, DB에서 조회되는 해당 클라이언트의 과거 진단 결과 데이터를 근거로, 취약점 진단 절차에 기 설정된 다수의 점검 항목 중에서 적어도 하나의 점검 항목을 선택할 수 있다.

[0069] 앞서 설명한 바 있듯이, DB(도 1 참조)에는, 클라이언트 별로 기 수행한 취약점 진단 절차에 대한 진단 결과 데이터(+ 취약점 진단 보고서)가 저장/관리되고 있다.

[0070] 이에, 본 발명에서 진단서버(100)는, 관리자에 의한 특정 스크립트 즉 IR.tar를 클라이언트로 전달하는 시점에 해당 클라이언트의 과거 진단 결과 데이터를 DB에서 조회할 수 있으며, 조회된 과거 진단 결과 데이터를 근거로 과거 일정 기간(예: N 개월, 또는 직접부터 N회의 과거 취약점 진단 절차) 동안에 취약점이 존재하지 않는다고 판단(예: 진단 결과 "만족")된 점검 항목을 취약점 진단 절차에 기 설정된 다수의 점검 항목에서 제외하고 남은 점검 항목을 선택할 수 있다.

[0071] 즉, 과거 일정 기간(예: N 개월, 또는 직접부터 N회의 과거 취약점 진단 절차) 동안에 취약점이 존재하지 않는

다고 판단(예: 진단 결과 "만족")된 점검 항목은, 현 시점에 수행할 취약점 진단 절차에 포함시키더라도 취약점이 존재하지 않는다고 판단(예: 진단 결과 "만족")될 가능성이 높으므로, 진단 대상의 점검 항목에서 제외하는 것이다.

[0072] 그리고, 진단서버(100)는, 클라이언트의 과거 진단 결과 데이터를 근거로 선택된 점검 항목들로 이루어지는 취약점 진단 절차가 수행되도록 할 수 있다.

[0073] 이렇게 되면, 본 발명에서는, 동일 클라이언트에 대해 매번 동일하게 보안 취약점 진단 절차를 수행하는 기존과 달리, 동일 클라이언트에 대해 취약점이 존재하지 않는다고 판단(예: 진단 결과 "만족")될 가능성이 높은 점검 항목을 제외한 나머지 점검 항목으로 선택/이루어진 동적인 취약점 진단 절차를 수행함으로써, 보안 취약점 진단 과정에서의 불필요한 부하 및 비용 소요를 최소화하여 줄이는 효과를 기대할 수 있다.

[0074] 또 다른 일 예를 들면, 진단서버(100)는, 클라이언트의 진단 요청을 처리하는 원격지의 관리자를 근거로, 취약점 진단 절차에 기 설정된 다수의 점검 항목 중에서 적어도 하나의 점검 항목을 선택할 수 있다.

[0075] 구체적으로, 본 발명에서 진단서버(100)는, 원격지의 관리자가 클라이언트로의 접속 서비스(SSH) 기반 접속을 시도하는 시점에 해당 관리자에 대한 레벨 또는 전문성 점수 또는 권한 등을 확인할 수 있다. 이를 위해서는, 기 정의된 정책에 따라서, 관리자 별로 과거에 진행했던 취약점 진단 절차들에 대한 평가가 이루어지고 평가에 따른 레벨 또는 전문성 점수 또는 권한 등이 부여될 수 있다.

[0076] 이에, 진단서버(100)는, 관리자에 대해 확인한 레벨 또는 전문성 점수 또는 권한 등을 근거로, 레벨 또는 전문성 점수 또는 권한 등이 높을수록 보안 이슈가 큰 점검 항목을 단계적으로 선택할 수 있게 하는 방식으로, 취약점 진단 절차에 기 설정된 다수의 점검 항목 중에서 적어도 하나의 점검 항목을 선택할 수 있다.

[0077] 그리고, 진단서버(100)는, 관리자를 근거로 선택된 점검 항목들로 이루어지는 취약점 진단 절차가 수행되도록 할 수 있다.

[0078] 이렇게 되면, 본 발명에서는, 동일 클라이언트에 대해 매번 동일하게 보안 취약점 진단 절차를 수행하는 기존과 달리, 금번 관리자의 레벨 또는 전문성 점수 또는 권한 등에 따라 선택되는 점검 항목으로 이루어진 동적인 취약점 진단 절차를 수행함으로써, 관리자에 따라 달라질 수 있는 진단 결과의 정확도 및 정당성, 보안 이슈를 만족시켜 진단 결과의 신뢰도를 향상시키는 효과를 기대할 수 있다.

[0079] 물론, 진단서버(100)는, 클라이언트의 과거 진단 결과 데이터 및 해당 클라이언트의 진단 요청을 처리하는 원격지의 관리자를 조합(필요 시 과거 진단 결과 데이터 및 관리자 각각에 가중치를 적용)하여, 그 조합 결과를 근거로 취약점 진단 절차에 기 설정된 다수의 점검 항목 중 점검 항목을 선택하여 선택된 점검 항목들로 이루어지는 취약점 진단 절차가 수행되도록 할 수도 있다.

[0080] 한편, 본 발명의 일 실시 예에 따른 취약점 진단 방법에서 진단서버(100)는, 전술과 같이 IR.tar 실행을 통해 클라이언트에서 취약점 진단 절차가 수행되도록 함에 있어, 기 특정된 중요 파일에 대해서는 상기 취약점 진단 절차 수행 중에 접근되는 일이 없도록 예외 처리하도록 한다(S50).

[0081] 여기서, 특정된 중요 파일은, 클라이언트의 파일시스템의 데이터 구조로부터 확인되는 파일 생성시점 및 파일 위치 중 적어도 하나를 근거로, 루트 파일시스템(root filesystem) 생성 시점에 생성된 것으로 식별되는 파일인 것으로 설명할 수 있다.

[0082] 안전한 컴퓨팅 환경을 위해 진행되는 취약점 진단 절차는 진단 항목이 달라지거나 진단 방식은 달라질 수 있겠지만, 결국 그 성능에 가장 중요한 영향을 미치는 공통된 점은 클라이언트 내 시스템에 영향을 미치지 않으면서도 신뢰도 높은 진단 결과를 얻는 것이라 하겠다.

[0083] 즉, 취약점 진단 과정 중에 접근해선 안 되는 중요 시스템파일에 접근하게 되는 경우가 발생한다면, 클라이언트 시스템에 치명적인 상황을 야기시키는 문제가 발생될 수 있고 점검 결과의 신뢰도 역시 낮아질 것이다.

[0084] 이에, 취약점 진단 절차 수행 중에 접근되는 일 없이 예외 처리해야만 하는 중요 파일(이하, 중요 시스템파일)은 반드시 식별/판단될 수 있어야만 한다.

[0085] 예를 들면, 윈도우 OS의 경우, OS 제작사가 배포한 중요 시스템파일에 대해서는 제작사의 디지털서명(또는 인증서)을 가지고 있어, 해당 디지털서명을 통해 예외 처리해야만 하는 중요 시스템파일인지 여부를 식별/판단할 수 있도록 하고 있다.

[0086] 이러한 점을 고려하여, 본 발명에서 진단서버(100)는, 윈도우 OS의 경우, 전술과 같이 IR.tar 실행을 통해 클라이언트에서 취약점 진단 절차가 수행되도록 함에 있어, 디지털서명(또는 인증서)를 통해 식별되는 중요 시스템 파일에 대해서는 취약점 진단 절차 수행 중에 접근되는 일이 없도록 예외 처리하도록 할 수 있다.

[0087] 헌데, 리눅스 OS의 경우, 윈도우 OS와 달리 디지털서명(또는 인증서)을 지원하고 있지 않기 때문에, 예외 처리해야만 하는 중요 시스템파일을 식별/판단해내기 어렵다.

[0088] 이러한 점을 고려하여, 본 발명에서 진단서버(100)는, 리눅스 OS의 경우, 전술과 같이 IR.tar 실행을 통해 클라이언트에서 취약점 진단 절차가 수행되도록 함에 있어, 클라이언트의 파일시스템의 데이터 구조로부터 확인되는 파일 생성시점 및 파일 위치 중 적어도 하나를 근거로, 루트 파일시스템(root filesystem) 생성 시점에 생성된 것으로 식별되는 파일을 중요 시스템파일로 판단하고, 이러한 중요 시스템파일에 대해서는 취약점 진단 절차 수행 중에 접근되는 일이 없도록 예외 처리하도록 할 수 있다.

[0089] 예를 들어, 중요 시스템파일을 식별 및 취약점 진단 절차 수행 중에 접근되는 일이 없도록 예외 처리하도록 하는 주체는, 클라이언트에서 실행되는 특정 스크립트(예: IR.tar)이거나 진단서버(100)일 수 있다.

[0090] 보다 구체적으로 설명하면, 리눅스 OS에서 사용되고 있는 파일시스템의 기본적인 데이터 구조는, 부트 블록과 여러 개의 블록 그룹(0,1,...,n)으로 구성될 수 있다.

[0091] 각각의 블록 그룹은, 슈퍼 블록, 그룹 디스크립터, 블록 비트맵, I-노드 비트맵, 그리고 디렉토리 및 데이터 블록으로 구성된다.

[0092] 이러한 블록 그룹의 구조는, 주요 데이터와 파일 데이터 의 집약도를 높여주기 때문에 파일을 저장할 때 단편화를 줄여주며, 디스크 I/O와 관련한 응답 속도를 줄여줄 수 있는 계기가 된다.

[0093] 본 발명에서는, 이러한 파일시스템의 데이터 구조에서 특히 블록 비트맵 및 I-노드 비트맵을 활용한다.

[0094] 블록 비트맵은, 블록 그룹의 구조에서 그룹 디스크립터의 다음에 위치한다. 블록 비트맵은, 말 그대로 블록의 사용(할당) 상태를 비트로 표현하여 나타낸 것이다. 블록 그룹 내에 존재하는 각각의 블록은 블록 비트맵에서 하나하나의 비트에 해당한다. 그래서 특정 블록이 사용되고 있으면 블록 비트맵의 해당 블록의 인덱스가 할당(사용)된 상태임을 나타내게 된다.

[0095] 여기서, 블록 그룹의 데이터 구조에서는, 낮은 넘버의 블록부터 차례대로 사용(할당)하는 특징이 있다.

[0096] Ext filesystem에서 파일 데이터가 저장되는 곳은 I-노드 데이터 구조이다. 모든 파일들과 디렉토리들은 각각 1개의 I-노드를 할당하고 있으며, 모든 I-노드들은 고유한 주소(또는 인덱스)를 가지고 있다.

[0097] 따라서, 리눅스 OS에서 사용되고 있는 기본적인 파일시스템의 데이터 구조는, I-노드로 식별되는 위치에 블록 단위로 파일을 저장하는 데이터 구조라 할 수 있다.

[0098] I-노드 비트맵은, 블록 비트맵과 마찬가지로, 해당 블록 그룹이 관리하는 모든 I-노드의 사용(할당) 상태를 비트로 표현하여 나타낸 것이다. 즉, 블록 그룹 내에 존재하는 각각의 I-노드는 I-노드 비트맵에서 하나하나의 비트에 해당한다. 그래서 특정 I-노드가 사용되고 있으면 I-노드 비트맵의 해당 I-노드의 인덱스가 할당(사용)된 상태임을 나타내게 된다.

[0099] 여기서, 블록 그룹의 데이터 구조에서는, 낮은 넘버의 I-노드부터 차례대로 사용(할당)하는 특징이 있다.

[0100] 다시 중요 시스템파일을 식별/판단하는 방식을 설명하면, 진단서버(100)는, 클라이언트의 파일시스템의 데이터 구조로부터, 블록 비트맵 및 I-노드 비트맵에서 파일 데이터가 가장 먼저 저장되는 가장 작은 블록 넘버 및 I-노드 넘버 중 적어도 하나의 생성시간을 확인할 수 있게 한다. 이때의 생성시간은, 파일 생성시점을 의미할 것이다.

[0101] 그리고, 진단서버(100)는, 블록 비트맵 및 I-노드 비트맵으로부터, 상기 확인한 생성시간과 같은 생성시간을 가지며 기 설정된 임계 블록 넘버 및 I-노드 넘버 중 적어도 하나 보다 작은 넘버를 가지는 파일을 확인할 수 있게 한다. 이때의 넘버는 파일 위치를 의미할 것이다.

[0102] 여기서, 기 설정된 임계 블록 넘버 및 I-노드 넘버는, 클라이언트 내 파일시스템의 실제 중요 시스템파일 모두에 대하여 블록 및 I-노드를 사용(할당)하는 경우, 최대로 사용하게 되는 블록 넘버 및 I-노드 넘버와 동일하거나 일정 크기 큰 넘버로 기 설정되는 것이 바람직하다.

[0103] 이에, 진단서버(100)는, 블록 비트맵 및 I-노드 비트맵에서 가장 작은 블록 넘버 및 I-노드 넘버 중 적어도 하나의 생성시간, 그리고 임계 블록 넘버 및 I-노드 넘버 중 적어도 하나 보다 작은 넘버를 가지는 것으로 확인되는 파일을, 루트 파일시스템 생성 시점에 생성된 파일로 식별하게 하여, 이 파일 즉 중요 시스템파일에 대해서는 취약점 진단 절차 수행 중에 접근되는 일이 없도록 예외 처리하도록 할 수 있다.

[0104] 이렇게 되면, 본 발명에서는, 중요 시스템파일을 식별하고 보안 취약점 진단 과정 중에 중요 시스템파일에 접근되는 일 이 없도록 예외 처리하는 취약점 진단 절차를 수행함으로써, 진단 결과의 신뢰도를 향상시키고 아울러 클라이언트 시스템에 치명적인 상황을 야기시키는 문제 발생을 회피하는 효과를 기대할 수 있다.

[0105] 계속해서 설명하면, 본 발명의 일 실시 예에 따른 취약점 진단 방법에서 진단서버(100)는, S50단계에서 수행되는 취약점 진단 절차에 따라 생성되는 진단 결과 데이터가 DB(도 1 참조)에 저장/관리될 수 있도록 한다.

[0106] 구체적으로 설명하면, 본 발명에서는, 클라이언트에서 취약점 진단 절차가 끝나면, 취약점 진단 절차에 따른 진단 결과 데이터가 생성된다. 예를 들면, 진단 결과 데이터는, 날짜 + IP주소.txt 형태로서, 모든 결과가 들어있는 txt 파일로 생성될 수 있다.

[0107] 이때 본 발명의 일 실시 예에 따른 취약점 진단 방법에서는, 생성된 진단 결과 데이터가 웹에서 표시 가능한 형태로 가공된 후 DB(도 1 참조)에 저장되도록 한다.

[0108] 즉, 이에 클라이언트에서는, 취약점 진단 절차를 통해 생성된 진단 결과 데이터(예: 날짜 + IP주소.txt 파일)를 가공시켜 줄 "./final_DB.sh"라는 스크립트를 실행시켜 진단 결과 데이터(예: 날짜 + IP주소.txt 파일)를 웹에서 표시 가능한 형태로 가공한 후(S60), mysql 명령어를 이용하여 DB에 저장할 수 있다(S80).

[0109] 예를 들어, 진단 결과 데이터가 가공 후 DB에 저장되도록 하는 주체는 클라이언트에서 실행되는 특정 스크립트(예: IR.tar)이거나 진단서버(100)일 수 있다.

[0110] 도 3는 본 발명의 취약점 진단 방법에 의해 얻을 수 있는 진단 결과 데이터의 일 예를 도시하고 있다.

[0111] 그리고, 도 4는 도 3의 진단 결과 데이터가 웹에서 표시 가능한 형태로 가공된 이후를 보여주는 일 예시도이다.

[0112] 계속해서 설명하면, 본 발명의 일 실시 예에 따른 취약점 진단 방법에서 진단서버(100)는, 사용자(클라이언트)가 로그인을 통해 진단서버(100)에 접속하면, DB 내 저장 데이터를 기반으로 고객 페이지(마이 페이지)에서 앞서 진행한 취약점 진단 절차의 진단 결과 데이터에 따른 취약점 진단 보고서를 열람할 수 있게 한다(S90).

[0113] 도 4에서 알 수 있듯이, 취약점 진단 보고서에는, 사용자의 회원 정보, 진단 결과, 상세 결과가 포함될 수 있으며, 상세 결과에는 진행한 점검 항목 별로 항목 코드, 중요도, 점검 현황, 진단 결과, 그리고 조치 사항(해결방법)이 포함되어, 사용자는 취약점 진단 보고서를 통해 자신의 PC(클라이언트)에서 발견된 취약점들 및 상세 원인 분석부터 해결방법까지 확인할 수 있게 된다.

[0114] 한편, 취약점 진달 절차에 따른 진단 결과의 신뢰도를 향상시키기 위한 또 다른 실시 예를 설명하면, 본 발명의 일 실시 예에 따른 취약점 진단 방법에서 진단서버(100)는, 진단 결과 데이터가 생성된 시점(가공 이전)에, 금번 진단 결과 데이터를 근거로 해당 클라이언트에서 취약점으로 확인되는 특정 점검 항목과 관련하여 기 설정된 모의 공격을 수행해 볼 수 있다(S70).

[0115] 예를 들어, 도 4를 참조하여 설명하면, 진단서버(100)는, 항목 코드 U-01, U-04, U-45와 같이 진단 결과가 "부분만족"인 점검 항목과 관련하여, 해당 점검 항목 별로 기 설정된 모의 공격을 수행할 수 있고, 또는 중요도가 "상"이면서 진단 결과가 "부분만족"인 점검 항목과 관련하여, 해당 점검 항목 별로 기 설정된 모의 공격을 수행할 수도 있고, 중요도가 "상"인 점검 항목 모두에 대해 해당 점검 항목 별로 기 설정된 모의 공격을 수행할 수도 있다.

[0116] 그리고, 진단서버(100)는, 해당 클라이언트에 대한 금번 진단 결과 데이터가 DB에 저장될 때 모의 공격 수행 결과가 함께 저장되도록 할 수 있고, 또는 DB에 저장되는 금번 진단 결과 데이터에 모의 공격 수행 결과를 반영 및 처리한 후 저장되도록 할 수 있다(S80).

[0117] 이렇게 되면, 본 발명에서는, 사용자(클라이언트)가 로그인을 통해 진단서버(100)에 접속하여 웹 상의 고객 페이지(마이 페이지)를 통해, 취약점 진단 절차를 수행하여 얻어지는 진단 결과에서 더 나아가 모의 공격 수행 결과까지 반영된 취약점 진단 보고서를 조회할 수 있도록 제공함으로써, 단순한 진단 결과 보다 더 강력한 정확도가 필요한 상황까지도 만족시켜 취약점 진단의 신뢰도를 향상시키는 효과를 기대할 수 있다.

[0118] 이상 설명한 바와 같이, 본 발명의 실시 예들에 따르면, 클라이언트의 과거 진단 결과 및 원격지의 관리자에 따라 선택되는 점검 항목으로 이루어진 동적 취약점 진단 절차를 수행하는 방식, 진단 결과 기반의 모의 공격을 수행하는 방식, 취약점 진단 절차 중에 중요 시스템파일에 접근되는 일 이 없도록 예외 처리하는 방식 실현을 통해, 새로운 방식의 취약점 진단 기술을 구현해 내고 있다.

[0119] 이로써, 본 발명에 따르면, 전술과 같이 구현한 새로운 방식의 취약점 진단 기술을 통해 취약점 진단을 수행함으로써, 취약점 진단 절차를 수행함에 있어서 진단 결과의 신뢰도를 높이고 진단 과정에서의 불필요한 부하 및 비용을 절감할 수 있는 효과를 도출할 수 있다.

[0120] 본 발명의 실시 예에 따르는 취약점 진단 방법은, 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0121] 이상과 같이 본 발명에서는 구체적인 구성 요소 등과 같은 특정 사항들과 한정된 실시예 및 도면에 의해 설명되었으나 이는 본 발명의 보다 전반적인 이해를 돕기 위해서 제공된 것일 뿐, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상적인 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다.

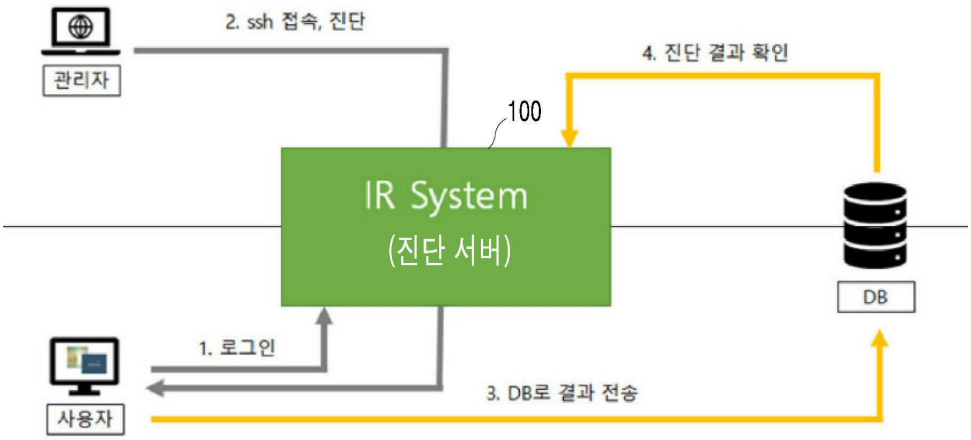
[0122] 따라서, 본 발명의 사상은 설명된 실시예에 국한되어 정해져서는 아니되며, 후술하는 특허청구범위뿐 아니라 이 특허청구범위와 균등하거나 등가적 변형이 있는 모든 것들은 본 발명 사상의 범주에 속한다고 할 것이다.

부호의 설명

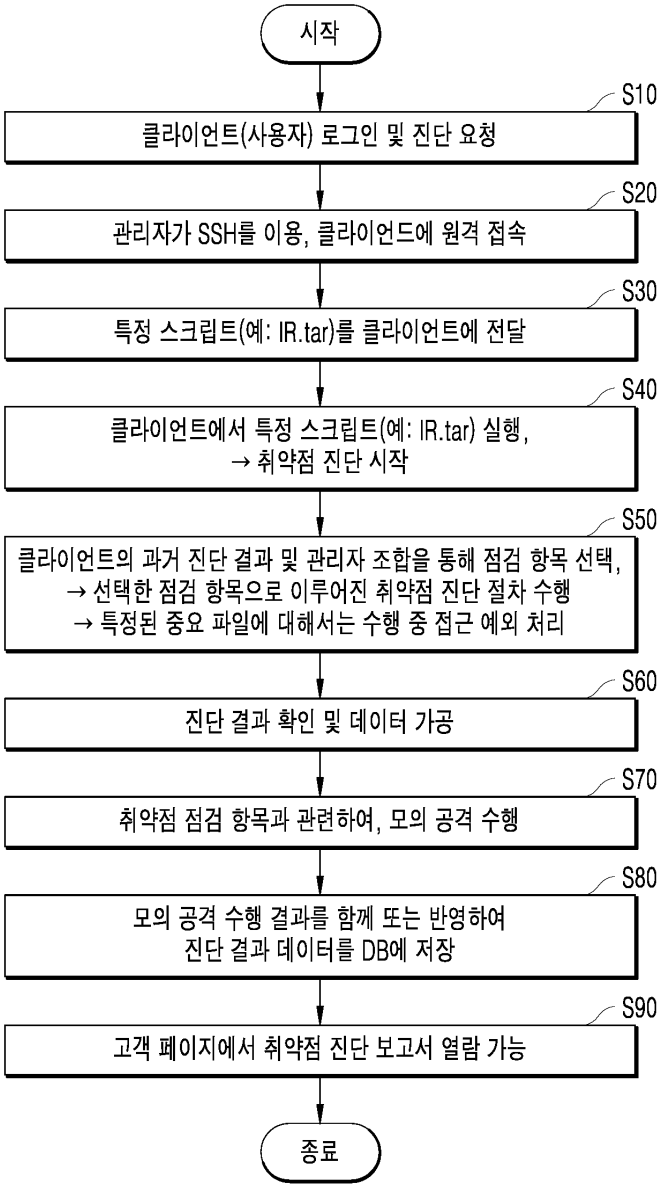
[0123] 100 : 진단서버(IR System)

도면

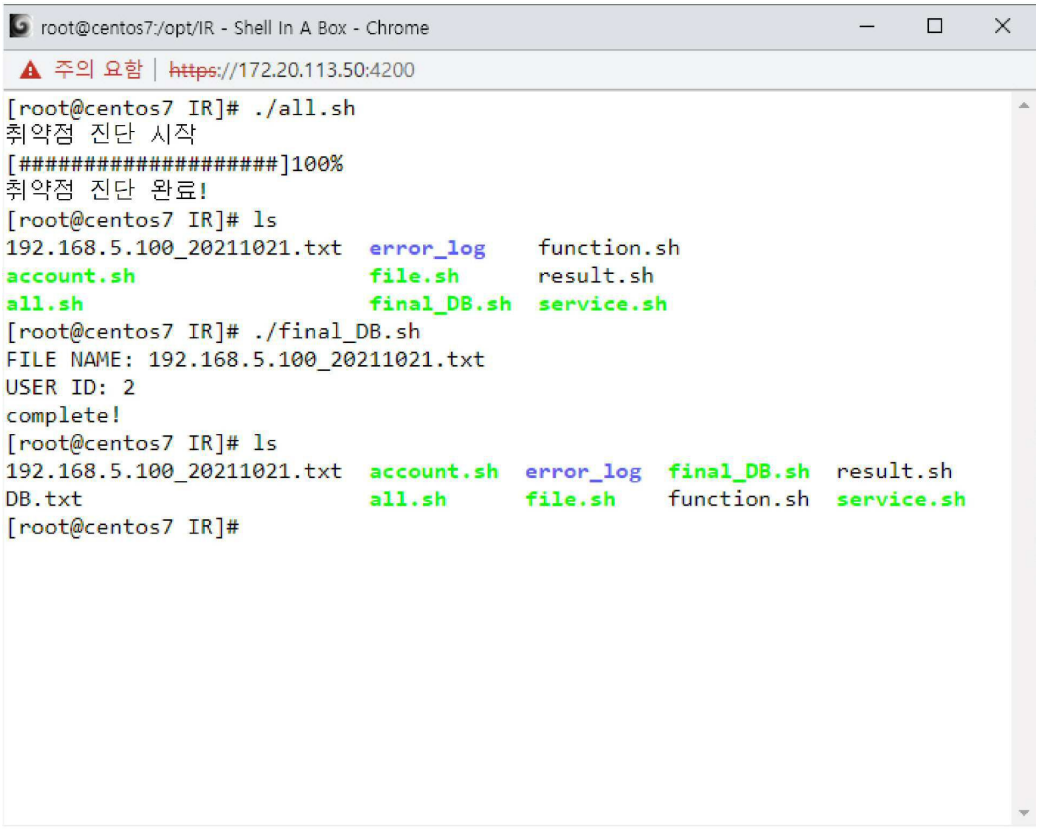
도면1



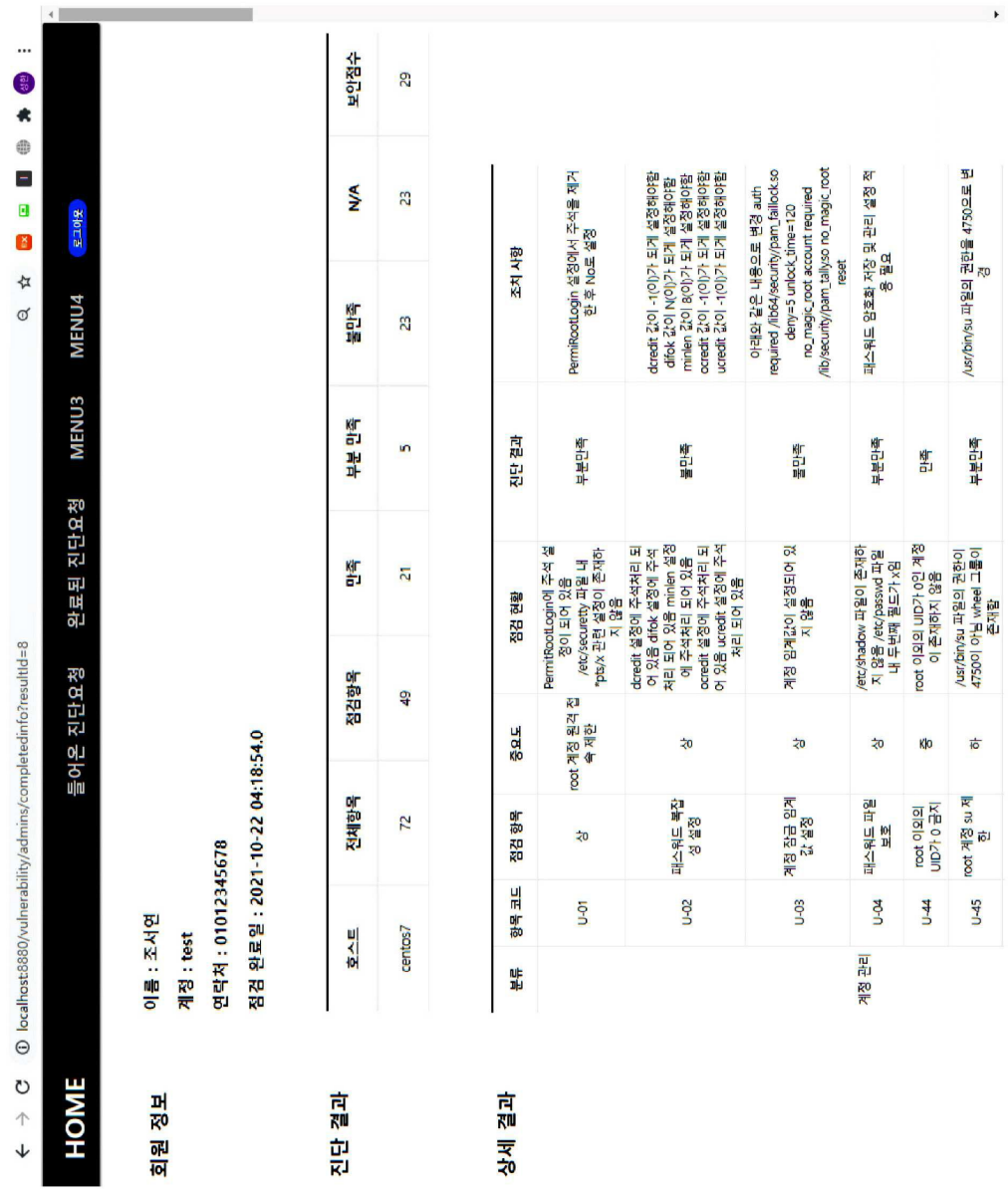
도면2


시작
S10
클라이언트(사용자) 로그인 및 진단 요청
S20
관리자가 SSH를 이용, 클라이언드에 원격 접속
S30
특정 스크립트(예: IR.tar)를 클라이언트에 전달
S40
클라이언트에서 특정 스크립트(예: IR.tar) 실행,
→ 취약점 진단 시작
S50
클라이언트의 과거 진단 결과 및 관리자 조합을 통해 점검 항목 선택,
→ 선택한 점검 항목으로 이루어진 취약점 진단 절차 수행
→ 특정된 중요 파일에 대해서는 수행 중 접근 예외 처리
S60
진단 결과 확인 및 데이터 가공
S70
취약점 점검 항목과 관련하여, 모의 공격 수행
S80
모의 공격 수행 결과를 함께 또는 반영하여
진단 결과 데이터를 DB에 저장
S90
고객 페이지에서 취약점 진단 보고서 열람 가능
종료

도면3


root@centos7:/opt/IR - Shell In A Box - Chrome
주의 요함 | https://172.20.113.50:4200
[root@centos7 IR]# ./all.sh
취약점 진단 시작
[####################]100%
취약점 진단 완료!
[root@centos7 IR]# ls
192.168.5.100_20211021.txt error_log function.sh
account.sh file.sh result.sh
all.sh final_DB.sh service.sh
[root@centos7 IR]# ./final_DB.sh
FILE NAME: 192.168.5.100_20211021.txt
USER ID: 2
complete!
[root@centos7 IR]# ls
192.168.5.100_20211021.txt account.sh error_log final_DB.sh result.sh
DB.txt all.sh file.sh function.sh service.sh
[root@centos7 IR]#

도면4


HOME
들어온 진단요청
완료된 진단요청
MENU3
MENU4
로그아웃
localhost:8880/vulnerability/admins/completedinfo?resultId=8
회원 정보
이름 : 조서연
계정 : test
연락처 : 01012345678
점검 완료일 : 2021-10-22 04:18:54.0
진단 결과
호스트 전체항목 점검항목 만족 부분 만족 불만족 N/A 보안점수
centos7 72 49 21 5 23 23 29
상세 결과
분류 항목 코드 점검 항목 중요도 점검 현황 진단 결과 조치 사항
계정 관리 U-01 상 root 계정 원격 접속 제한 PermitRootLogin에 주석 설정이 되어 있음 /etc/securetty 파일 내 *pts/x 관련 설정이 존재하지 않음 부분만족 PermitRootLogin 설정에서 주석을 제거한 후 No로 설정
U-02 패스워드 복잡성 설정 상 dcredit 설정에 주석처리 되어 있음 difok 설정에 주석처리 되어 있음 minlen 설정에 주석처리 되어 있음 ocredit 설정에 주석처리 되어 있음 ucredit 설정에 주석처리 되어 있음 불만족 dcredit 값이 -1이가 되게 설정해야함 difok 값이 N이가 되게 설정해야함 minlen 값이 8이가 되게 설정해야함 ocredit 값이 -1이가 되게 설정해야함 ucredit 값이 -1이가 되게 설정해야함
U-03 계정 잠금 임계값 설정 상 계정 임계값이 설정되어 있지 않음 불만족 아래와 같은 내용으로 변경 auth required /lib64/security/pam_faillock.so deny=5 unlock_time=120 no_magic_root account required /lib/security/pam_tally.so no_magic_root reset
U-04 패스워드 파일 보호 상 /etc/shadow 파일이 존재하지 않음 /etc/passwd 파일 내 두번째 필드가 x임 부분만족 패스워드 암호화 저장 및 관리 설정 적용 필요
U-44 root 이외의 UID가 0 금지 중 root 이외의 UID가 0인 계정이 존재하지 않음 만족
U-45 root 계정 su 제한 하 /usr/bin/su 파일의 권한이 4750이 아닌 wheel 그룹이 존재함 부분만족 /usr/bin/su 파일의 권한을 4750으로 변경