SW 업데이트 체계 보안 가이드라인

NSRI 국가보안기술연구소

1장 개요

1 목적

- SW 업데이트 체계는 최신 버전의 소프트웨어를 검색, 다운로드 및 설치하기 위한 자동화된 소프트웨어 관리체계를 의미하며, 운영체제 및 응용프로그램의 버그 패치, 성능 개선 등의 목적으로 마이크로소프트, 애플, 레드햇 사의 SW 제품을 포함한 대부분의 상용 및 공개 소프트웨어에서 사용된다.
- 그러나 SW 업데이트 체계에 대한 보안은 소프트웨어 개발자 및 사용자들에게 소프트웨어 자체의 보안에 비해 상대적으로 보안에 대한 인식이 부족한 상황이며, 사용자 수가 많은 SW 업데이트 체계의 취약점이 공격자에 의해 악용될 경우 대규모의 피해를 발생시킬 수 있다.
- 특히, 최근 발생한 '3.20 및 6.25 사이버공격(2013년)', 'SK컴즈 개인정보 유출 사태(2011년)', '농협 금융 전산망 마비 사태(2011년)', '7.7 DDos 대란(2009년)' 등과 같은 사이버 보안사고는 SW 업데이트 체계의 취약점을 악용하여 악성코드를 확산시켜서 공공 및 민간 기반시설에 대한 대규모 피해를 유발하였다.
- 이에 따라 본 보안가이드라인에서는 SW 업데이트 체계의 보안을 향상시키기 위해 필요한 주요 보안문제에 대해 기술한다.

2 적용대상

- 본 보안가이드라인은 SW 개발 업체에서 자동 업데이트 기능 개발 시 준수해야 할 보안 항목 및 주의사항을 따르지 않을 경우 발생할 수 있는 위험성, 해킹에 악용된 사례 및 문제해결 방안을 설명한다.

3 구성

- 2장에서는 SW 업데이트 체계의 정의 및 현황에 대하여 살펴본다.
- 3장에서는 SW 업데이트 체계의 보안위협과 보안위협 사례에 대해 기술한다.
- 4장에서는 SW 업데이트 체계의 보안위협에 대한 대응방안을 기술한다.
- 5장에서는 SW 업데이트 체계의 보안을 향상시키기 위해 SW 업데이트 체계 개발 시 점검해야 할 사항을 10가지로 분류하여 기술한다.

2장 SW 업데이트 체계 정의 및 현황

1 정의

1-1 SW 업데이트 체계 정의

- SW 업데이트는 이미 배포된 소프트웨어 제품의 기능 개선 또는 버그나 보안 취약점 등을 수정하기 위해 개발사가 내놓는 프로그램을 지칭한다. (위키백과)
- SW 업데이트 체계는 최신 버전의 소프트웨어를 검색, 다운로드, 설치하는 체계를 말한다.
- SW 업데이트 체계는 업데이트 프로그램을 전송하는 업데이트 서버, 실제 업데이트를 담당하는 업데이트 클라이언트 그리고, 업데이트 서버와 클라이언트 사이에 설치프로그램을 전송하는 통신 채널로 구성된다.

1-2 SW 업데이트 유형

- SW 업데이트는 사용자가 직접 업데이트 파일을 웹으로부터 다운로드 받아서 설치하는 수동 업데이트 방식과 자동으로 정해진 일정에 따라 업데이트 필요 유무를 검사하고 필요 시 다운로드 및 설치하는 자동 업데이트 방식으로 구분할 수 있다. 최근에는 빈번한 보안 취약점 및 버그 수정을 위해 자동업데이트가 일반적으로 사용되고 있다.
- SW 업데이트는 그 내용의 중요도에 따라 다음과 같이 구분할 수 있다.
  - 중요 업데이트는 보안 취약점에 대한 패치, 개인 정보 보호 및 버그 수정 등의 안정성 및 안전성과 관련된 필수 업데이트를 의미한다.
  - 선택 업데이트에는 컴퓨터 작업 환경을 향상시킬 수 있는 업데이트, 드라이버 또는 해당 제조사의 새로운 소프트웨어가 포함될 수 있으며, 사용자의 선택에 따라 설치되는 업데이트이다.

1-3 SW 업데이트 체계 동작구조

- (설정정보 다운로드) 사용자 PC에 설치된 SW에서 약속된 업데이트 서버에 접속해서 최신 버전에 대한 버전 번호와 업데이트를 위해서 필요한 파일들의 URL 목록을 다운로드
- (파일 다운로드) 사용자 PC에 설치된 SW의 버전이 최신 버전이 아니라면 서버로부터 획득한 URL 목록을 이용해서 업데이트에 필요한 파일들을 다운로드
- (파일 설치) 다운로드한 파일들을 사용자 PC에 설치 (파일 교체 및 실행)

※업데이트 서버와 사용자 PC간의 연결은 보통 HTTP 또는 HTTPS로 이루어짐

2 현황

2-1 전 세계 대부분의 SW는 자체적인 업데이트 기능 탑재

- 대부분의 SW 제품은 기능개선 및 보안패치를 자사 제품에 적용하기 위한 업데이트 체계를 갖추고 있다.
- SW 업데이트 체계는 국제적인 표준이나 가이드라인이 없어 각 SW 개발사마다 서로 다른 방식으로 구축·운영하고 있는 실정이다.
- PC 1대당 평균 24개의 제작사로부터 74개의 프로그램이 설치되어 있으며, 사용자는 74개 프로그램을 패치하기 위해 24개의 각기 다른 업데이트 체계를 관리해야 한다.
  ※출처 : 영국 Secunia PSI Report Q1, 2013.

제품		업데이트 체계
Microsoft	Windows 제품군 Office 제품군 기타	WSUS(Windows Server Update Services)
Linux	Fedora, Redhat, Ubuntu 등	YUM(Yellowdog Updater, Modified) YaST(Yet another Setup Tool) APT(Advanced Packaging Tool)
Adobe	Reader, AIR 등	Adobe Updater
Mozilla	Firefox 등	Firefox Update
Google	Chrome, Earth 등	Google Update (Omaha)
	Android (스마트폰)	Google Play
Apple	Mac OS X, iTunes 등	Apple Software Update
	iPhone (스마트폰), iPad 등	AppStore

- 국산 SW에서도 자동업데이트는 필수적인 요소가 되었으며, V3나 알약 같은 백신은 물론이고 한글, 곰플레이어 등 대부분의 사용자 응용프로그램에도 포함되어 있다.
- 사용자는 컴퓨터를 켜는 순간부터, 문서 작성, 메신저 대화, 음악 감상, 파일 압축, 파일 전송, 주식거래, 스마트폰 관리, 내비게이션 관리 등의 순간마다 사용자가 인지하거나 인지하지 않은 상태에서 업데이트를 수행하고 있다.

분야	세부분야	국산 소프트웨어 예
응용 소프트웨어	문서편집/변환	한글, 알마인드, 훈민정음, 나모 웹에디터
	문서뷰어	한컴뷰어, 별PDF 리더, 훈민정음2000 뷰어
	그래픽 프로그램	알씨, 꿀뷰, 알캡처
	음악 재생 프로그램	곰오디오, 벅스플레이어, 멜론
	동영상	곰플레이어, 팟플레이어, KMPlayer
	압축프로그램	알집, 반디집, 빵집
	브라우저 툴바	알툴바, 네이버 툴바
	메신저	네이트온, 마이피플, 라인
	게임	한게임, 피망
	보안	V3, Virobot, 알약, 네이버백신
	스마트폰 관리	삼성 Kies, LG Mobile Support Tool
	주식거래	키움증권, 대신증권 등 HTS
	네비게이션	아틀란, 지니
	P2P, 웹하드	미래창조과학부 등록 웹하드 전용 SW

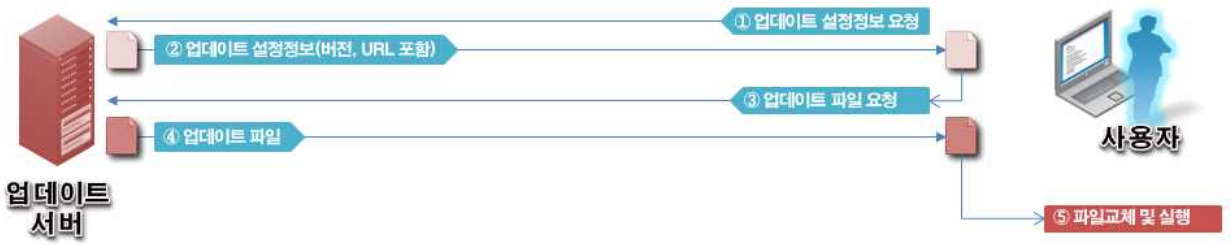
- 특히 국내에서는 금융을 위한 ActiveX Control의 사용이 많으며, ActiveX Control은 사이트 방문 시 최신 버전으로 유지하기 위한 자동 업데이트를 수행한다. ActiveX Control의 자동 업데이트 기능을 포함한다면 국내 사용자들은 외국에 비해 더 많은 업데이트 프로그램을 사용하고 있는 것이 현실이다.

3장 SW 업데이트 체계 보안위협

1 보안위협 유형

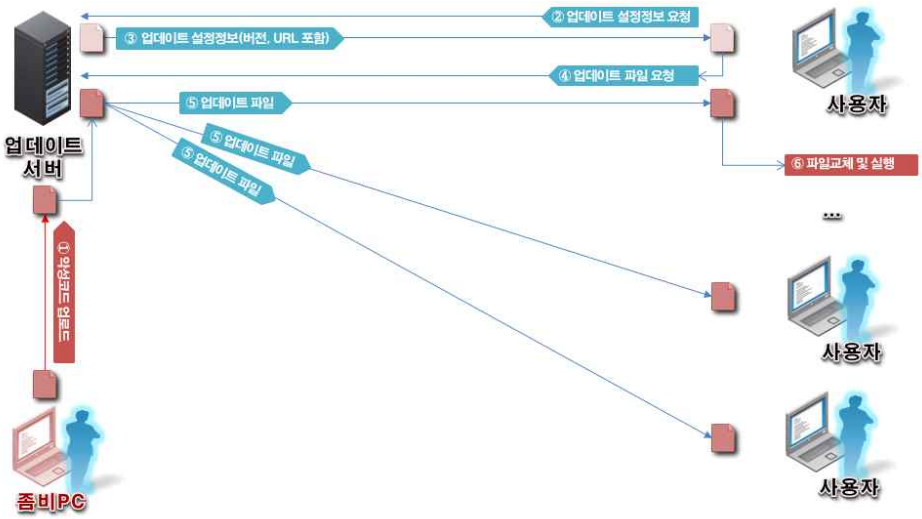
1-1 (서버) 서버장악 (사례: '09년 7.7 DDoS, '11년 3.4 DDoS)

- 위협개요: 별도의 해킹을 통해서 업데이트 서버에 침투한 후 업데이트에 사용되는 설정정보 또는 업데이트 파일 위·변조
- 파급효과: 해당 SW 사용자 단말 전체에 악성코드 유포 가능
- 발생원인: 외부에서 업데이트 서버로 원격 침투



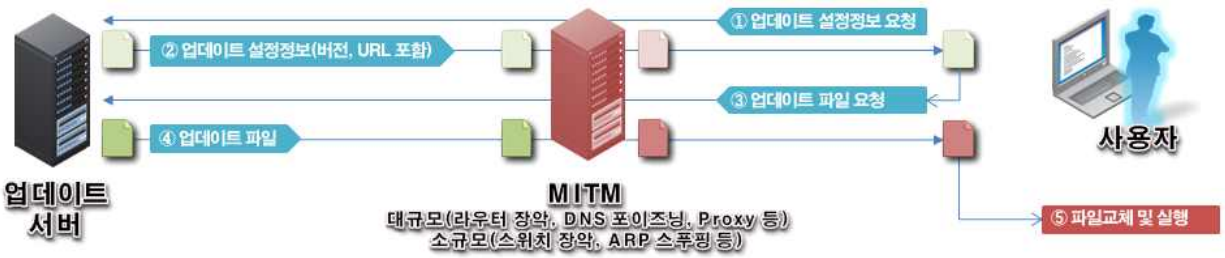
1-2 (서버) 악성코드 업로드 (사례: '13년 3·20 사태)

- 위협개요: 업데이트 서버를 직접 해킹해서 장악하지 않고도 업데이트 서버에 악성코드를 업로드해서 다른 사용자 단말로 악성코드를 유포 (서버장악과 동일한 효과)
- 파급효과: 해당 SW 사용자 단말 전체에 악성코드 유포 가능
- 발생원인: 업데이트 서버에서 파일 업로드에 필요한 권한(사용자 또는 기기) 미확인



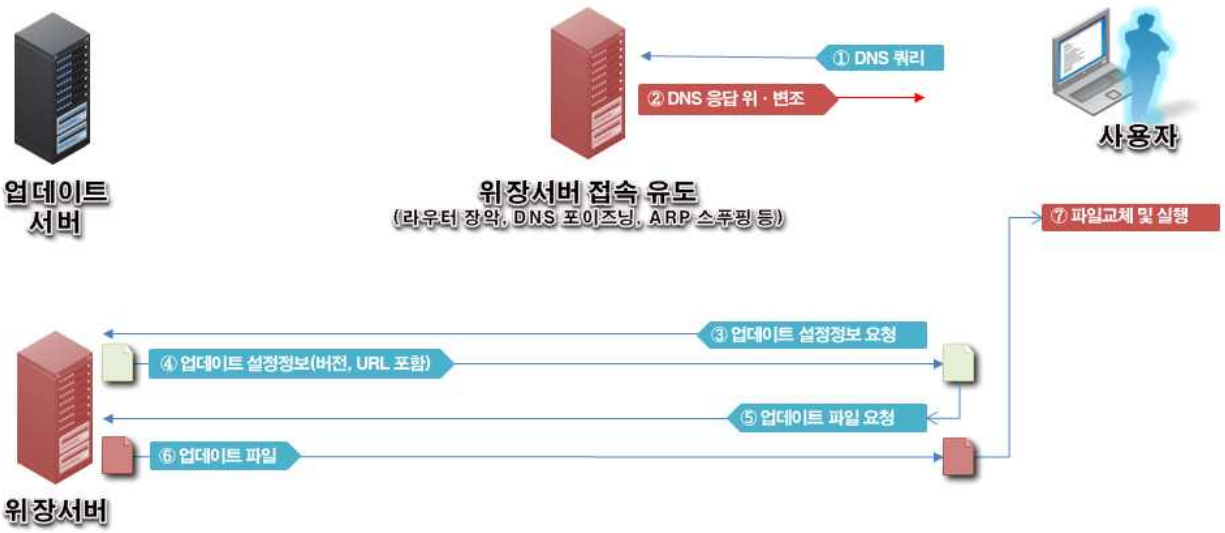
1-3 (채널) MITM(Man-In-The-Middle)

- 위협개요: 업데이트 서버와 사용자 단말 간의 통신경로 사이에서 업데이트 설정정보 또는 업데이트 파일 위·변조
- 파급효과: 통신경로 상의 위치에 따라 영향 받는 사용자 단말에 악성코드 유포 가능
- 발생원인: 클라이언트가 업데이트 서버에서 전송한 설정정보 및 파일의 위·변조 여부 미확인



1-4 (클라이언트) 위장서버 접속 (사례: '11년 SK컴즈 개인정보유출)

- 위협개요: DNS 응답 위·변조 등의 방법으로 업데이트 파일 다운로드 전(前) 단계에서 사용자 단말이 정상적인 업데이트 서버가 아닌 악의적인 위장서버로 접속하도록 유도하여 위장서버에서 악성코드를 사용자 단말에 전송
- 파급효과: 위장서버로 유도하는 지점의 통신경로 상의 위치에 따라 영향 받는 사용자 단말에 악성코드 유포 가능
- 발생원인: 클라이언트가 통신채널 상대방의 정상적인 업데이트 서버 여부 미확인



2 보안위협 사례

2-1 국내 보안 취약점 사례

- 인터넷에서 다운로드 받아 사용할 수 있는 국내 주요 SW 20종에서 SW 업데이트 체계 취약점 발견('13.6.12. 관련업체 통보)
- 발견된 취약점은 최악의 경우 해당 SW 사용자 PC 전체에 악성코드를 유포·확산시킬 수 있는 치명적인 수준인 것으로 판단됨
- 특히, A사의 경우는 지난 '11년 발생한 업데이트 관련 사고 이후, 보안강화 대책을 적용한 것으로 알려졌으나, 여전히 취약점이 발견되었음

SW	보안 위협		사용자 (단위:명)	취약점 설명
	(채널) MITM	(클라이언트) 위장서버 접속		
A사	○	○	천만 이상	■ 병행제품 설치 시 위장서버로 접속 악성코드 다운로드 및 실행 가능 ■ 임의의 인증서를 제시하는 SSL 사용 웹서버에 접속하는 경우 인증서 유효성 및 발급주체 검증 미실시로 악성코드 다운로드 및 실행 가능
B사	○	○	추정 불가	■ 서명검증 미실시
C사	○	○	천만 이상	■ 서명검증 미실시
D사	○	○	추정 불가	■ 서명검증 미실시
E사	○	○	백만 이상	■ 제휴제품 설치 시 서명검증 미실시

2-2 국외 보안 취약점 사례

- 2006년부터 지속적으로 SW 업데이트 체계 취약점이 발견되고 있음
- 특히, Apple, Oracle, McAfee, Mozilla 등의 대표적인 글로벌 SW 개발사들의 제품에서도 취약점이 발견되고 있음
- 더욱이, 미국 보안회사인 McAfee의 바이러스 백신의 업데이트 체계에서도 취약점이 발생되었던 것으로 볼 때 보안회사 제품도 악성코드 침투경로로 악용될 수 있음

SW		보안 위협 (추정)		취약점 설명
		(채널) MITM	(클라이언트) 위장서버 접속	
2012년	Java (Oracle)	추정 불가	추정 불가	▪ 입력값 검사 미실시로 외부의 임의의 바이너리 파일 다운로드 및 실행 가능
	Catalyst Control Center (AMD)	추정 불가	추정 불가	▪ 자동 업데이트 모듈의 취약점으로 AMD Catalyst 13.1 버전부터 드라이버 자동 업데이트 기능 제거키로 함
2010년	iTunes (Apple)	○	○	▪ Infobyte의 Francisco Amato는 WinZip, iTunes, Winamp 등 63종의 자동 업데이트 모듈에서 서명 검사를 수행하지 않는 취약점을 이용하여 클라이언트에 악성 프로그램을 설치할 수 있는 도구인 Evilgrade를 개발 ▪ 2008년 1.0 버전을 발표한 이래 지속적으로 취약한 업데이트 SW 추가
	Winzip (Corel)			
	Winamp (AOL)			
	MacOS (Apple)			
	OpenOffices			
	LinkedIn			
	기타 57종			
2007년	Cam2PC (Nabosoft)	○	○	▪ 서명검증 미실시
2006년	Firefox (Mozilla)	○	○	▪ 업데이트 설정정보 미인증으로 MITM 공격 가능
	Fugu (Freeware)	○	○	▪ 서명검증 미실시
	VirusScan (McAfee)	○	○	▪ 서명검증 미실시
	Virex (McAfee)	○	○	▪ 서명검증 미실시
	Zango (180Solutions)	○	○	▪ 서명검증 미실시

※국외 SW 업데이트 체계 취약점 현황은 알려진 취약점 정보의 부족으로 보안위협을 추정하여 기재하였으며 추정이 불가한 경우도 있음

2-3 피해보상 사례

- ‘SK 커뮤니케이션즈 해킹 사건’으로 인해 법원은 네이트 해킹 피해 사용자 위자료 100만원 지급 판결 내림

※ **“법원, 네이트 해킹피해 위자료 100만원, 첫 보상 판결”, 매일경제, 2012.04.**

- ‘농협 금융전산망 마비 사태(2011.4)’로 인해 농협은 IBM으로부터 현물 약 115억원을 보상받음

※ **“농협, 전산대란 책임있는 IBM과 굴욕적 협상”, 서울파이낸스, 2012.10.**

- ‘3·20 사이버테러’로 피해를 입은 농협이 보안 솔루션 공급업체 안랩을 상대로 손해배상 청구를 검토

※ **“농협, 안랩에 ‘3.20 사이버테러’ 피해 손해배상 청구 검토 논란”, 이투데이, 2013.04.**

3 기존 SW 업데이트 체계의 문제점

3-1 상호 인증 보장 기술 미적용

- 국내의 SW 업데이트 체계에서는 업데이트 서버와 업데이트 클라이언트 사이의 상호 인증 기능을 제공하지 않기 때문에 위장 업데이트 서버를 구축할 경우 업데이트 클라이언트에서는 정상 업데이트 서버로 오인하여 업데이트가 수행됨

3-2 통신 채널 보호 기술 미적용

- 국내의 일부 SW 업데이트 체계를 제외한 대부분의 업데이트 체계에서는 업데이트 서버와 업데이트 클라이언트 사이의 통신 채널을 보호하지 않기 때문에 업데이트와 관련된 중요한 정보파일 및 통신 프로토콜 정보가 노출되어 공격자가 업데이트 체계를 공격하는데 활용됨
  - 업데이트 정보파일 및 실행파일을 암호화하지 않고 평문으로 전송
  - 업데이트 정보파일을 XOR 인코딩하여 전송
  - 업데이트 정보파일 및 실행파일을 잘 알려진 알고리즘(zip, bzip2, cab 등)으로 압축
  - 업데이트 서버와 업데이트 클라이언트간의 통신 프로토콜 분석 가능

3-3 기밀성 보장 기술 미적용

- 국내의 SW 업데이트 체계에서는 업데이트와 관련된 중요한 정보파일 및 설치파일을 보호하지 않고 평문 형태로 해당 업데이트 클라이언트의 임시폴더 및 설치 폴더에 저장하거나 낮은 수준의 파일 보호만을 하고 있기 때문에 소프트웨어 역공학을 이용한 분석을 통해 공격자가 업데이트 체계를 공격하는데 활용됨
  - 업데이트 정보파일(xml, ini 등) 및 실행 파일(exe, dll 등)을 암호화하지 않고 평문으로 저장

3-4 저수준의 무결성 보장 기술 적용

- 국내의 SW 업데이트 체계에서는 업데이트 정보파일 및 실행파일 실행 시에 파일에 대한 무결성 검사를 수행하지 않거나, 무결성 검사를 수행하는 경우에도 안전하지 않은 방법을 이용하고 있으므로 공격자가 무결성 검사를 우회하여 업데이트 클라이언트에서 악성프로그램을 실행가능함
  - 업데이트 정보파일과 실행파일에 대한 무결성 검사를 수행하지 않고 업데이트
  - 업데이트 정보파일에 대한 무결성 검사만 수행하고 실행파일에 대한 무결성 검사 미실시
  - 업데이트 정보파일에 대한 무결성 검사 없이 실행파일 무결성 검사
  - 실행파일 이외의 파일(플래시 파일, 그림 파일, 설정파일 등)에 대한 무결성 검사 미실시
  - 단순한 XOR 인코딩 방법을 이용하여 업데이트 정보파일의 무결성을 검사
  - 해쉬(MD5)를 이용하여 정보파일 및 실행파일의 무결성 검사
  - 해쉬(MD5)와 대칭키 암호화기법(AES)을 이용하여 실행파일의 무결성 검사

4장 대응방안

보안위협		파급력 (L/M/H)	보안강화 대책					
			파일보호 (수준: L/M/H)			채널보호 (수준: L/M/H)		
			파일 체크섬	파일 암호화	디지털 서명	고유값 상호 인증	PKI 상호 인증	PKI 상호 인증 + 채널 암호화
서버	**서버장악** ('09년 7·7, '11년 3·4 DDoS)	H	L	M	H	–	–	–
	악성코드 업로드 ('13년 3·20 사태)	H	L	M	H	L	M	H
채널	**대규모 MITM** DNS 포이즈닝 라우터 장악 공개 Proxy	H	L	M	H	–	–	H
	소규모 MITM 스위치 장악 ARP 스푸핑	M	L	M	H	–	–	H
클라이언트	**위장서버 접속** ('11년 SK컴즈 개인정보유출)	H~M	L	M	H	L	M	H

※파급력 및 보호대책의 수준을 3단계로 분류(L: Low, M: Medium, H: High)

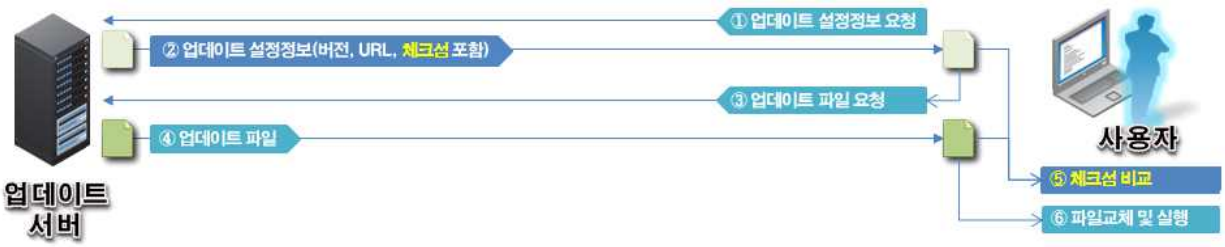
1 파일보호

- 파일 체크섬, 파일 암호화, 디지털 서명 등의 방법으로 업데이트 파일을 조작할 수 없도록 조치하는 보안강화 방법
- 업데이트 서버로부터 클라이언트로 파일만 전송되는 체계에서는 채널보호 없이 파일보호 수단만으로도 안전성을 강화할 수 있음

1-1 파일 체크섬을 이용한 무결성 검증

- 구조
  - 서버: 업데이트 설정정보에 파일 체크섬 정보 포함
  - 클라이언트: 실제로 다운로드 받은 파일에 대해서 체크섬 값을 계산한 후 업데이트 설정정보에 기재된 체크섬 값과 일치하는지 여부를 확인

- 장점
  - 가장 적은 비용으로 빠르게 구현 가능 (대부분의 업데이트 체계가 이 방법 채택)
- 단점(주의사항)
  - 체크섬 생성 알고리즘이 클라이언트 프로그램에 포함되므로 역공학 등을 통해서 체크섬 생성 알고리즘이 노출되기 쉬우며, 이 경우 보안강화 조치는 무력화됨



1-2 파일 암호화를 이용한 기밀성 보장

- 구조
  - 서버: 업데이트 파일을 암호화해서 저장
  - 클라이언트: 다운로드 받은 파일을 복호화해서 설치 (복호화 실패 시 설치 실패)
- 장점
  - 비교적 적은 비용으로 구현 가능
  - 암호화 알고리즘은 서버에만 존재하므로 서버장악 외에는 파일 위·변조 불가
  - 암호화된 파일을 변조하는 경우 올바르게 복호화 되지 않으므로 설치 불가
- 단점(주의사항)
  - 알려진 대칭키 암호화 알고리즘을 사용하면 클라이언트 프로그램에 대한 역공학 등을 통해 암호키와 알고리즘이 노출될 수 있으며, 이 경우 보안강화 조치는 무력화됨

1-3 디지털 서명을 이용한 무결성 검증

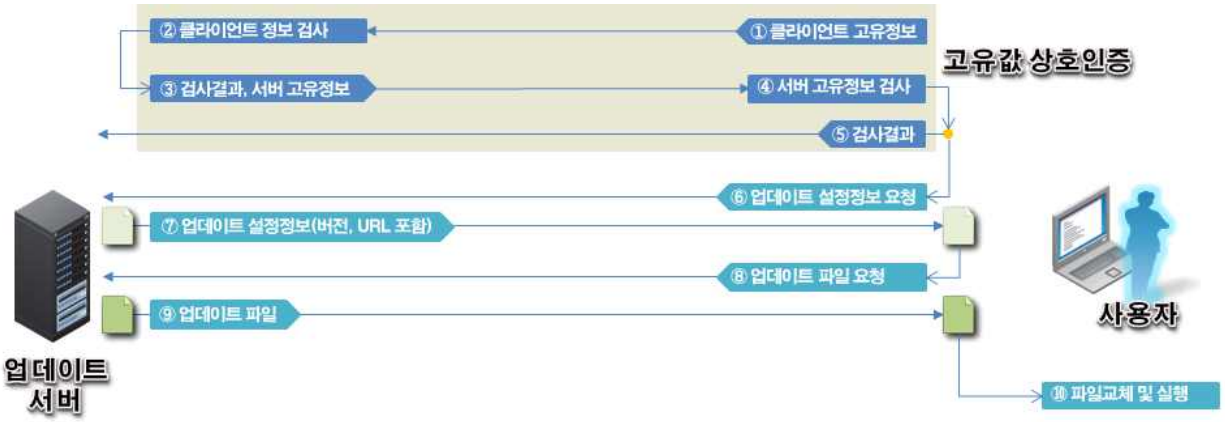
- 구조
  - 서버: 업데이트 파일에 대한 디지털 서명
  - 클라이언트: 다운로드 받은 파일의 디지털 서명 유효성 검사 (무결성 및 발급주체 확인)
- 장점
  - 현존하는 파일보호 대책 중 가장 안전 (PKI 방식)
-

- 단점(주의사항)
  - 업데이트 서버에서 디지털 서명 생성이 가능한 경우 서버장악 후 악성코드에 디지털 서명을 붙여서 배포 가능
  - 따라서, 반드시 디지털 서명의 생성은 업데이트 서버 외부의 별도 PC에서 수행해야 함
  - 또한, 디지털 서명 생성이 가능한 PC가 해커에게 장악되는 경우 안전성이 보장되지 않음
  - 인증기관(CA)에 유지보수 비용을 지불해야 함

2 채널보호

- 채널보호는 통신채널의 양 주체인 서버와 클라이언트가 서로 상대방을 인증하는 방법으로, 크게 고유값 상호인증, PKI 상호인증, PKI 상호인증+채널암호화의 3가지 방법이 사용됨
- 일반적으로 업데이트 체계에서 파일만 송수신된다면 채널보호 대책 없이도 디지털 서명과 같은 강력한 파일보호 대책만으로도 안전성 강화가 가능하나, 업데이트 체계의 특성상 파일 이외의 개체(예: 프로그램 실행 명령 등)를 송수신하는 경우라면 파일보호 대책에 보호를 받지 못하므로 채널보호 대책도 함께 강구해야 함

2-1 고유값 상호인증을 이용한 인증

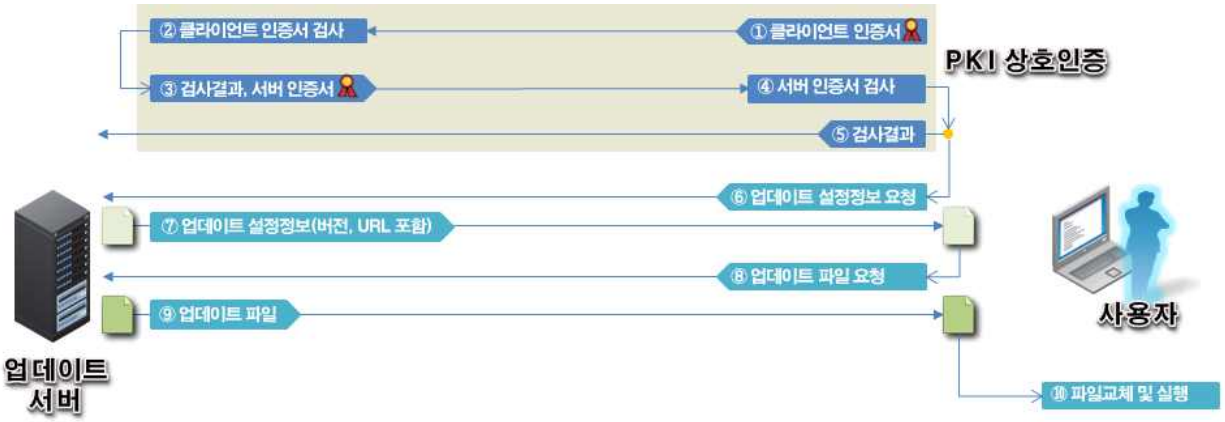


- 구조
  - 서버: 클라이언트가 제공한 고유값이 약속한(등록된) 값인지 검사 (예: IP주소 또는 ID/PW)
  - 클라이언트: 서버가 제공한 고유값이 약속한(등록된) 값인지 검사 (예: 서버 S/N)
- 장점
  - 비교적 적은 비용으로 구현 가능
- 단점(주의사항)
  - 클라이언트가 인정하는 서버의 고유값은 각 클라이언트마다 서로 달라야함
  - 서버의 고유값이 모든 클라이언트에게 동일하다면 쉽게 노출되어 채널보호 대책이 무력화됨
  - 서버 측에서는 유효한 클라이언트의 고유값을 등록하는 절차가 필요함 (예: 최초 설치 시)

2-2 PKI 상호인증을 이용한 인증

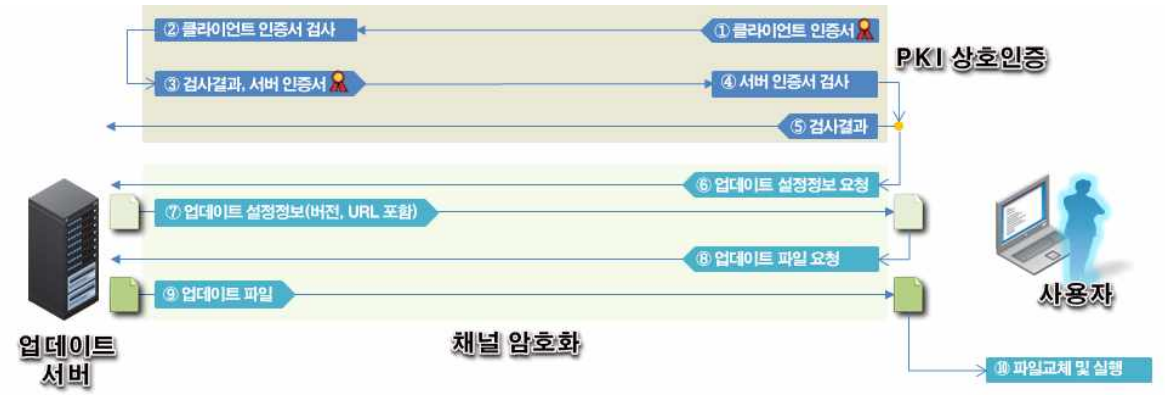
- 구조
  - 서버: 클라이언트가 제공한 인증서 검사(유효성 및 발급주체)
  - 클라이언트: 서버가 제공한 인증서 검사(유효성 및 발급주체)
- 장점
  - 인증서의 유효성과 발급주체 검사로 강력한 상호인증 방법임
- 단점(주의사항)

– 제시된 인증서가 유효성과 함께 발급주체가 올바른 서버 또는 클라이언트인지 확인하지 않는 경우 가짜 인증서를 사용해서 MITM 공격 가능



2-3 PKI 상호인증 + 채널 암호화를 이용한 인증 및 기밀성 보장

- 구조: PKI 상호인증과 더불어 채널 암호화 수행 (예: SSL)
- 장점: 현존하는 채널보호 대책 중 가장 안전
- 단점(주의사항)
  – 제시된 인증서의 유효성과 함께 발급주체에 대한 검사를 수행하지 않는 경우 가짜 인증서를 사용해서 (SSL) MITM 공격 가능



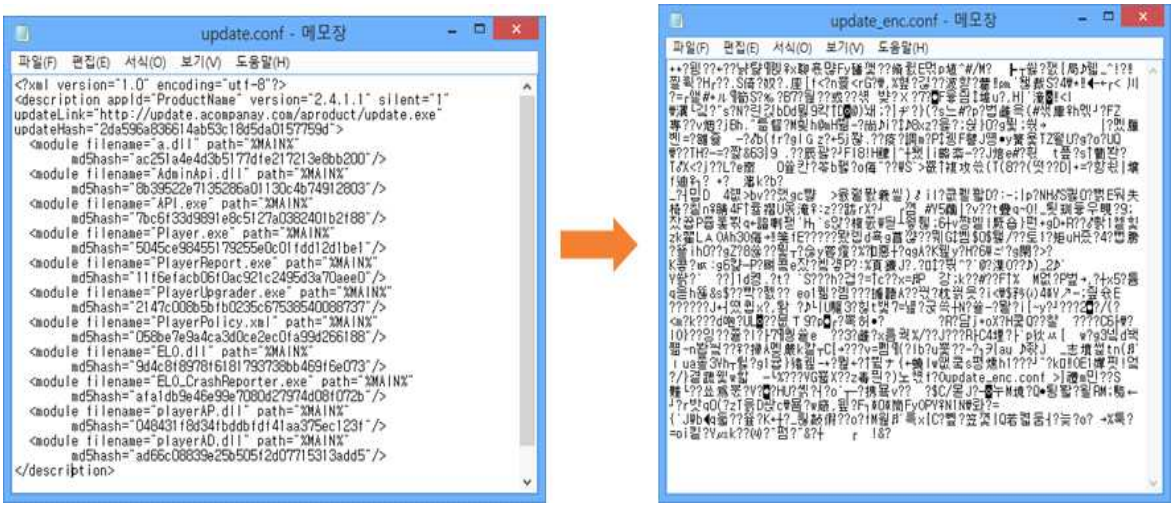
5장 SW 업데이트 체계 보안 십계명

1 업데이트 설정파일 무결성 검사

- 업데이트 프로그램은 주기적으로 또는 사용자가 해당 SW를 실행할 때 업데이트 필요 유무를 판단하기 위해 업데이트 정보파일을 업데이트 서버에 요청한다. 서버로부터 전송받은 업데이트 정보파일의 버전정보를 비교하여 로컬에 설치된 버전보다 최신 버전일 경우, 업데이트 정보파일에 기재된 업데이트 URL에 접속하여 업데이트 파일을 다운로드 받고 설치한다.
- 업데이트 설정파일을 해커가 통신 채널 중간에서 변조할 수 없도록 디지털서명 검증을 수행해야 한다. (검증 실패 시 파일 즉시 삭제)
- 위협
  - 업데이트 설정 파일의 무결성 검사를 수행하지 않을 경우 해커가 변조하여 업데이트가 없을 경우에도 업데이트가 있는 것으로 위장할 수 있으며, 위장 서버로 접속하여 업데이트 파일을 가장한 악성파일을 다운로드 및 실행하도록 할 수 있다.
- 개발 시 주의사항
  - MD5 등의 해쉬만 사용할 경우에는 공격자가 쉽게 변조 가능하므로, 반드시 디지털서명 검증 방식으로 무결성 검사를 수행해야 한다.
  - 서명의 유효성 검사뿐만 아니라, 서명에 사용된 인증서의 상태도 반드시 확인해야 한다.

2 업데이트 설정파일 암호화

- 업데이트 정보파일을 암호화하여 사용할 경우 해커가 업데이트 정보파일을 변조하는 것을 어렵게 만들 수 있다.



- 위협
  - 업데이트 정보파일을 암호화하지 않고 평문으로 사용할 경우, 해커에게 버전정보 및 업데이트 URL 정보 등의 업데이트 관련 정보가 노출되어 조작될 수 있다.

- 개발 시 주의사항
  - 암호키는 소스코드 상에 고정적으로 사용할 경우, 역공학에 의해 노출 될 수 있으므로, 공개키 기반의 키 암호화 방식을 사용 할 것을 권고한다.

3 실행파일 디지털 서명 검증

- 실행파일에 대한 무결성 검증 방법으로 디지털서명을 사용해야 함. 디지털 서명에 사용된 인증서의 유효성을 검증하고, 검증 실패 시에는 관련파일을 즉시 삭제하고 업데이트를 종료하여야 한다.
- 위협
  - 디지털서명에 사용된 인증서의 유효성을 검사하지 않을 경우 유효하지 않은 인증서에 의해 서명된 악성코드가 실행될 가능성이 있다.
  - 디지털서명 검증에 성공할 경우에도, 서명자와 버전 정보가 같은지 확인하지 않을 경우, 다른 회사의 취약한 제품이 설치되거나, 동일 제품의 취약한 버전 또는 동일 회사의 다른 취약한 제품으로 설치가 가능한 문제가 있다.
- 개발 시 주의사항
  - 인증서의 유효성 검증 실패 시에는 즉시 해당 파일을 삭제 조치하고 업데이트 과정을 종료한다.
  - 디지털 서명 검증 시에 서명자와 버전 정보의 최신성을 확인을 위해 파일의 무결성 검증(해쉬 검사 등)도 반드시 수행해야 한다.

4 비실행파일 디지털 서명 검증

- 실행파일 뿐만 아니라 그림, 플래시, 문서 등을 포함한 모든 비실행 파일을 업데이트 할 경우에도 반드시 디지털서명을 사용하여 무결성 검증을 수행하고, 검증 실패 시에는 즉시 삭제하여야 한다.
- 위협
  - S사 보안 프로그램의 자동업데이트 모듈에서 업데이트 파일 중 실행파일에 대해서는 무결성 검증을 수행하지만 광고를 위한 그림파일에 대해서는 무결성 검증을 수행하지 않음으로 인하여 공격자가 업로드한 악성코드가 내장된 그림파일을 클라이언트의 특정 폴더에 저장한 뒤, 기타 취약점을 이용하여 악성코드를 실행하는 사례가 있었다.
  - C사는 PKI 기반의 디지털서명 검증을 이용한 무결성 검증을 수행하였으나, 해커에 의해 개발자 PC에 저장된 개인키가 유출됨으로써 악용된 사례가 있으므로, 배포와 관련된 개인키는 별도의 저장소에 안전하게 보관하여야 한다.
- 개발 시 주의사항
  - 비실행파일 각각에 대하여 무결성 검증을 수행하기 보다는 Cab 파일 형태로 압축한 후 압축파일에 대하여 디지털서명을 삽입하는 방법을 권장한다.
  - 코드 서명 검증 실패 시에는 즉시 해당 파일을 삭제 조치하고 업데이트 과정을 종료한다.
  - MD5 등의 해쉬는 공격자가 쉽게 변조 가능하므로 지양하고, 반드시 PKI 기반의 디지털 서명 검증 방식으로 구현하는 것이 안전하다.
  - 디지털 서명 검증 시에 서명자와 버전 정보의 최신성을 확인을 위해 파일의 무결성 검증(해쉬 검사 등)도 반드시 수행해야 한다.

5 제휴 서비스 프로그램 디지털서명 검증

- 제휴 서비스 프로그램 설치 시에도 무결성을 검증한 뒤 설치하도록 한다. 검증 실패 시에는 즉시 삭제해야 한다.
- 위협
  - H사에서 무료로 배포하는 SW의 경우, H사의 프로그램에 대한 업데이트에 대해서는 무결성 검증을 수행하지만, 기본 설정으로 설치되는 제휴사의 프로그램에 대해서는 무결성 검증을 수행하지 않는 취약점이 있었다. 이 경우, MITM 공격으로 위장 서버에 접속하여 악성코드를 다운로드 받고 설치할 수 있다.
- 개발 시 주의사항
  - 제휴 서비스로 설치되는 프로그램의 경우에도, 다운로드 받는 설치 파일에 대한 디지털서명 검증을 반드시 확인해야 한다.
  - 디지털서명 검증 시에 서명자와 버전 정보의 최신성을 확인을 위해 파일의 무결성 검증(해쉬 검사 등)도 수행해야 한다.
  - 파일에 대한 디지털서명 검증 실패 시에는 즉시 해당 파일을 삭제 조치해야 한다.

6 PKI 기반 통신채널 보호

- 정상적인 업데이트 절차에 해커가 끼어들지 못하도록 SSL과 같은 통신채널 보호를 적용하는 것이 좋다.
- PKI기반의 상호 인증 및 통신채널 암호화를 수행하므로 중간에 MITM 공격에 대응할 수 있다.
- 위협
  - E사가 무료로 배포하는 SW는 통신 채널 보호를 위해 SSL을 사용하였으나, 서버 인증서의 유효성을 검사하지 않음으로 인하여, 누구나 발급할 수 있는 자가 서명된(Self-Signed) 테스트용 인증서를 통한 MITM 공격이 가능하였다.
- 개발 시 주의사항
  - 업데이트 프로그램에서 SSL을 통한 업데이트 서버 인증 시 서버 인증서의 유효성 검증을 정확히 수행해야 한다.

7 업데이트 서버 주소 무결성 검증

- 업데이트 서버 URL이 변조되지 않음을 보장해야 한다. 이를 위해, 업데이트 서버 주소는 소스코드 상에 고정적으로 사용하도록 하고, 서버와 클라이언트 간의 상호 인증을 수행해야 한다.
- 위협
  - ActiveX Control이나 자바 애플릿 등과 같은 웹 플러그인 SW에서 간혹 업데이트 서버 URL을 고정적으로 사용하지 않고 웹페이지에서 메소드의 입력으로 받아서 설치하는 경우가 있다. 이런 경우에는 해커에 의해 변조되어 위장 서버로 접속할 수 있다.
  - 업데이트 서버 URL을 업데이트 프로그램 소스코드 상에 고정적으로 사용할 경우에도 ARP 스푸핑이나 파밍 공격 등에 의해 위장서버로 접속할 수 있다.

- 개발 시 주의사항
  - 업데이트 서버 URL을 업데이트 프로그램 소스코드 상에 고정적으로 사용하도록 한다.
  - 불가피하게 입력으로 받아야할 경우에는 특정 도메인으로 제한해야한다.
  - 도메인 및 URL 검사 시에는 문자열 비교를 정확히 수행해야 한다. 예를 들어, update.server.com 서버의 주소를 검사할 때 update.server.comp.com과 같은 유사 URL로 접속하는 것을 허용해서는 안된다.
  - 업데이트 서버와 클라이언트 간의 상호 인증을 수행하는 방법에는 SSL과 같은 통신채널 보호 또는 업데이트 파일 자체의 무결성 검증을 우선적으로 적용한다.

8 디지털서명에 사용된 인증서 상태 검사

- 업데이트 설정파일 및 업데이트 파일의 디지털서명 검증 시에는 디지털서명의 유효성 검증 뿐만 아니라, 서명에 사용된 인증서의 유효기간, 인증 경로, 인증서 효력정지 또는 해지 여부 등의 상태를 반드시 검사해야 한다.
- 위협
  - 유효한 인증서를 이용하여 서명된 업데이트 관련 파일일 경우에도 개발자의 실수에 의해 개인키가 유출됨으로써 유출된 개인키로 서명된 악성프로그램이 실행될 수 있다.
- 디지털서명 검증 시 주의사항
  - 디지털서명 검증 모듈 개발 시에 서명에 사용된 인증서의 CRL[1] 또는 OCSP[2] 정보를 이용하여 유효한 인증서인지를 확인해야 한다.

9 업데이트 프로그램 분석방지를 위해 코드 난독화 적용

- 소프트웨어 역공학을 통해 실행코드로부터 업데이트 체계 구현에 사용된 통신 프로토콜, 암호화 알고리즘, 업데이트 관련 설정파일 등의 모든 내용이 분석 가능하므로 코드 난독화를 적용하여 업데이트 체계에 대한 상세 분석을 어렵게 해야 한다.
- 코드 난독화 적용 시 주의사항
  - 코드 난독화를 적용할 경우 수행 속도 및 안전성의 문제가 발생할 수 있으므로 프로그램 내의 보호가 필요한 부분에 코드 난독화를 적용하도록 권장한다.

10 자동업데이트 보안 미비 시 기능 삭제

- 자동업데이트 체계에 대한 상기 필수 보안 대책이 미비할 경우, 자동업데이트 기능을 삭제하고 사용자가 수동으로 사이트에 접속하여 다운로드 받아 설치하는 수동 업데이트 방식을 사용하도록 유도해야 한다.
- 사용자가 제품 업데이트 다운로드 페이지에 접속하여 다운로드 받을 경우에도, 제작사의 서명이 올바른지 확인하도록 공지할 필요가 있다.

1) CRL(Certificate Revocation List) : 인증서 폐기 목록
2) OCSP(Online Certificate Status Protocol) : 온라인 인증서 상태 프로토콜

부록 SW 업데이트 체계 보안 십계명(요약)

번호	항목	내용	비고
1	업데이트 설정파일 디지털 서명 검증	-업데이트 설정파일을 해커가 통신채널 중간에서 변조할 수 없도록 디지털서명 검증을 수행해야 함 (검증 실패 시 파일 즉시 삭제) -디지털서명 검증이란 비대칭키 암호 알고리즘을 사용하여 설정파일을 업체의 개인키로 서명하고, 클라이언트 업데이트 프로그램에서 이를 업체의 공개키로 검증하는 방법을 말함 -MD5 등의 해쉬는 공격자가 쉽게 변조 가능하므로 지양해야함	**필수**
2	업데이트 설정파일 암호화	-업데이트 정보를 포함하는 설정파일은 암호화를 통해서 임의로 분석이 불가능하도록 변경하는 것이 좋음	선택
3	실행파일 디지털 서명 검증	-업데이트 서버를 통해 배포할 업데이트 파일은 반드시 디지털서명 검증을 수행해야 함. 이를 위해 업데이트 파일은 업체의 개인키로 서명한 후 업로드 해야 함 -클라이언트 업데이트 프로그램은 다운로드 받은 파일을 임시폴더에 먼저 저장하고 업체의 공개키로 디지털서명을 검증하여 신뢰된 파일만 사용하여야 함 (검증 실패 시 파일 즉시 삭제) -EXE, DLL, SYS와 같은 실행파일의 코드 서명 검증 시에는 서명의 유효성뿐만 아니라 서명자 정보의 일치 유무도 정확히 검사해야 함 -MD5 등의 해쉬는 공격자가 쉽게 변조 가능하므로 사용을 지양해야함	**필수**
4	비실행파일 디지털 서명 검증	-실행파일 외에 그림, 플래시 등의 비실행파일을 업데이트 할 경우에도 반드시 디지털 서명 검증을 수행해야 함 (검증 실패 시 파일 즉시 삭제) -비실행파일의 경우 cab파일 형태로 압축한 후 압축파일에 디지털 서명을 삽입할 것을 권장 -MD5 등의 해쉬는 공격자가 쉽게 변조 가능하므로 지양해야 함	**필수**
5	제휴 서비스 프로그램 설치시 디지털서명 검증	-제휴 프로그램 설치 시에도 무결성을 검증한 뒤 설치하도록 해야함 (검증 실패 시 파일 즉시 삭제) -제휴 프로그램의 무결성 검증을 위해서는 공개키 기반의 디지털서명 방식을 사용해야 함 -제휴 프로그램 디지털서명 확인 시에는 반드시 서명자, 버전을 확인해야함(파일 해쉬 검증)	**필수**
6	PKI 기반 통신채널 보호	-정상적인 업데이트 절차에 해커가 끼어들지 못하도록 SSL과 같은 채널 보호를 적용하는 것이 좋음	선택

7	업데이트 서버 주소 무결성 검증	-업데이트 서버 URL이 변조되지 않도록 업데이트 서버 주소는 소스코드 상에 고정적으로 사용하고, 서버와 클라이언트 간의 상호 인증을 수행해야 함	선택
8	인증서 상태 검사	-업데이트 관련 파일 디지털서명 검증 시에 서명에 사용된 인증서의 유효기간, 인증 경로, 인증서 효력 정지 또는 해지 여부 등의 상태를 검사해야 함	**필수**
9	실행파일 난독화	-업데이트 프로그램 등의 실행파일은 난독화 도구를 활용하여 임의로 분석이 어렵도록 하는 것이 좋음	선택
10	자동업데이트 보안 미비 시 기능 삭제	-자동업데이트 체계에 대한 상기 필수 보안 대책이 미비할 경우, 자동업데이트 기능을 삭제하고 사용자가 수동으로 사이트에 접속하여 다운로드 받아 설치하는 수동업데이트 방식을 사용하도록 유도해야함 -사용자가 수동 다운로드 시에도, 제작사의 서명이 올바른지 확인하도록 공지해야 함	**필수**