JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 27 No. 12, pp. 179-188, December 2022
https://doi.org/10.9708/jksci.2022.27.12.179

A Study on Methods for Efficient Enterprise Software Patch Management

Chang-Hoon Kang*

*Director, Solution Bank Co., Ltd, Jeonju, Korea

[Abstract]

In this paper, we propose an efficient and procedural software patch management phases. Every year, organizations have tens of thousands of known vulnerabilities and spend tens of thousands of hours and millions of dollars or more patching them. Despite these efforts, research has shown that the time it takes for an exploit to appear for a given patched vulnerability is shortening. As various types of organizations such as companies and universities manage patches in inconsistent ways, problems such as security problems, system instability, information leakage and work delay due to patches occur. In this paper, we look at the basics required for software patch management and define the factors to be considered for patch management and the effective steps for patch management. Therefore, this study will be used as a method to efficiently and procedurally execute the organization's patch management policy in the process of updating and patching the software in the organization to a new version as a solution to software function modification and security vulnerability.

▸ **Key words**: Enterprise patch management, Patch, Software Patch, Vulnerability Management, Software Update

[요 약]

본 논문에서는 효율적이고 절차적인 소프트웨어 패치 절차를 제안하였다. 매년 조직에서는 수만 개의 알려진 취약점이 존재하고 패치 작업에 수만 시간과 수백만 달러 이상을 소비한다. 이러한 노력에도 불구하고 주어진 패치된 취약점에 대한 익스플로잇이 나타나는 데 걸리는 시간이 단축되고 있는 것으로 조사되었다. 기업, 대학 등 다양한 형태의 조직에서 일관되지 않는 방법으로 패치를 관리함에 따라 보안 문제 발생, 시스템 불안정, 패치로 인한 정보 유출 및 작업지연 등의 문제가 발생하고 있다. 본 연구에서는 소프트웨어 패치 관리를 위해 필요한 기본적인 사항들을 살펴보고 패치 관리를 위해 고려되어야 하는 사항과 패치 관리를 위한 효율적인 단계를 정의한다. 따라서 본 연구는 소프트웨어 기능의 수정, 보안 취약점에 대한 해결 방법으로 조직 내의 소프트웨어를 새로운 버전으로 갱신하고 패치하는 과정에서 조직의 패치 관리 정책을 효율적이고 절차적으로 실행하는 방안으로 활용될 것이다.

▸ **주제어**: 조직 패치관리, 패치, 소프트웨어 패치, 취약점 관리, 소프트웨어 업데이트

• First Author: Chang-Hoon Kang, Corresponding Author: Chang-Hoon Kang
*Chang-Hoon Kang (attozen@gmail.com), Solution Bank Co., Ltd
• Received: 2022. 12. 02, Revised: 2022. 12. 20, Accepted: 2022. 12. 22.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

소프트웨어 취약성을 악용하는 사이버 공격자는 여전히 현대가 직면한 가장 심각한 위험 중 하나이다. 기업 네트워크. 패칭은 조직 보호를 위한 가장 효과적이고 널리 알려진 전략이다. 이러한 사이버 공격에 대한 소프트웨어 시스템에 대한 보안 패치의 빠른 릴리스에도 불구하고 소프트웨어 제품에서 새로 발견된 취약점으로 인해 대부분의 사이버 공격이 발생하였다[1]. 패치가 이미 존재하는 알려진 취약점의 악용 [2,3,4,5]하여 더 불안한 이러한 사이버 공격으로 막대한 재정적 손실과 명성을 얻는 파괴적인 결과를 초래했다는 것이다.

Equifax 사건[6,7]과 같은 회사 데이터의 기밀성 및 무결성 위반으로 인한 손실, 심지어 소프트웨어 시스템의 가용성으로 인한 인간의 죽음 [8], 이러한 결과는 주로 조직 IT 환경에서 소프트웨어 보안 패치를 적용할 때 직면하는 본질적인 복잡한 문제 즉, 소프트웨어 보안 패치 관리는 보안 취약점에 대한 패치를 적용하는 과정을 의미한다. 관리되는 소프트웨어 시스템의 기존 취약점 식별, 획득, 테스트, 설치, 소프트웨어 보안 패치 확인[9,10,11,12]. 이러한 활동을 수행하려면 상호 종속성을 관리해야 한다. 그림 1은 패치 관리 기능의 3가지 핵심 운영 기능과 함께 기본 종속성이 보여준다. 패치 관리 기본 종속성은 효율성과 효과를 보장하기 위해 마련되어야 한다[13].

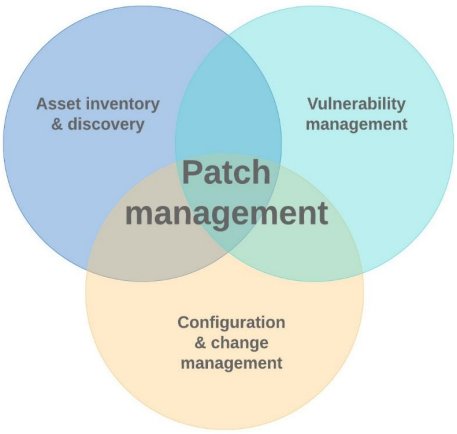

Fig. 1. Patch management foundational dependencies

소프트웨어 패치는 일반적으로 버그나 보안 취약점을 수정하는 데 사용되는 기존 소프트웨어에 대한 작거나 큰 수정 과정이다. 기업, 기관, 관공서 및 개인이 사용하는 모든 소프트웨어 및 시스템에 패치 과정은 적합하게 이루어져야 한다. 패치는 소프트웨어의 단순한 버그를 수정하는 과정부터 소프트웨어 기능 업그레이드, 보안 관련 기능 개선 등 여러 가지 원인으로 이루어지게 된다.

기관에서의 패치는 개인의 컴퓨터나 모바일 장비의 패치 영향이 기관 전체에 영향을 미칠 수 있다. 특히 소프트웨어, 컴퓨터 및 장비에 대한 취약점이 발견되어 패치가 이루어지기 전까지는 해킹에 쉽게 노출되어 질 수 있다. 따라서 기관에서는 적합하고 효율적인 패치 관리 방안을 마련하여 계획적이고 통합적으로 패치가 이루어져야 해킹에 대한 방어가 이루어질 수 있다[14,15,16].

Adam Mutrray[17]는 평균 애플리케이션에서 코드 기반의 10%만이 사내에서 작성되고 지난 18개월 동안 21,000개의 알려진 취약점(CVE)이 보고되었다는 사실을 고려하면 알려진 취약점이 가장 취약하다는 것을 이해할 수 있다.

패치를 적용하려면 시간과 노력이 필요하다. 매년 조직은 패치 작업에 18,000시간과 백만 달러 이상을 소비한다. 이러한 노력에도 불구하고 많은 사람들이 주어진 패치된 취약점에 대한 익스플로잇이 나타나는 데 걸리는 시간이 단축되는 것을 확인되었다. 강력한 패치 관리 모범 사례를 구현하지 않으면 시간을 낭비하고 공격의 문을 열어 둘 위험이 있다.

Ponemon Institute의 연구에서 사이버 공격 피해자의 57%는 패치를 적용하면 공격을 막을 수 있다고 했고 34%는 공격 전에 취약점에 대해 알고 있었다고 하였다.

철저한 패치 관리 프로세스는 완벽한 보안 프레임워크의 필수적인 부분이다. 올바른 응용 프로그램에 올바른 패치를 더 빨리 적용 할수록 환경이 더욱 안전해진다[17].

1. Related works

여러 기관에서 패치 관리 업무를 수행하고 있다. 취약성 보완을 위한 여러 가지 패치 관리 정책들[18,19,20,21,22]이 기관의 환경에 따라 아주 많이 사용되고 있다. 또한 컴퓨터나 정보기기 그리고 소프트웨어 방대한 종류는 엄청나게 많이 존재한다. 이와 같이 패치 업무는 기관마다 다양하게 이루어지고 있다. 그중 두 가지 경우의 관련 연구를 살펴본다.

Adam Mutrray[17]는 원활한 패치 관리 프로세스를 위한 8가지 모범 사례를 제공하였다. 시스템 인벤토리(Inventory)를 구축은 환경 내의 모든 소프트웨어 및 하드웨어에 대한 포괄적인 인벤토리는 모든 패치 관리 프로세스의 중요한 부분이다. 보유하고 있는 항목에 대한 명확한 그림이 있으면 알려진 취약점을 인벤토리와 비교하여 중요한 패치를 빠르게 찾을 수 있다.

사용하는 소프트웨어 버전(및 소프트웨어 자체) 통합하는 방법이 좋다. 사용하는 소프트웨어 버전이 많을수록 노

출 위험이 높아진다. 또한 많은 양의 관리 오버헤드를 생성한다. Windows, Linux 또는 MacO 중 하나의 버전을 선택하고 패치를 통해 해당 버전을 최신 상태로 유지하는 것이 좋다[17].

패치 예외 완화 방법을 마련해야 한다. 패치를 바로 적용할 수 없는 경우가 있다. 소수의 시스템이 체크아웃되면 회사 전체가 패치될 때까지 더 큰 그룹에 패치를 배포하기 시작해야 한다. 빠르게 패치를 적용한다는 것은 한 번에 모든 곳에 패치를 적용하는 것을 의미하지 않는다. 패치가 균열을 통해 떨어지지 않고 모든 것을 적시에 패치 할 계획이 있는지 확인해야 한다.

소프트웨어 보안 패치 관리는 식별, 획득, 소프트웨어 제품 및 시스템에 대한 보안 패치 테스트, 설치 및 확인의 과정이다. 그림 2는 소프트웨어 취약점 주기에 따른 쇼프트웨어 보안 패치 관리의 과정을 보여준다[1].

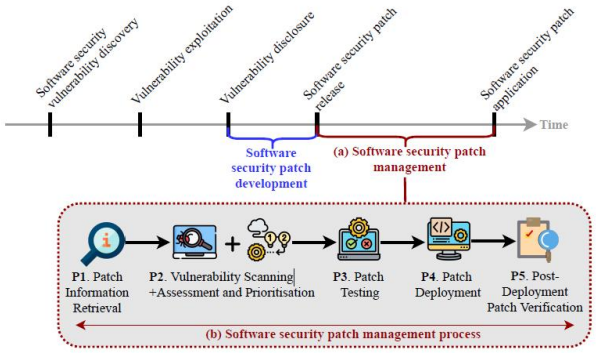

Fig. 2. An overview of the software security patch managemwnt process.

패키지 관리자 또는 패키지 관리 시스템은 일관된 방식으로 컴퓨터의 운영 체제에 대한 컴퓨터 프로그램을, 업그레이드를 설치, 구성 및 제거하는 과정을 자동화하는 소프트웨어 도구의 모음입니다. 그러나 이러한 보안 결함도 있을 수 있습니다. 사용자가 패키지 관리자를 사용하여 소프트웨어의 의미를 이해하지 못한 채 소프트웨어 측면에서 최신 상태를 유지하는 것이 일반적입니다. 패키지 관리자가 루트로 실행되고 제대로 구현되지 않으면 시스템에 악성 소프트웨어를 다운로드 하여 설치할 수 있습니다. 아무도 그것을 원하지 않을 것이라고 확신합니다.

패치 관리는 제품 및 시스템에 대한 패치를 평가, 획득, 테스트, 우선순위 지정, 배포 및 검증하는 프로세스이다. 반복 가능하고 표준화된 패치 관리 활동은 소프트웨어 결함 및 취약성을 완화하여 조직의 노출을 최소화하고 예방 가능한 손상을 방지함으로써 비용 절감을 지원한다. 효과적인 패치 관리를 위해서는 다양한 기업 관리자의 역할과 프로세스를 조정하여 이기종 IT 환경에서 구성을 최신 상태로 유지해야 한다. 보안 운영자와 서비스 운영자는 가용성에 대한 운영 요구 사항을 염두에 두고 시스템 및 애플리케이션 패치의 우선순위를 지정하고, 테스트하고, 적용하고, 확인하기 위해 협력해야 한다[10].

패치 관리는 조직의 전체 시스템 무결성 및 위험 관리 전략의 일부인 폐쇄 루프 프로세스이다. 프로세스는 취약성 관리와 자산 인벤토리 및 검색의 기초를 통합하는 엔터프라이즈 도구를 통해 반복 가능하고 대부분 자동화될 때 가장 잘 실행된다. 이 도구는 지속적인 모니터링 기능을 용이하게 하고 주기적 프로세스를 실행한다. 다음 그림 3은 패치 관리의 주기를 설명한다[10].

공고(Notification) 단계는 기업 패치 관리 소프트웨어는 애플리케이션 인벤토리 및 패치 정의 데이터베이스를 유지 관리하여 신뢰할 수 있는 소스에서 새 업데이트가 게시 및 다운로드될 때 관리자에게 알려준다. 평가(Assessment) 단계는 공급 업체나 개발자가 패치에 할당한 심각도 등급은 패치의 중요성과 우선순위를 나타내는 주요 지표이다. 그러나 패치 우선순위 지정은 운영 환경의 맥락에서 취약점이 나타내는 위험을 고려하는 위험 평가 활동이기도 하다. 고가치 자산에 대한 잠재적 영향, 위협 프로필, 악용 복잡성 및 가능성, 노출에 대한 완화 제어의 영향을 포함하여 패치관리 부서에서 패치의 우선순위를 결정할 때 고려해야 한다.

인수(Acquistion) 단계의 패치는 일반적으로 소프트웨어 공급업체 또는 기타 신뢰할 수 있는 소스(예: 커뮤니티 소프트웨어 개발 또는 표준 리눅스 리포지토리)에서 얻는다. 엔터프라이즈 패치 관리 소프트웨어는 일반적으로 무결성 체크섬 및 디지털 서명을 기반인 확인 방법을 사용하여 패치 다운로드 및 확인을 자동화할 수 있다[23].

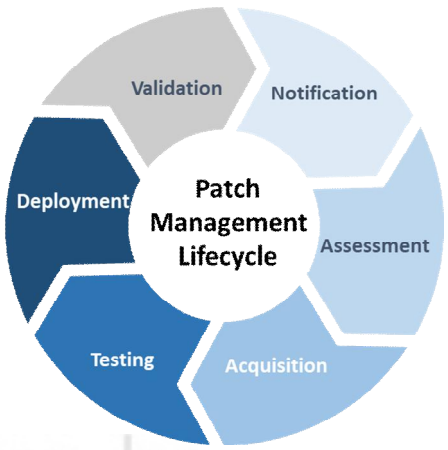

Fig. 3. Patch Management Lifecycle

테스트(Testing) 단계에서는 패치 배포를 하기전 패치에 대한 테스트가 이루어져야한다. 테스트 과정은 전체 기업의 시스템에 문제 발생 여부를 검토하는 과정이다. 패치 테스트는 프로덕션 환경을 시뮬레이션하는 테스트 또는 샌드박스 네트워크에서 수행 할 있다. 이것이 이상적인 접근 방식이지만 가장 비용이 많이 들고 비실용적이다.

배포(Deployment) 단계는 패치 소프트웨어를 배포하여 실제로 패치를 실행하는 과정이다. 기업 패치 관리 소프트웨어는 소프트웨어 업데이트 배포에 선호되는 방법이다. 패치는 번들로 제공되거나 점진적으로 적용될 수 있으며 테스트 단계에서 드러나지 않은 결함이 구체화 될 경우 서비스 중단이 발생하지 않도록 엔터프라이즈 전체에 걸쳐 시차를 두어야 한다.

자동 업데이트는 많은 애플리케이션에 공통적인 기능이지만 이 기능을 사용하면 프로세스에서 공급업체가 수행하는 것 이상으로 패치 테스트가 제거된다는 점을 감안할 때 일반적으로 기업 환경에는 권장되지 않는다. 하지만 웹 브라우저 및 모바일 앱의 경우와 같이 자동 업데이트가 선호되는 배포 방법이 될 수 있는 상황이 있다. 다른 시나리오가 적용될 수 있으며 패치관리 조직은 자동 업데이트 기능 활성 여부와 활성화 위치를 결정할 때 자체 위험 관리 실사를 수행해야 한다.

검증(Validation) 단계는 패치 배포에 성공 및 실패율에 대한 감사를 각 배포 후에 수행하여 이상치를 식별하고 패치 설치 실패를 추적 및 수정한다. 설치 실패의 조사 및 수정에서 얻은 통찰력은 패치 관리 프로그램을 보강하고 향후 유사한 롤백 및 수정 활동의 필요성을 최소화하는 데 사용해야 한다[23]. 패치 관리 오퍼레이션은 여러 가지 조직 및 프로그램에 따라 많은 형태가 존재 한다. 그림 4는 패치 오퍼레이션의 한가지 예를 보여준다[24].

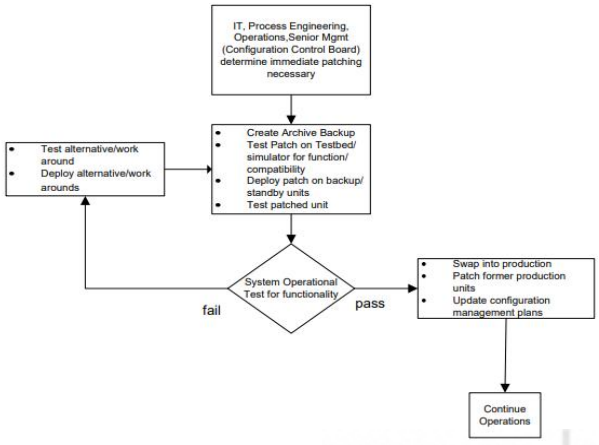

Fig. 4. An example of Flow chart of patching oerations

II. The Proposed Scheme

패치 관리 단계는 다음과 같이 5개의 개별 단계로 나눌 수 있다. 패치 관리 단계는 계속 반복되는 과정이다. 새로운 패치가 없는 경우에도 아래의 5단계를 반복하는 패치 관리 단계는 계속 진행되어야 한다. 그림 5는 본 연구에서 제안한 패치 관리의 전체적인 단계를 보여준다.

진단(Assessment) 단계는 조직 환경에 대한 상태를 파악하고, 직면할 수 있는 보안 위협 및 취약성, 조직이 새로운 소프트웨어 업데이트에 대응할 준비 여부를 진단하는 단계이다.

식별(Identification) 단계는 새로운 패치 업데이트를 안정적인 방식으로 검색하고 해당 업데이트가 조직 환경과 관련이 있는지 확인하고 패치 우선순위와 패치 배포, 설치 및 적용을 위한 시간을 설정하는 패치 중요도 순위를 생성한다.

Fig. 5. The Phase of Patch Management

평가(Evaluattion) 단계는 패치를 테스트하여 정보 시스템 구성요소에 대한 영향을 확인하고, 구성관리(Configuration) 단계는 패치 소유자를 식별하고, 보안 패치를 조직에 배포하는 체계적인 방법을 정의한다. 소프트웨어 업데이트 배포에 대한 실행 및 중단 결정, 업데이트 배포에 필요한 사항 결정, 실제와 유사한 환경에서 소프트웨어 업데이트를 테스트하여 업데이트가 비즈니스 긴급 시스템 및 애플리케이션을 손상 여부를 확인하여야 한다.

배포(Deployment) 단계는 배포 SLA(서비스 수준 계약)의 모든 요구 사항을 충족하는 방식으로 승인된 소프트웨어 업데이트를 업무 환경에 배포하는 단계로 배포 준비, 대상 컴퓨터에 배치 배포, 구현 후 검토 과정을 포함하여야 한다.

1. Assessment Phase

진단 단계는 조직 환경에 대한 상태를 파악하고, 직면할 수 있는 보안 위협 및 취약성, 조직이 새로운 소프트웨어

업데이트에 대응할 준비 여부를 진단하여야 한다. 새 패치가 배포되기 전에 잠재적인 배포를 위해 환경을 준비해야 한다. 환경에 대한 정보를 수집하거나 환경을 진단하면 패치를 성공적으로 배포하는 데 필요한 지식을 얻을 수 있다.

1.1 Inventory Management

조직은 자산 목록에 포함될 구성요소를 추적하고 보고하는 데 필요한 세분화 수준(예: 주변 장치를 개별적으로 식별하거나 워크스테이션의 일부로 포함)을 정의해야 한다. 효과적인 정보시스템 구성요소는 다음과 같은 정보를 수집 정리해야 한다.

- 고유 식별자 및/또는 일련번호
- 구성요소가 일부인 정보 시스템
- 구성요소 유형(예: 서버, 데스크탑, 애플리케이션),
- 제조사/모델 정보
- 운영 체제 유형 및 버전/서비스 팩 수준(적절한 공통 플랫폼 열거 이름 사용)
- 가상 머신의 존재
- 애플리케이션 소프트웨어 버전/라이센스 정보(적절한 공통 플랫폼 열거 이름 사용)
- 물리적 위치(예: 건물/방 번호)
- 논리적 위치(예: IP 주소)
- MAC(미디어 액세스 제어) 주소
- 소유자
- 운영 상태
- 기본 및 보조 관리자
- 기본 사용자(해당되는 경우)

1.2 Baselining Management

기준선 설정은 시스템 구축 및 배포를 위한 회사 표준으로 설정된 제품 또는 시스템에 대한 구성 집합이다. 기준선 아래로 식별된 모든 컴퓨터를 규정 준수 상태로 만들어야 한다. 클래스 기준을 초과하는 컴퓨터는 무단 변경 발생 여부를 확인해야 한다. 일부 시스템에서 기준선에 제외되는 특별한 상황에 대해 패치 관리가 이루어지도록 하는 기준을 설정해야 한다. 다양한 검색 기술을 사용하여 기준을 설정하고 조직의 컴퓨터가 설정된 기준 준수 여부를 결정하여야 한다. 많은 경우 오탐을 피하기 위하여 기준선의 일부로 여러 도구 집합을 사용하여야 한다.

보유하고 있는 컴퓨팅 자산이 무엇인지, 자산을 보호하는 방법과 소프트웨어 배포 아키텍처가 패치 관리를 지원할 수 있는지 확인하는 방법을 항상 알아야 한다. 지속적인 진단을 하기 위해 기존 컴퓨팅 자산 목록, 보안 위협 및 취약성 평가, 새 소프트웨어 업데이트에 대한 최상의 정보 출처 결정, 기존 소프트웨어 배포 인프라 평가, 운영 효율성 진단 작업을 수행하여야 한다. 다음 그림은 자산목록(Inventory) 관리의 예를 보여준다[25].

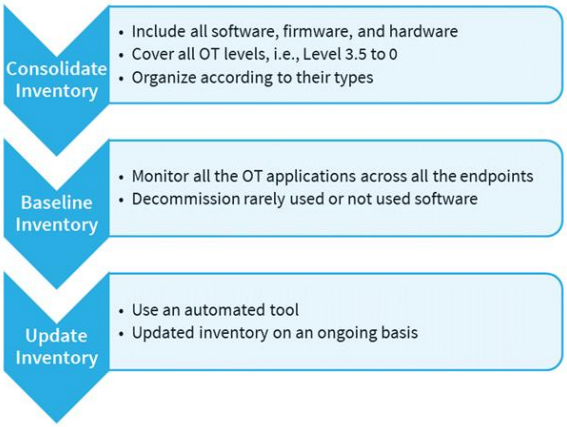

Fig. 6. An example of Inventory Management

2. Identification Phase

식별 단계는 새로운 소프트웨어 패치를 안정적인 방식으로 검색하고 해당 업데이트가 조직 환경과 관련이 있는지 확인하고 업데이트에 일반 프로세스 또는 긴급 배포 필요 여부를 결정하여야 한다. 새로운 패치의 우선순위와 패치 배포, 설치 및 적용을 위한 시간 프레임을 설정하는 패치 중요도 순위를 생성한다.

새 패치가 출시되면 패치를 사용할 수 있는 시점을 파악하여야 한다. 새 패치가 릴리스 될 때 이메일 서비스 및 기타 알림 방법 등과 같이 패치 대상 소프트웨어의 알림 방식 외의 알림 방식을 사용해야 한다.

2.1 Patch Time and Priority

패치 시점, 우선순위 지정 및 테스트는 기업 패치 관리에서 서로 연관되어 있다. 따라서 세 가지 특성을 잘 반영하는 패치 관리 지침이 필요하다. 시스템이 관련 소프트웨어 결함에 취약한 시간을 최소화하기 위해 모든 새 패치를 즉시 배포한다. 그러나 실제로는 기업의 리소스가 제한되어 있어 불가능한 경우 다른 패치보다 먼저 설치해야 하는 패치의 우선순위를 정해야 한다.

제품 공급 업체가 제품용 패치를 번들로 제공하여 한 번에 하나씩 수십 개의 패치를 릴리스하여 며칠마다 테스트하고 패치를 배포해야 하는 대신, 공급업체는 분기에 한 번씩 단일 번들로 패치를 배포할 수 있다. 이를 통해 조직은 테스트를 한 번 수행하고 패치를 한 번 배포할 수 있다. 이는 모든 패치를 개별적으로 테스트하고 배포하는 것보

다 훨씬 효율적이다. 또한 패치의 우선순위를 지정할 필요가 줄어든다. 조직은 포함된 각 패치의 우선순위를 별도로 지정하는 대신 번들의 우선순위를 지정하기만 하면 된다.

패치를 번들로 제공하는 공급업체는 패치를 적용하지 않은 취약점이 적극적으로 악용되는 경우를 제외하고는 월별 또는 분기별로 배포한다. 패치 번들링의 단점으로 취약점이 발견된 후 패치가 공개될 때까지의 시간이 늘어난다. 공격자가 패치가 배포되기 전에 동일한 취약점을 발견하면 의도적으로 패치 릴리스를 지연하기 때문에 공격자가 취약점을 악용할 수 있는 기회가 더 길어질 수 있다.

패치 배포는 공격자가 해당 취약성을 악용하는 데 필요한 정보(예: 패치의 취약성을 리버스 엔지니어링)를 제공할 수 있다. 즉, 새로 배포된 패치를 손상을 피하기 위하여 즉시 적용해야 할 수도 있다. 그러나 취약성이 아직 악용되지 않는 경우 조직은 먼저 철저한 테스트를 수행하지 않고 패치를 적용하지 않을 경우의 보안 위험과 패치 적용의 운영 위험을 신중하게 비교해야 한다. 스냅샷 기능이 활성화된 가상 호스트와 같은 일부 운영 환경에서는 조직이 사용성 또는 기능 문제를 일으킬 경우 패치를 복구할 준비가 되어 있는 한 테스트 없이 패치하는 것이 더 나을 수 있다.

타이밍과 관련된 또 다른 근본적인 문제는 패치를 적용하려면 변경 사항을 강제로 구현해야 할 수도 있다. 이를 위해서는 패치가 적용된 애플리케이션 또는 서비스를 다시 시작하거나, 운영 체제를 재부팅하거나 호스트 상태를 변경해야 할 수 있다(이는 전체 디스크 암호화(FDE) 소프트웨어 사용과 같이 호스트가 부팅 전에 인증을 요구할 때 문제가 될 수 있다. 부팅 전에 인증이 필요한 FDE 소프트웨어 또는 기타 기술을 사용하는 조직은 이러한 기술이 패치 설치에 미칠 수 있는 영향을 신중하게 고려해야 한다). 궁극적으로 중요한 것은 패치가 설치된 시점이 아니라 패치가 실제로 적용되는 시점이다.

패치의 우선순위를 정하는 것은 타이밍과 밀접한 관련이 있지만 다른 고려사항도 있다. 이는 취약한 시스템의 상대적 중요성(예: 서버 대 클라이언트)과 각 취약성의 상대적 심각도(예: CVSS(Common Vulnerability Scoring System)와 같은 취약성 심각도 측정항목)에 따라 달라질 수 있다. 또 다른 고려사항은 패치가 서로에 대해 가질 수 있는 종속성으로 하나의 패치를 설치하려면 먼저 다른 패치를 설치해야 할 수 있으며, 경우에 따라 패치를 순차적으로 적용하려면 애플리케이션을 다시 시작하거나 호스트를 여러 번 재부팅해야 한다.

대역외 또는 긴급 패치의 경우 대역 외 또는 긴급 패치는 관련 위험 등급과 함께 소프트웨어 제조업체가 발행할 때 평가되어야 한다. 대부분의 경우 이는 National Vulnerability Database에서 유지 관리되는 CVE(Common Vulnerabilities and Exposures) 등급 또는 식별된 취약점에 대해 마이크로소프트에서 할당한 등급을 기반으로 한다. 다음 표는 긴급 패치 처리 사례를 보여준다[26].

Table 1. Out of Band/Emergency Patching

CVE Rating	Micro Rating	Patching Window
High	Critical	< 7 Days
Medium	Important	< 15 Days
Low	Moderate/Low	< 30 Days

패치 관리는 패치 주소의 취약점 심각도에 따라 우선순위가 지정되어야 한다. 대부분의 경우 심각도 등급은 CVSS 및 CISA 지침을 기반으로 한다. 표 2의 설명과 같이 CVSS 점수 7~10은 영향이 큰 취약점으로 간주되고 4~6.9는 중간 수준의 취약점으로 간주되며 0~3.9는 낮은 영향의 취약점으로 간주 된다[27]. 영향도에 따라 패치 시작 및 패치 완료 시점을 고려해야 한다.

Table 2. The example of CVSS

Impact/ Severity	CVSS Score	Patch Initiated	Patch Completed
Critcal	10	Within 24 hours of patch release	Within 1 week of patch release
High	7~9.9	Within 24~72 hours of patch detected	Within 2 week of patch detection
Medium	4~6.9	Within 1 week of patch detected	Within 1 month of patch detection
Low	<4.0	Within 1 month of patch detected	Within 365 days during normal maintenance cycles

2.2 Inventory Risk Management

대부분의 패치 작업은 일상적인 정기 업데이트로 간주되어 예정대로 배포되며 일상적인 패치는 때때로 작업 중단 가능성(예:재부팅), 직원이 패치 배포를 연기하거나 무시하는 제한된 긴급성과 같은 이유로 취약성 위험이 나타난다.

위기 시 배포되는 비상 패치는 사고 대응 프로토콜에 매우 중요하다. 효율적인 긴급 패치는 취약한 장치 또는 시스템의 악용을 최소화 한다.

긴급 패치를 사용할 수 없는 위기에서 긴급 해결 방법은 취약성 악용을 일시적으로 완화하는 데 도움이 된다. 해결 방법은 특정 패치 시나리오에 따라 다르다. 해결 방법 배포 후 롤백이 필요할 수 있다. 일상적인 패치에는 해결 방

법이 필요한 문제가 있는 경우가 있다.

특히 자산에 대한 지원 업데이트를 제공하지 않는 공급업체(예: 수명 종료), 지원 커뮤니티가 덜 활성화된 오픈소스 도구, 중요한 업무수행 기능을 위한 자산의 중단 없는 사용 등과 같이 적시에 일상적인 패치를 사용할 수 없는 경우 노출 위험을 완화하기 위해 쉽게 패치할 수 없는 시스템을 격리해야 할 수 있다.

2.3 Patch Identification Considerations

조직 패치 식별관리의 추가 고려사항으로 다음과 같은 일반적으로 패치를 적용하기 위한 여러 메커니즘이 있다.

- 소프트웨어가 자동으로 업데이트될 수 있다.
- 중앙 집중식 OS 관리 도구가 패치를 시작할 수 있다.
- 타사 패치 관리 응용 프로그램이 패치를 시작할 수 있다.
- 네트워크 액세스 제어, 상태 확인 기술 및 유사한 기술이 패치를 시작할 수 있다.
- 사용자는 소프트웨어가 스스로 업데이트하도록 수동으로 지시할 수 있다.
- 사용자가 패치 또는 소프트웨어의 새 버전을 수동으로 설치할 수 있다.

여러 가지 방법으로 패치를 적용하면 충돌이 발생할 수 있다. 여러 방법이 각각 동일한 소프트웨어에 패치를 시도할 수 있으며, 이는 조직이 해당 패치 문제, 테스트 지연 등으로 인해 특정 패치 적용을 원하지 않을 때 특히 문제가 된다. 각 도구나 관리자는 다른 도구나 관리자가 이미 특정 패치를 처리하고 있다고 가정할 수 있기 때문에 여러 방법으로 인해 패치가 지연되거나 누락 될 수도 있다. 조직은 패치를 적용할 수 있는 모든 방법을 식별하고 패치 적용 방법 간의 충돌을 해결하기 위한 조치를 취해야 한다.

패치 관리 구성과 관련된 어려운 문제는 사용자가 패치 관리 단계를 무시하거나 우회할 수 있다. 사용자가 설정 변경(예: 직접 업데이트 활성화, 패치 관리 소프트웨어 비활성화), 이전 버전의 소프트웨어 설치, 패치 제거와 같이 호스트 소프트웨어를 변경할 수 있는 경우 패치 관리 프로세스의 무결성이 손상될 수 있다. 이러한 문제를 해결하기 위해 조직은 사용자가 엔터프라이즈 패치 관리 기술을 비활성화하거나 부정적인 영향을 미칠 수 없도록 해야 하며 조직은 엔터프라이즈 패치 관리 기술을 지속적으로 모니터링하여 발생하는 문제를 식별해야 한다.

3. Evaluation and Plan Phase

평가 및 계획 관리 과정에서는 패치를 테스트하여 정보시스템 구성요소에 대한 영향을 확인하고, 패치 소유자를 식별하고, 보안 패치를 조직에 배포하는 체계적인 방법을 정의한다. 평가 및 계획 관리 단계의 목표는 다음과 같다.

- 소프트웨어 업데이트 배포에 대한 실행/중단 결정
- 업데이트 배포에 필요한 사항 결정
- 사용할 때와 유사한 환경에서 소프트웨어 업데이트를 테스트하여 업데이트가 비즈니스 크리티컬 시스템 및 애플리케이션을 손상시키지 않는지 확인
- 배포 승인 받기
- 테스트 업데이트를 배포팀에게 전달

각 조직의 분류 및 관련 배포 시간 프레임은 조직의 요구 사항에 따라 다르기 때문에 모든 사람이 패치에 대해 요구 사항을 이해할 수 있도록 패치 분류 시스템을 개발해야 한다. 패치에 부여하는 분류 수준에 따라 다른 테스트 절차를 개발해야 한다.

사용 환경과 매우 유사한 환경에서 테스트해야 한다. 가상 머신을 사용하여 패치를 테스트하는 데 필요한 테스트 환경을 제공해야 한다. 이 설정은 대부분의 경우 패치가 설치되어 회사의 설치된 응용 프로그램 및 서비스 집합과 함께 작동하는지 확인하기 위한 적절한 테스트 환경 제공할 수 있지만 이 설정에 주의해야 한다.

가상 머신을 사용하여 테스트를 수행하는 경우 또한 몇 가지 사용자 시스템에서 테스트하여 가상 머신 결과의 정확성을 확인하여야 한다. 따라서 랩 환경(실제 또는 가상)에서 패키지를 테스트한 다음 사용자 환경에서 파일럿 테스트하여 비즈니스 애플리케이션을 손상시키지 않는지 확인해야 한다.

테스트하는 동안 식별 단계에서 전달된 정보를 사용하여 설치 및 제거에 대한 모든 패치 옵션을 완전히 테스트해야 한다. 패치를 배포하는 방법(예: 워크스테이션에 로그온해야 함, 패치를 자동으로 배포해야 함, 패치를 성공적으로 설치하려면 재부팅 필요) 및 패치를 제거하는 방법(롤백)에 대한 계획을 만들어야 한다.

4. Configuration Management Phase

패치를 테스트하고 배포할 준비가 되면 시스템에 대한 제안된 변경 사항과 테스트 결과를 문서화하고 변경 관리자가 승인해야 한다. 시스템 소유자는 재해 복구 단계가 필요한 경우 대기할 준비가 되어 있어야 한다.

자동화된 시스템 모니터링이 활성화된 경우 모니터링되는 시스템이 오프라인 상태가 되어 경고가 발생하면 해당 담당자에게 알려야 한다. 배포 중에 부작용이 발생하는 경우 발생한 상황과 시스템에 대한 세부정보를 문서화하고 향후 테스트에 통합해야 한다.

4.1 Configuration Management Guide

패치를 통해 변경하려는 모든 변경 사항을 문서화 해야 한다. 구성관리 프로세스를 통해 정보시스템 구성요소가 호환성, 기능 및 성능을 최적화하도록 구성해야 한다.

- 정보시스템의 일관성을 구축하고 유지해야 한다.
- 시스템 구성요소가 요구 사항을 준수하고 제품 수명 주기를 지원하기에 충분히 상세하게 식별 및 문서화 되었는지 확인해야 한다.
- 정확한 시스템 구성 정보를 보장하여 안전한 운영 및 유지 보수를 지원해야 한다.
- 기능 향상, 결함 수정, 성능, 안정성 또는 유지 관리 개선, 시스템 수명 연장, 비용, 위험 및 책임을 줄이기 위한 시스템 제어를 제공해야 한다.

4.2 Configuration Component Search

구성요소 검색 방법은 세 가지 방법이 있다. 에이전트 기반 패치 관리 기술은 패치를 적용할 각 호스트에서 에이전트가 실행되고 패치 프로세스를 관리하고 에이전트들을 조정하는 하나 이상의 서버가 필요하다. 각 에이전트는 호스트에 설치된 취약한 소프트웨어를 결정하고, 패치 관리 서버와 통신하고, 호스트에 사용할 수 있는 새 패치를 결정하고, 해당 패치를 설치하고, 패치를 적용하는 데 필요한 상태를 변경한다(예: 애플리케이션 다시 시작, OS 재부팅). 각 에이전트는 이러한 작업을 수행할 수 있도록 관리자 권한으로 실행된다. 패치 관리 서버는 패치를 얻을 수 있는 위치와 필요한 상태 변경을 포함하여 취약한 소프트웨어 및 사용 가능한 패치에 대한 정보를 에이전트에 제공하는 역할을 한다.

두 번째 방법으로 에이전트 없는 스캐닝 패치 관리 기술에는 패치할 각 호스트의 네트워크 스캐닝을 수행하고 각 호스트에 필요한 패치를 결정하는 하나 이상의 서버가 필요하다. 서버에 각 호스트에 대한 관리 권한이 있어야 정확한 검색 결과를 반환하고 호스트에서 패치를 설치하고 상태 변경(응용 프로그램 다시 시작, OS 재부팅 등)을 구현할 수 있는 기능을 갖출 수 있다.

수동 네트워크 모니터링 기술은 로컬 네트워크 트래픽을 모니터링하여 패치가 필요한 애플리케이션(경우에 따라 운영 체제)을 식별한다. 수동 네트워크 모니터링의 장점으로 다른 패치 관리 솔루션(에이전트 기반, 에이전트 없는 검색)에서 유지 관리하지 않는 호스트를 식별하는 데 효과적일 수 있다. 모니터링 할 호스트에 대한 권한이 필요하지 않으므로 조직에서 제어하지 않는 호스트(비관리 시스템, 방문자 시스템, 계약자 시스템 등)의 패치 상태를 모니터링하는 데 사용할 수 있다. 수동 네트워크 모니터링의 주요 단점은 네트워크 트래픽(암호화되지 않은 것으로 가정)을 기반으로 버전을 식별할 수 있는 소프트웨어에서만 작동하고 로컬 네트워크의 호스트에서만 작동한다.

5. Deployment Phase

배포 단계의 목표는 배포 SLA(서비스 수준 계약)의 모든 요구 사항을 충족하는 방식으로 승인된 소프트웨어 업데이트를 사용자 환경에 성공적으로 배포해야 한다. 배포 단계에서는 배포 준비, 대상 컴퓨터에 배치 배포, 구현 후 검토 과정을 포함하여야 한다.

5.1 Preperation of Deployment

배포 준비를 위해 회사의 다양한 영역에서 대표하는 그룹으로 정의된 통신 채널의 모든 사람에게 패치가 배포를 보류 중임을 알려야 한다. 패치를 배포할 준비가 되었음을 알리고 패치 배포 일정을 전달해야 한다.

커뮤니케이션에서 성공적인 배포를 위해 패치 배포에 대한 충분한 정보를 제공해야 하고 패치가 컴퓨터에 언제 도착할 것인지에 대한 명확한 지침 제공해야 한다. 통신이 중계되고 필요한 그룹으로 확산 될 수 있는 충분한 시간이 주어지면 패치를 준비하여 전달할 준비가 되도록 해야 한다.

5.2 Deployment Management

패치 관리 도구를 배포하면 아래와 같이 조직에 추가적인 보안 위험이 발생할 수 있다. 이러한 위험을 파악하여 대책을 고려해야 한다.

- 패치가 변경되었을 수 있다(실수로 또는 의도적으로).
- 자격 증명이 오용될 수 있다.
- 솔루션 구성 요소(에이전트 포함)의 취약점이 악용될 수 있다.
- 기업은 취약점을 식별하기 위해 도구 통신을 모니터링 할 수 있다(특히 호스트가 외부 네트워크에 있는 경우).

조직은 단계적 접근 방식을 사용하여 엔터프라이즈 패치 관리 도구를 배포해야 한다. 조직은 표준화된 데스크탑

시스템과 유사하게 구성된 서버의 단일 플랫폼 서버 팜에 먼저 패치 관리 도구를 배포한다. 다음으로 조직은 다중 플랫폼 환경, 비표준 데스크탑 시스템, 구식 컴퓨터 및 비정상적인 구성의 컴퓨터를 통합하는 문제를 해결해야 한다. 자동 패치 도구가 지원하지 않는 운영 체제 및 응용 프로그램과 비정상적인 구성의 일부 컴퓨터에는 수동 방법을 사용해야 할 수 있다(임베디드 시스템, 산업 제어 시스템, 의료 기기 및 실험 시스템). 이러한 컴퓨터의 경우 수동 패치 프로세스에 대한 절차를 작성하고 구현해야 한다.

조직은 보안 요구 사항과 유용성 및 가용성 요구 사항 간의 균형을 맞춰야 한다. 즉, 패치를 적용해야 하는 필요성과 운영을 지원해야 하는 필요성 사이에서 균형을 유지해야 한다. 패치 설치 후 다른 응용 프로그램의 중단이 발생할 수 있고 애플리케이션의 다시 시작, OS 재부팅 및 기타 호스트 상태 변경을 강제하게 되면 중단되고 데이터 또는 서비스가 손실될 수 있다.

조직은 패치 솔루션이 저대역폭 또는 데이터 통신 네트워크에서 사용되는 모바일 호스트 및 기타 호스트에 대해 작동하는지 확인하기 위한 규정을 마련해야 한다.

5.3 Deploy of Patch

패치 배포 후 배포 진행 상황 실시간 모니터링 및 보고, 실패한 배포 처리를 수행해야 한다. 패치가 컴퓨터에 적용되었는지 여부, 컴퓨터에서 패치 설치 시도 여부, 설치 성공 여부에 대한 정보 포함하여야 한다.

배포에 실패하여 복구해야 하는 경우 패치 관리 프로세스의 일부로 배포를 중지하고 실패한 업데이트를 제거하고 다시 배포하는 계획이 있어야 한다.

5.4 Implemetation and Check

배포 단계의 마지막 단계로 배포 성공 여부, 배포 성공 여부, 미래에 더 나은 성공을 보장하기 위해 프로세스를 조정 여부, 프로세스에서 특정 작업을 수행하는 개인의 성과, SLA를 조정 여부 등을 결정해야 한다. 또한 다음 진단 단계를 준비하기 위한 기준선을 만들거나 업데이트 해야 한다. 조직은 패치 관리가 리소스 과부하 상황을 피할 수 있도록 해야 한다. 작은 패치 설치로 인한 보안 구성 설정 변경과 같은 부작용을 감지할 수 있어야 한다. 조직은 패치 관리가 리소스 과부하 상황을 피할 수 있도록 해야 한다. 조직은 패치 관리 시스템과 독립적인 취약성 스캐너와 같은 다른 설치 확인 방법을 사용해서 검증해야 한다.

III. Conclusions

소프트웨어 보안 취약점으로 인한 보안 사고가 증가함에 따라서 패치 관리의 중요성이 크게 부각되고 있다. 기업, 대학 등 다양한 형태의 조직에서 일관되지 않은 방법으로 패치를 관리하여 보안 문제 발생, 시스템 불안정, 패치로 인한 정보 유출 및 작업지연 등의 문제가 발생하고 있다. 취약점 발견에서 악용까지의 시간이 점점 짧아지고 있기 때문에 IT 관리자는 시스템을 신속하게 패치하고 최신 업데이트를 따라야 하는 부담을 안고 있다. 그러나 적절한 패치 관리 정책을 정의하면 시간과 비용을 절약하고 보안 문제를 크게 줄일 수 있다. 또한 자동 패치 관리 시스템이 주기적으로 패치를 설치하므로 패치 관리의 수동 구성 요소가 제거된다. 또한 소프트웨어 결함이 발견되는 즉시 감지하고 신속하게 패치할 수 있도록 한다.

분명히 본 연구에서 제시한 패치 관리만으로는 모든 취약성 관련 문제를 해결할 수는 없지만 전반적인 취약성 관리 프로그램에서 근본적인 보호 메커니즘으로 작용할 것이다. 또한 본 연구는 소프트웨어 기능의 수정, 보안 취약점에 대한 해결 등으로 조직 내의 구성원들이 사용하고 있는 소프트웨어를 새로운 버전으로 갱신하여 설치하는 과정에서 조직의 패치 관리 정책을 설정하는데 효율적이고 안전하게 활용될 것이다. 향후 시스템을 보유한 기관에 따라 패치 절차를 구체적으로 세분화할 필요가 있고, 각 소기관에 적합한 간결한 인벤토리 관리 및 패치 절차를 연구할 예정이다.

REFERENCES

[1] Nesara Dissanayake, “Software Security Patch Management - A Systematic Literature Review of Challenges, Approaches, Tools and Practices“, Journal of LATEX, Aug. 23, 2021.

[2] A. Security, 2020 cyber threatscape report, https://www.accenture.com/_acnmedia/PDF-136/Accenture-2020-Cyber-Threatscape-Full-Report.pdf, accessed: 2020-10-30.

[3] B. Thomas, New windows vulnerabilities highlight patch management challenges, https://www.bitsight.com/blog/new-windows-vulnerabilities-highlight-patch-management-challenges,accessed: 2020-10-30.

[4] D. o. P. M. . O. C. D. S. Scott Coleman, Cyber security review, https://www.cybersecurity-review.com/what-if-you-cant-patch/, accessed: 2020-10-30.

[5] R. Brandom, Former equifax ceo blames breach on a single person who failed to deploy patch, https://www.theverge.com/2017/10/3/1

6410806/equifax-ceo-blame-breach-patch-congress-testimony, accessed: 2020-08-24.

[6] M. Lee, Equifax data breach impacts 143 million americans, https://www.forbes.com/sites/leemathews/2017/09/07/equifax-data-breach-impacts-143-million-americans/#7aa9c117356f, accessed: 2020-08-24. 40

[7] L. H. Newman, Equifax ocially has no excuse, https://www.wired.com/story/equifax-breach-no-excuse/, accessed: 2020-08-24.

[8] N. P. Melissa Eddy, Cyber attack suspected in german womans death, https://www.nytimes.com/2020/09/18/world/europe/cyber-attack-germany-ransomeware-death.html?smid=tw-share, accessed:2020-09-21.

[9] M. Souppaya, K. Scarfone, Guide to enterprise patch management technologies, NIST Special Publication800 (2013) 40.

[10] F. Li, L. Rogers, A. Mathur, N. Malkin, M. Chetty, Keepers of the machines: Examining how system administrators manage software updates for multiple machines, in: Fifteenth Symposium on Usable Privacy and Security (fSOUPSg 2019), 2019.

[11] C. Tiefenau, M. H□aring, K. Krombholz, E. von Zezschwitz, Security, availability, and multiple information sources: Exploring update behavior of system administrators, in: Sixteenth Symposium on Usable Privacy and Security (fSOUPSg 2020), 2020, pp. 239{258.

[12] U. Gentile, L. Serio, Survey on international standards and best practices for patch management of complex industrial control systems: the critical infrastructure of particle accelerators case study, International Journal of Critical Computer-Based Systems 9 (1-2) (2019) 115{132.

[13] Goverment of Canada , https://www.canada.ca/en/government/system/digital-government/online-security-privacy/patch-management-guidance.html

[14] NIST Special Publication 800-40 Revision 3, “Guide to Enterprise Patch Management Technologies”.

[15] Jason Chan, “Essentials of Patch Management Policy and Practice, ” on Patch Management.org website, hosted by Shavlik Technologies, LLC.

[16] US-CERT homepage, http://www.kb.cert.org/vuls/

[17] Adam Murray, MEND homepage, https://www.whitesourcesoftware.com/resources/blog/patch-management-best-practices/

[18] G DATA tech. report, https://file.gdatasoftware.com/web/en/documents/techpaper/G_DATA_TechPaper_Patch_Management_Best_Practices_English.pdf1.3.pdf

[19] GTI Software homepage, https://www.gfi.com/patch-management/

[20] Northwestern Information Technology Tech. paper, https://www.it.northwestern.edu/policies/patches.html

[21] NIST Computer Security Resocure Center, http://csrc.nist.gov/groups/SMA/fisma/index.html

[22] NIST Infomation Technology Laboratory National Vulnerability Database, http://nvd.nist.gov/home.cfm

[23] Goverment of Canada , https://www.canada.ca/en/government/system/digital-government/online-security-privacy/patch-management-guidance.html

[24] Homeland Security, Recommended Practice for Patch Management of Control Systems , https://www.cisa.gov/uscert/sites/default/files/recommended_practices/RP_Patch_Management_S508C.pdf

[25] Global Cybersecirity Alliance, The Top 7 Operarional Technical Patch Management Best Practices, https://gca.isa.org/blog/the-top-7-operational-technology-patch-management-best-practices

[26] CCIT, Patching Guidelines, https://ccit.clemson.edu/cybersecurity/policy/patching-guidelines/

[27] Oklahoma Office of Management&Enterprise Services, https://oklahoma.gov/content/dam/ok/en/omes/documents/PatchManagementStandard.pdf

Authors

Chang-Hoon Kang received the B.S. and M.S. degrees in Dept. of Computational Statistics from Chungnam National University and the Ph.D. degrees in Dept. of Computer Engineering from Ajou University, Korea, in 1986, 1988 and 2006, respectively. Dr. Kang joined the faculty of the Department of Visual Broadcasting Media at Gangdong University, Gamgok, Korea, from 1994 to 2021. He is currently a Director of Solution Bank Co. Ltd. He is interested in computer system, compuer security and broadcasting media.