

nProtect Online Security V1.0

적용 가이드



© 2000-2016 잉카인터넷

본 사용설명서의 내용과 프로그램은 저작권법과
컴퓨터 프로그램 보호법에 의해 보호 받고 있습니다.

(주)잉카인터넷

서울특별시 구로구 디지털로26길 5 에이스하이엔드타워 1차 12층

개정이력

Version	Updates	Date
1.00	Initial Draft	2014-11-25
1.01	FAQ 내용 추가	2014-12-15
1.02	WAS 서버의 E2E 데이터 검증 내용 추가	2015-01-05
1.03	마우스입력기 적용 가이드 추가	2015-01-10
1.04	마우스입력기 Replace 입력 처리 방식 추가	2015-01-20
1.05	FDS 데이터 검증 추가, 가상운영체제 확인 함수 추가	2015-02-09
1.06	FAQ 수정	2015-02-11
1.07	HTML 웹 표준 처리방법, 키보드보안 키 필터링 추가	2015-02-27
1.08	데이터 재가공 방법, 모바일 키패드 적용방법 추가	2015-03-01
1.09	수동복호화, 렌더링 모드 참조, <script charset> 추가	2015-03-02
1.10	키보드보안/마우스입력기 동적필드지원. 마우스입력기 노출 위치 옵션추가	2015-03-12
1.11	수동 복호화 사용방법 추가 모바일 단말정보수집 적용방법 추가	2015-04-07
1.12	단말정보수집 방법 수정. 모바일 단말정보수집 방법 수정. 마우스입력기 크기조절 기능 사용 추가	2015-04-23
1.13	키보드보안/마우스입력기 동시 사용방법 추가 모바일 보안 키패드 적용가이드 내 API 내용 추가	2015-05-29
1.14	모바일 보안 키패드 적용가이드 내용 수정 및 API 추가 - 툴바 뷰 추가, 입력 값 제한 설정 추가 - 쿠키 미 지원 브라우저에서의 사용방법 추가 - 환경설정 파일 상세설정 추가 / FDS 추가수집항목 추가	2015-06-03 2015-06-10 2015-06-12 2015-06-24
1.15	NOS 기능명세 추가. 일반키보드보안 설정방법 추가	2015-07-07
1.16	일반 키보드 보안, 확장 E2E 키보드 보안 흐름도 설명 변경	2015-07-08
1.17	Linux 소프트웨어 요구사항 및 삭제 방법 수정	2015-08-17
1.18	모바일 기기정보 수집 API 추가 - raw 데이터 전달 함수 추가	2015-08-27
1.19	마우스입력기 toggle기능 옵션 추가	2015-08-31
1.20	모바일-안드로이드 UI Visibility 메소드 추가	2015-08-31

1.21	키보드보안 Flash SDK 적용가이드 추가 마우스입력기 임의의 버튼 기능 정책 설명 추가	2015-09-08
1.22	web.xml 에 필터처리 예외 설정(ExcludePattern) 추가 nprotect.properties 정책(9개) 추가 PC및 모바일 환경에서 이미지 축소/확대 설명추가 java.util.Map방식의 복호화 방법 추가 NOS 기능확장 및 발생 이벤트 및 사용방법 추가 자바스크립트 toJson, serializeArray, encryptValue 함수 추가	2015-11-16
1.23	모바일 보안 키패드 안드로이드 내용 정리 - 구현 가이드 내용 최신 내용에 맞게 재수정 - 사용하지 않는 API 리스트 삭제	2015-11-20
1.24	다른 개발 언어 지원(Java 기반 WAS를 통한) 설명 추가	2015-11-30
1.25	Flash SDK 적용가이드 수정	2015-12-07
1.26	넥사크로플랫폼 적용가이드 추가	2015-12-10
1.27	HttpRequest 값의 문자열 인코딩 예외추가 키보드보안/마우스입력기 복호화 방식 구분 API추가	2016-01-07
1.28	PC지정서비스 가이드 추가 키보드보안/마우스입력기 색상변경 인터페이스추가 환경설정(web.xml)에 페이지별 문자열 인코딩 가이드 추가	2016-03-07
1.29	넥사크로플랫폼 적용가이드 추가	2016-03-09
1.30	ASP.NET 적용가이드 추가	2016-05-23
1.31	Node.js 적용가이드 추가 데이터베이스 컬럼 타입 및 길이 별첨 추가	2016-07-14
1.32	단말정보수집 사용자 추가데이터 전달 방법 추가	2016-07-20
1.33	모바일 보안 키패드 (안드로이드) API 추가	2016-08-10
1.34	모바일 보안 키패드 (iOS) API 추가	2016-08-16
1.35	마우스입력기 인증서연동을 위한 데이터 재 암호화 방식 추가 마우스입력기/키보드보안 검증 함수(verify) 예외 구체화 마우스입력기 터치효과 모드 추가 구간암호화/보안전송 적용가이드 추가 무설치 단말정보수집 적용가이드 추가	2016-11-04
1.36	모바일 단말정보수집 수집 정보 추가	2016-11-10
1.37	npPfsCtrl.doFocusOut(), 넥사크로플랫폼 serializeArray, destroyFields 함수 추가	2016-11-28
1.38	nprotect.properties 정책(5개) 추가 운영환경 수정	2017-03-21

목차

1. nProtect Online Security 소개.....	9
1.1. 시스템 구성 및 주요 기능	9
1.1.1. 시스템 구성	9
1.1.2. 주요 기능.....	9
1.1.3. 동작 구성도	10
1.2. 기능 명세	11
1.2.1. 키보드보안	11
1.2.2. 방화벽.....	18
1.2.3. 실시간 프로세스 감시	18
1.2.4. 단말정보수집(FDS).....	19
1.2.5. 피싱방지(Anti-Phishing)	19
1.2.6. 파밍방지(Anti-Pharming).....	21
1.2.7. 화면 캡처 방지	22
1.2.8. 마우스 입력기	23
1.2.9. 구간암호화/보안전송	25
1.2.10. 무설치 단말정보 수집	26
1.3. 암호화 알고리즘 및 키 생명주기.....	28
1.3.1. 암호화 알고리즘.....	28
1.3.2. 암호화 키 생명주기.....	28
2. 사업수행 계획.....	29
2.1. 사업개요	29
2.1.1. 사업명.....	29
2.1.2. 개요	29
2.1.3. 목적	29
2.1.4. 주요 업무내용	29
2.1.5. 수행 기간.....	29
2.1.6. 산출물.....	30
2.2. 투입인력 계획	31
2.2.1. 투입 인력 및 구성 개요 : 총 2명.....	31
2.2.2. 인력 구성 및 수행 업무	31

2.3. 수행일정 계획	32
2.3.1. 수행일정	32
2.3.2. 세부 계획	33
2.4. 교육계획	35
2.4.1. 설치 및 운영 교육	35
2.4.2. 장애유형 및 대응(응급)조치 교육	35
3. 운영 환경	36
3.1. 사용자 운영 환경	36
3.1.1. 하드웨어 요구사항	36
3.1.2. 소프트웨어 요구사항	36
3.2. 서버 운영 환경	39
3.2.1. 소프트웨어 요구사항	39
4. 서버 적용 가이드	40
4.1. 라이브러리 설치	40
4.1.1. jar 파일 복사	40
4.1.2. /pluginfree/ 폴더 복사	40
4.1.3. 마우스입력기 리소스 복사	40
4.1.4. 환경설정 파일 복사	40
4.1.5. nppfs-x.x.x.js의 수정	41
4.2. 환경설정	41
4.2.1. Filter 기본 설정	41
4.2.2. 환경설정 파일 상세설정	45
4.3. 페이지 적용	53
4.3.1. 문서의 HTML 표준 확인	53
4.3.2. nProtect Online Security 공통	55
4.3.3. nProtect Online Security 공통 - 업무페이지	58
4.3.4. 키보드보안(E2E) 상세설정	62
4.3.5. 키보드보안(일반) 상세설정	62
4.3.6. 마우스입력기 상세설정	63
4.3.7. 키보드보안, 마우스입력기 치환기능 사용	66
4.3.8. 단말정보수집 적용방법	67
4.3.9. 무설치 단말정보수집 적용방법	67

4.3.10.	구간암호화/보안전송 적용방법	68
4.3.11.	설치 안내 페이지	69
4.4.	고급 페이지 적용.....	72
4.4.1.	쿠키 차단 환경에서 운용 방법.....	72
4.4.2.	마우스입력기 PKI 연동가이드	73
4.4.3.	자바 도구.....	73
4.4.4.	자바스크립트 도구	76
5.	고급 적용 가이드.....	84
5.1.	NOS 이벤트 활용	84
5.1.1.	NOS 발생 이벤트	84
5.1.2.	NOS 발생 이벤트 처리	87
5.2.	ASP.NET(C#) 언어 지원.....	87
5.2.1.	ASP.NET 프로젝트 설정	87
5.2.2.	암호화 페이지	90
5.2.3.	복호화 페이지	90
5.3.	NODE.JS 언어 지원.....	91
5.3.1.	Node.js 프로젝트 설정	91
5.3.2.	암호화 페이지	94
5.3.3.	복호화 페이지	94
5.4.	다른 개발 언어 지원	95
5.4.1.	구조 설명(Java기반 WAS를 통한) API.....	95
5.4.2.	WAS 서버(NOS) 설정	96
5.4.3.	WAS 서버(PHP) 설정	97
5.5.	PC지정서비스 개발가이드	99
5.5.1.	PC지정서비스 처리 개요.....	99
5.5.2.	PC지정서비스 식별 정보 및 WAS 서버 인증	100
5.5.3.	PC지정서비스 데이터베이스 테이블 설계.....	101
5.5.4.	PC지정서비스 화면 절차 가이드	103
5.6.	넥사크로플랫폼 적용가이드	104
5.6.1.	넥사크로 스튜디오 확장 라이브러리 추가	104
5.6.2.	키보드보안 및 마우스입력기 상세 설정	105
5.6.3.	WAS 서버로 데이터 전송 처리	107
5.6.4.	넥사크로플랫폼 추가 Javascript API.....	112

6. 모바일 적용 가이드	113
6.1. 모바일 보안 키패드 적용가이드	113
6.1.1. 개요	113
6.1.2. Android 적용 가이드	114
6.1.3. iOS 적용 가이드	126
6.1.4. 서버 적용 가이드	134
6.2. 모바일 기기정보 수집 적용가이드	137
6.2.1. 모바일 기기정보 수집 방법	137
6.2.2. Android 적용 가이드	138
6.2.3. iOS 적용 가이드	141
6.2.4. 서버 적용 가이드	143
 7. 키보드보안 Flash SDK 적용 가이드	 145
7.1. FLASH SDK 파일 구성 및 API	145
7.1.1. Flash SDK 파일의 구성	145
7.1.2. PluginFreeFlashSDK 클래스	145
7.2. FLASH SDK 프로젝트 설정 및 적용방법	145
7.2.1. Flash SDK 사용을 위한 ActionScript 추가 및 수정	145
7.2.2. FlashSDK 사용을 위한 MXML 수정	147
7.2.3. FlashSDK Javascript 적용	149
 8. 제품 설치 및 삭제	 151
8.1. 설치하기	151
8.1.1. Windows	151
8.1.2. Linux	152
8.1.3. Macintosh	153
8.2. 삭제하기	155
8.2.1. Windows	155
8.2.2. Linux	156
8.2.3. Macintosh	157
 9. FAQ	 158
9.1. 설치 및 삭제	158
9.2. 제품 일반 문의	159

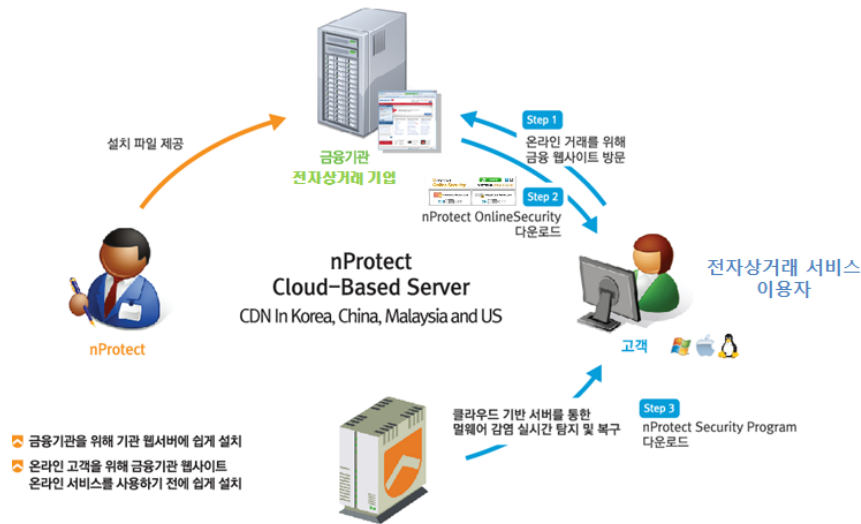
9.3. 제품 운영 관련 문의	164
9.4. 장애 관련 문의	167
9.5. FDS 수집항목	171
9.6. 미지원 환경.....	174
 10. 문의 및 고객지원	 177
10.1. 프로그램 공급자	177
10.2. 고객지원 센터.....	177
 1. 별첨 – 단말정보수집 수집항목	 178
1.1. 단말정보수집 수집항목.....	178
1.2. 무설치 단말정보수집 수집항목	180
1.3. 단말정보수집 수집항목.....	187
 2. 별첨 – 단말기 수집정보 테이블 가이드.....	 189
2.1. PC 단말기 수집정보 테이블	189
2.2. 모바일 단말기 수집정보 테이블	191
 3. 별첨 – PC지정서비스 절차 가이드.....	 193
3.1. PC 지정서비스 절차.....	193
3.1.1. 보안강화 내용고지	193
3.1.2. PC 지정서비스 안내.....	193
3.1.3. PC 지정 등록.....	194
3.1.4. PC 지정 삭제.....	195
3.1.5. PC 지정 등록 PC목록.....	196

1. nProtect Online Security 소개

1.1. 시스템 구성 및 주요 기능

1.1.1. 시스템 구성

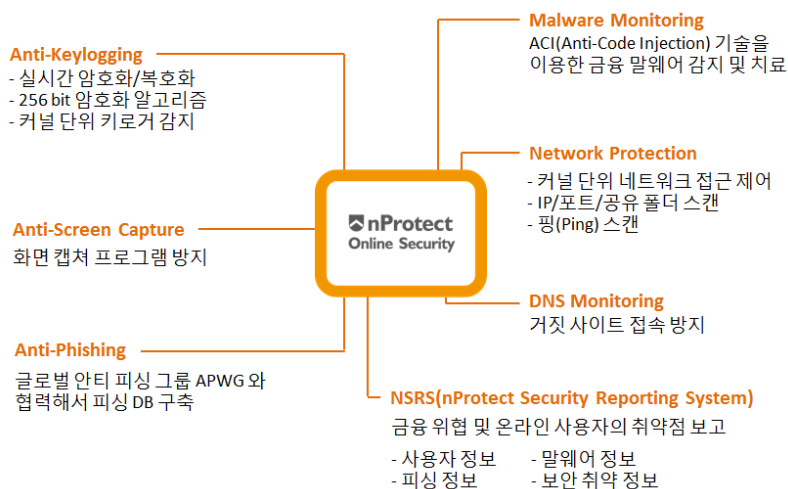
nProtect Online Security 시스템은 E-Biz 환경을 제공하는 전자상거래기업, E-biz 서비스를 사용하는 서비스 이용자, 통합보안 관제서비스를 제공하는 잉카인터넷으로 구성



[그림 1-1] 시스템 구성

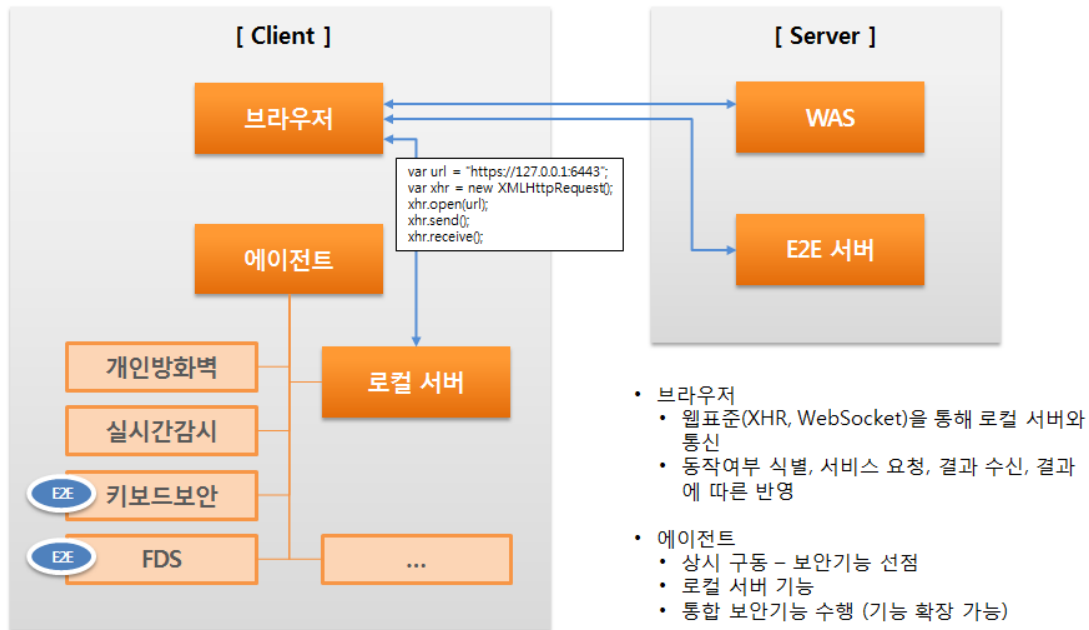
1. 서버 모듈 (jar) : 금융사 서버에 적용
2. 스크립트 파일 : 금융사 서버에 적용
3. 설치 파일, Live 업데이트 파일 : 잉카인터넷 CDN 서버 이용

1.1.2. 주요 기능



[그림 1-2] 주요 기능

1.1.3. 동작 구성도



[그림 1-3] 동작 구성도

1. 웹 브라우저를 통해 웹사이트의 보안기능을 동작시키는 JavaScript 가 적용된 페이지에 접속한다.
2. JavaScript 는 에이전트의 로컬 서버에 접속시도를 통해 에이전트가 구동 중인지 확인한다.¹
3. 접속시도가 실패하면 보안제품 설치 페이지로 유도한다.
4. 접속시도가 성공하면 Handshake 를 통해 상호인증 과정을 거친다.
5. 상호인증 과정을 성공하면 해당 페이지에 적용된 보안 기능을 요청/처리한다.

¹ 로컬 서버는 SSL 인증서를 적용하여 보안통신을 기본적으로 제공하며 SSL 인증서는 에이전트가 구동될 때마다 CDN 서버로부터 다운받습니다. 다운받는 과정에서 파일로 저장하지 않고 메모리에만 가지고 있습니다. 또한 그 내용이 암호화되어 있으므로 기밀성이 유지되며 해시 체크에 의해 무결성을 확인한 후에 사용합니다. 따라서, 파일에 대한 공격, 메모리 공격, 네트워크 구간에서의 공격에 안전합니다.

1.2. 기능 명세

1.2.1. 키보드보안

1.2.1.1. 기능 설명

디바이스 드라이버(Device Driver)의 동적 로딩(Dynamic Loading)을 통한 개인 키보드보안모듈을 포함하고 있으며, 이를 통해 원천적인 키로거를 차단 할 수 있습니다. 따라서, 원천적으로 차단함으로써 사용자가 안심하고 패스워드를 입력함으로써, 해킹이나 정보 유출에 보다 확실하게 대응이 가능합니다.

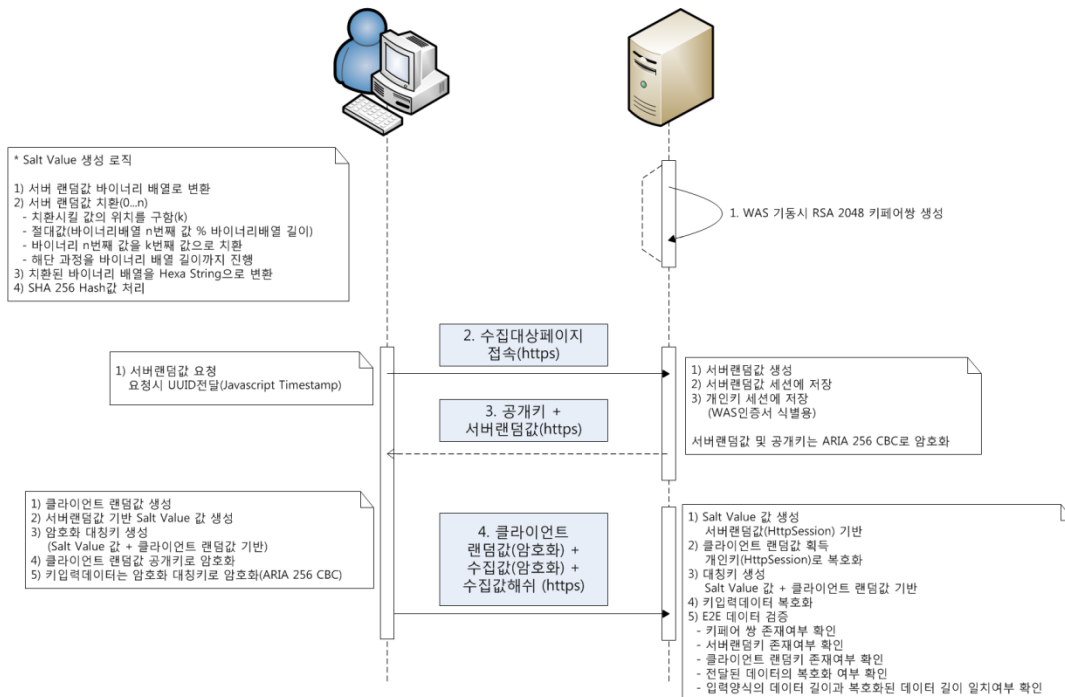
일반 키보드 보안은 실시간 암호 복호화를 통한 방식으로 키보드를 입력 할 경우 무력화 기능뿐만 아니라, 입력사항에 대한 암호방식으로 패스워드 혹은 정보 유출을 시도하는 프로그램의 기능을 무력화 합니다.

- 백 도어 또는 해킹 툴에 의한 키 입력 가로채기 무력화
- 일반적인 키보드 후킹(Hooking) 프로그램 무력화
- 상용 키로그 프로그램 무력화
- 실시간 암호 복호화를 통한 키보드 보안

1.2.1.2. 기대 효과

- 키보드입력 정보 Hooking 방지
- 키보드 Hooking 프로그램 자동 탐지
- SEED 및 PKI 방식 등 다양한 암호화 알고리즘과 호환
- Auto Tab 기능 지원
- Auto Scroll 기능 지원
- 백도어 해킹도구(백 오리피스, 스쿨버스 등)에 의한 개인정보 유출의 완벽 차단 및 예방
- 별도의 S/W 설치 과정 없이 인터넷 접속만으로 자동으로 프로그램 인스톨
- 신종 키로거에 대한 대응력 강화

1.2.1.3. E2E 키 교환방식 도식도



[그림 1-4] E2E 키 교환방식 도식도

1. WAS 기동시에 RSA-2048 키페어 쌍을 생성합니다. 각 WAS 마다 다른 키페어 쌍을 가지고 있습니다. 이 키페어 쌍은 서버 재시작 전까지 변경되지 않습니다. (페이지 호출시마다 생성이 가능하나 공개키 알고리즘에서 키 생성은 속도가 느립니다. 대략 2~3 초가 소요됩니다. 따라서 서비스 시에는 적용이 어려움이 있습니다.)
2. 페이지 요청 시 임의의 서버 랜덤 값을 생성하고 이 키를 RSA 공개키와 함께 ARIA256-CBC 알고리즘으로 암호화하여 클라이언트 모듈에 전송합니다. (전송시 임의의 32byte IV 값 생성)
3. 클라이언트 모듈은 서버의 랜덤키를 salt 처리하고 임의의 클라이언트 랜덤키값과 함께 [데이터 암호화용 키]를 생성합니다.
4. 생성된 클라이언트 랜덤키는 RSA 공개키로 암호화하여 서버에 전달합니다.
5. 키보드를 통한 키입력정보는 [데이터 암호화용 키]를 바탕으로 ARIA256-CBC 알고리즘을 사용하여 암호화합니다. (각 키입력마다 임의의 32byte IV 값 생성)
6. 서버는 공개키로 암호화된 클라이언트 랜덤값을 RSA 개인키로 복호화하고 서버 랜덤키를 salt 처리하여 [데이터 복호화 키]를 생성하고 [데이터 복호화 키]를 사용하여 키입력 데이터를 복호화합니다.

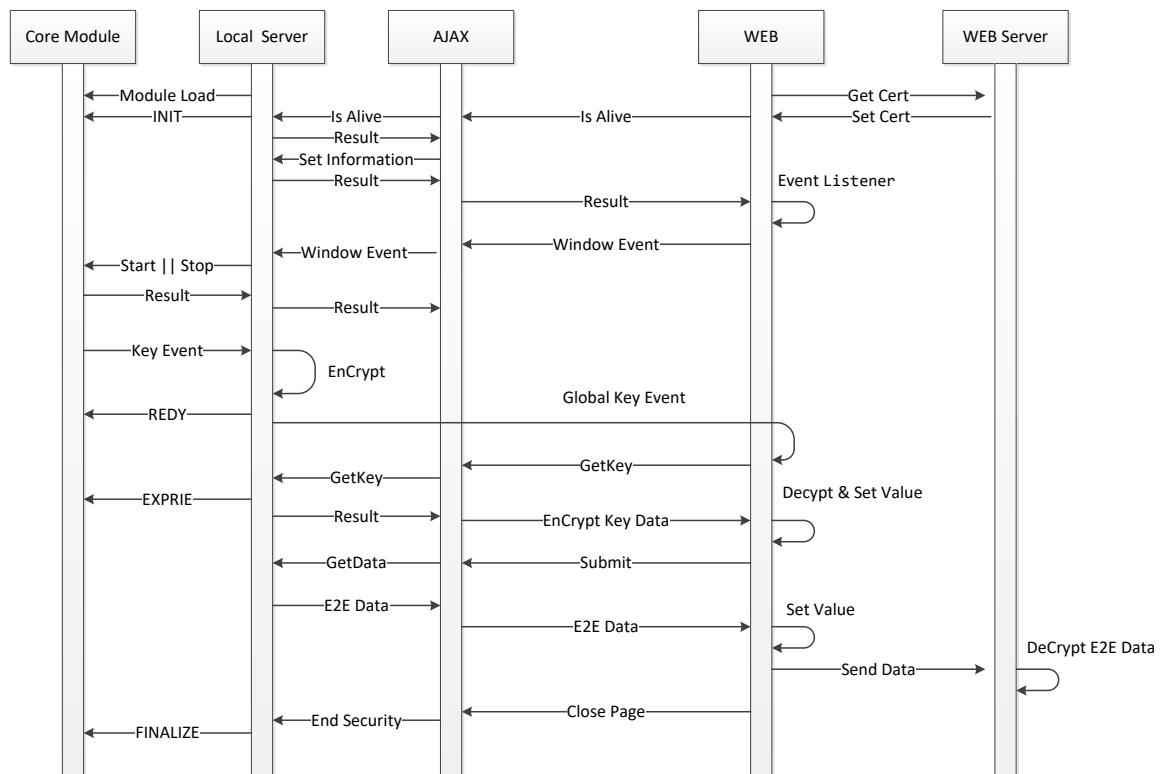
- WAS 서버의 E2E 데이터 검증

아래의 조건에 1 개라도 부합되지 않으면 WAS 서버에서 예외발생

1. 서버의 Http Session 에 개인키의 존재여부 확인

2. 서버의 Http Session 에 서버랜덤키 존재여부 확인
3. 클라이언트 랜덤키 존재여부 확인
4. 클라이언트 랜덤키가 개인키로 복호화 가능한지 확인
5. 서버 랜덤키와 클라이언트 랜덤키로 암복호화 키가 생성
6. 암복호화 키를 통해 전달된 데이터의 복호화 여부 확인
7. 입력양식의 데이터길이와 복호화된 E2E 데이터의 데이터길이의 일치여부 확인

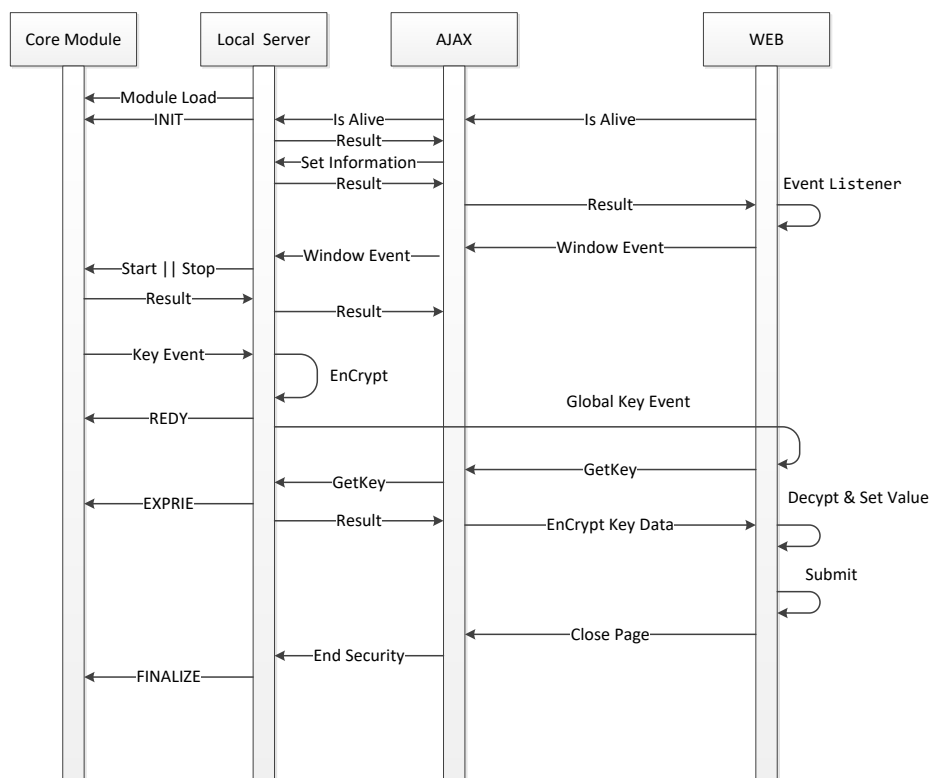
1.2.1.4. 키보드보안 흐름도(일반 키보드보안 / 확장 E2E 키보드보안)



[그림 1-5] 확장 E2E 키보드 보안 흐름도

1. NOS 가 설치 된 후 NOS 가 연동되어 있는 페이지 접속 시 보호 준비를 진행함
2. WAS 서버에 암호화 된 RSA 공개키 요청
3. NOS 모듈 내부에서 확장 E2E 의 KEY GEN 및 RSA 암호화 진행
4. 공개키로 암호화된 ARIA256 KEY 를 웹페이지로 전송하여 Hidden 필드에 저장
5. NOS 는 보호 시작을 디바이스 드라이버로 전달
6. OS Kernel Level 의 디바이스 드라이버를 사용하여 OS Kernel 과 외부 프로그램 사이 구간에 대한 보호를 진행
7. 사용자가 키보드를 이용하여 키 데이터 입력

8. 사용자가 입력한 키입력 데이터를 디바이스 드라이버에서 ARIA256 를 이용하여 암호화한 후 NOS 에 전달
9. 윈도우의 전체에서 키보드 입력 여부를 확인 할 수 없음으로 가상키 이벤트 생성 (숫자"1", 문자"A")
- 10.가상키 이벤트 발생
- 11.윈도우에서 키보드 입력 이벤트를 웹 브라우저에 발생
- 12.받은 이벤트를 NOS 에 전달받고 디바이스 드라이버에서 전달받은 암호화된 키 데이터를 복호화
- 13.복호화 된 키가 정상적인 키로 확인 될 시 확장 E2E 용 ARIA256 으로 암호화, 더미 키 값을 AES256 으로 암호화하여 웹 페이지로 전달
- 14.웹 페이지의 NOS 스크립트에서 AES 로 암호화 된 더미 값을 복호화 하여 필드에 저장 , 확장 E2E 용 ARIA256 으로 암호화된 실제 값은 Hidden 필드에 저장
- 15.사용자 Submit 시, 공개키로 암호화된 ARIA256 KEY, 필드별 Hidden 필드에 저장된 암호화된 데이터를 서버로 전송하여 WAS 에서 복호화 진행



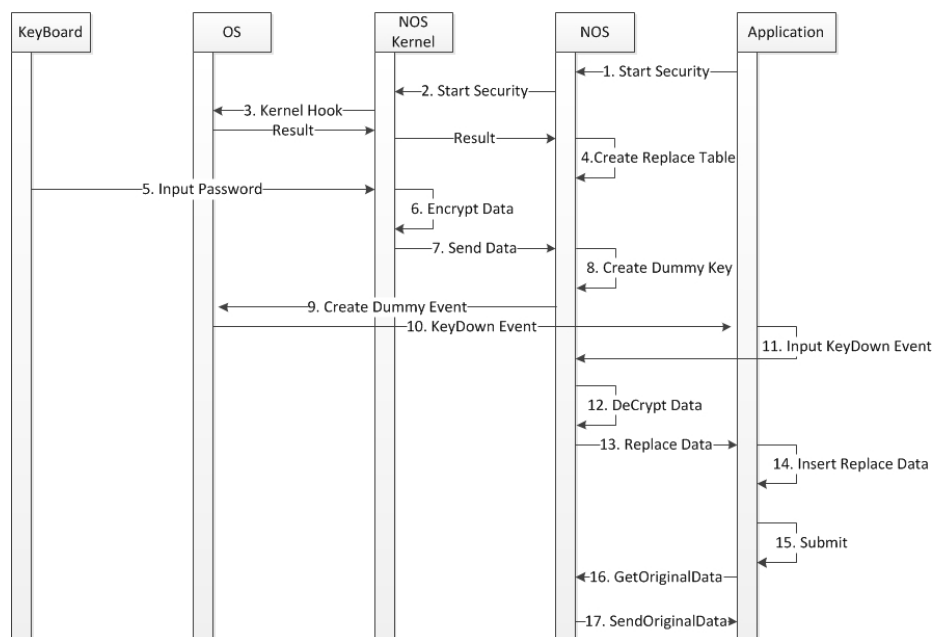
[그림 1-6] 일반 키보드 보안 흐름도

1. NOS 가 설치 된 후 NOS 가 연동되어 있는 페이지 접속 시 보호 준비를 진행함

2. NOS 는 보호 시작을 디바이스 드라이버로 전달
3. OS Kernel Level 의 디바이스 드라이버를 사용하여 OS Kernel 과 외부 프로그램 사이 구간에 대한 보호를 진행
4. 사용자가 키보드를 이용하여 키 데이터 입력
5. 사용자가 입력한 키입력 데이터를 디바이스 드라이버에서 ARIA256 를 이용하여 암호화한 후 NOS 에 전달
6. 윈도우의 전체에서 키보드 입력 여부를 확인 할 수 없음으로 가상키 이벤트 생성 (숫자"1", 문자"A")
7. 가상키 이벤트 발생
8. 윈도우에서 키보드 입력 이벤트를 웹 브라우저에 발생
9. 받은 이벤트를 NOS 에 전달받고 디바이스 드라이버에서 전달받은 암호화된 키 데이터를 복호화
- 10.복호화 된 키가 정상적인 키로 확인 될 시 AES256 으로 암호화하여 웹 페이지로 전달
- 11.웹 페이지의 NOS 스크립트에서 해당 AES 로 암호화 된 값을 복호화 하여 TEXT, PASSWORD Field 에 저장

1.2.1.5. 외부 프로그램 연동 흐름도(SDK 형태)

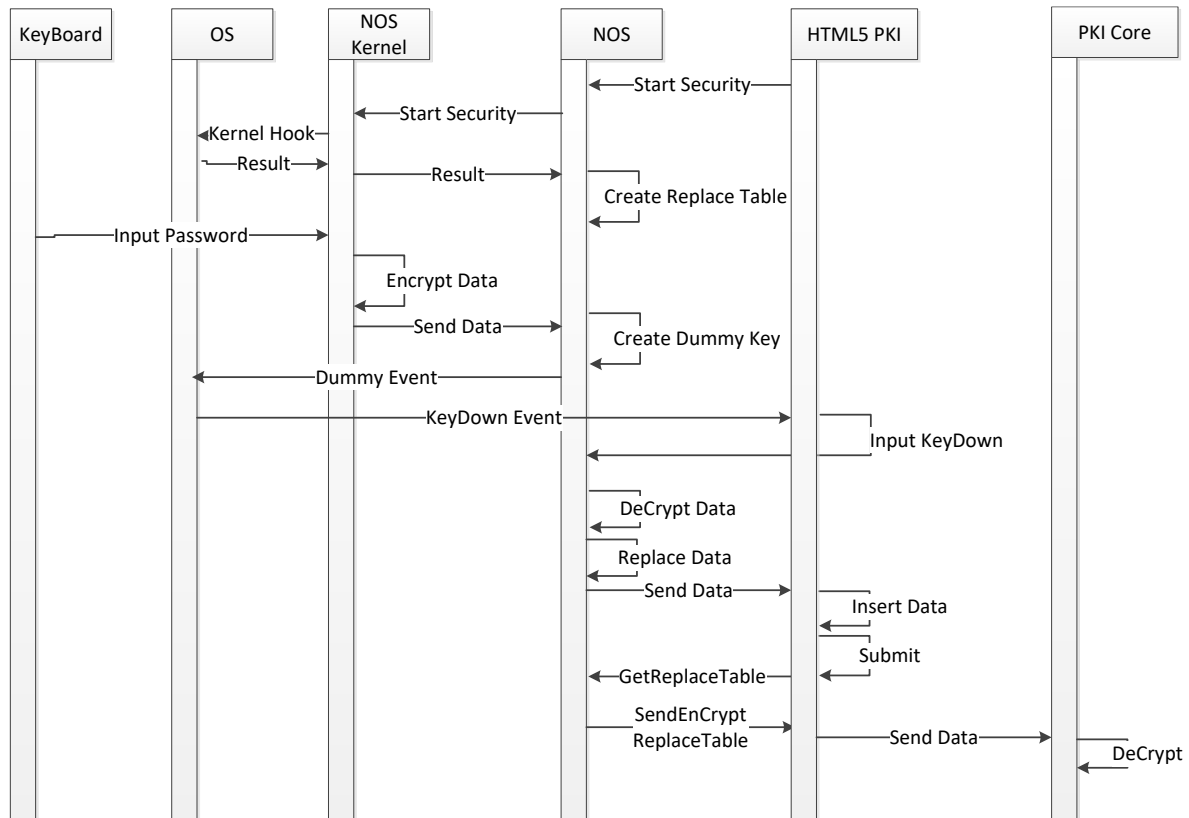
- 외부프로그램(예시: PKI)과 키보드보안 S/W(이하 NOS)의 연동 후 키보드이벤트 보호 흐름도



[그림 1-7] 외부 프로그램 연동 흐름도

1. NOS 를 연동한 외부 프로그램에서 활성화와 동시에 자동으로 보호 시작을 알림
2. NOS 에서는 보호 시작을 디바이스 드라이버로 전달
3. OS Kernel Level 의 디바이스 드라이버를 사용하여 OS Kernel 과 외부 프로그램 사이 구간에 대한 보호를 진행
4. 외부 프로그램 전용 1 회용 치환 테이블 생성
5. 사용자가 키보드를 이용하여 비밀번호 입력
6. 디바이스 드라이버에서 키를 받아 ARIA256 암호화
7. 암호화된 키 데이터를 NOS 에 전송
8. 윈도우의 전체에서 키보드 입력 여부를 확인 할 수 없음으로 가상키 이벤트 생성 (숫자"1", 문자"A")
9. 가상키 이벤트 발생
- 10.윈도우에서 키보드 입력 이벤트를 PKI 프로그램에 발생
- 11.외부 프로그램의 Password Edit 에 가상 이벤트 전달 받음
- 12.11 번에서 받은 이벤트를 NOS 에 전달받고 디바이스 드라이버에서 전달받은 암호화된 키 데이터를 복호화
- 13.복호화된 키 데이터를 사전에 제작된 치환테이블과 매칭하여 데이터 변경
- 14.변환된 치환 데이터를 외부 프로그램의 Password 입력 Edit 에 전달.
(Password 입력 Edit 에 치환 데이터 삽입(Edit 에는 치환된 임의 문자열 존재)
- 15.외부 프로그램 내에서 "확인" 클릭
- 16.Edit 에 입력된 치환데이터를 NOS 에 전달하여 실제 입력한 값으로 변경

1.2.1.6. HTML5 UI 형태의 PKI 연동



[그림 1-8] HTML5 UI형태 PKI 연동 키보드 보안 흐름도

1. NOS 가 설치 된 후 NOS 가 연동되어 있는 페이지 접속 시 보호 준비를 진행
2. NOS 는 보호 시작을 디바이스 드라이버로 전달
3. OS Kernel Level 의 디바이스 드라이버를 사용하여 OS Kernel 과 외부 프로그램 사이 구간에 대한 보호를 진행
4. PKI 에서 사용될 치환 테이블 생성
5. 사용자가 키보드를 이용하여 키 데이터 입력
6. 사용자가 입력한 키 입력 데이터를 디바이스 드라이버에서 ARIA256 를 이용하여 암호화한 후 NOS 에 전달
7. 윈도우의 전체에서 키보드 입력 여부를 확인 할 수 없음으로 가상키 이벤트 생성 (숫자"1", 문자"A")
8. 가상키 이벤트 발생
9. 윈도우에서 키보드 입력 이벤트를 웹 브라우저에 발생
10. 받은 이벤트를 NOS 에 전달하고 디바이스 드라이버에서 전달받은 암호화된 키 데이터를 복호화
11. 치환 테이블을 이용하여 데이터 치환

- 12.치환된 데이터를 AES256 으로 암호화하여 웹 페이지로 전달
- 13.웹 페이지의 NOS 스크립트에서 해당 AES 로 암호화 된 값을 복호화 하여 PASSWORD Field 에 저장
- 14.Submit 시 Password 필드에 저장되어 있는 치환 데이터 및 NOS 에 암호화된 치환 테이블 요청하여 두 값을 이용하여 데이터 복호화 진행

1.2.2. 방화벽

1.2.2.1. 기능 설명

방화벽은 고객PC에서 외부 서버에 접근하는 프로그램을 허용/차단 할 수 있는 기능 및 IP/Port 차단, 공유폴더 차단, 핑 차단을 제공합니다.

1.2.2.2. 기대 효과

특정 프로그램이 외부로 패킷을 전송할 때, 탐지하여 사용자에게 알리고, 허용/차단 여부를 선택 함으로 써, 사용자는 안정적인 서비스를 제공 받을 수 있습니다.

1.2.2.3. 상세 기능 설명

- 프로세스 인증 : 프로그램이 실행이 된 후, 외부로 접속을 할 때, Detect 를 해서, 사용자 동의에 의해 허용/차단이 되는 기능입니다. 내부적으로 허용정책이 있어서, 고객사의 프로그램 및 운영체제 기본 프로그램등은 사용자 동의 없이, 자동으로 허용으로 가능합니다.
- IP/Port 차단 : 특정 IP/Port 접근하는 프로그램의 인터넷 사용을 제한 합니다.
- ICMP 차단(Ping 차단) : ICMP 패킷 즉 핑 패킷에 대해서 차단을 합니다.
- 공유폴더 차단 : 외부에서 들어오는 공유폴더 접근을 차단합니다.

1.2.3. 실시간 프로세스 감시

1.2.3.1. 기능 설명

실시간 프로세스 감시는 고객 PC에 실행된 프로세스에 대해 악성코드 진단 및 치료 기능을 제공합니다.

1.2.3.2. 기대 효과

악성코드가 실행되어 있거나, 실행될 때 이를 탐지하여 사용자에게 알리고 치료 할 수 있는 기능을 제공함으로써, 사용자는 안정적인 서비스를 제공 받을 수 있습니다.

1.2.3.3. 상세 기능 설명

- 프로세스 감시 : 프로그램이 실행된 후, 실행된 프로세스에 대해 악성코드 검사를 진행합니다. 실행된 프로세스 중 악성코드가 발견되면 사용자 동의하에 악성코드를 치료하는 기능입니다.

1.2.4. 단말정보수집(FDS)

1.2.4.1. 기능 설명

단말기정보수집은 고객 PC 의 환경정보를 수집하여, 비정상 거래 징후를 탐지하고 추적하는 이상징후 탐지 시스템의 RAW 데이터를 수집하여 제공합니다. 또한, 변조 또는 우회된 Real IP 주소 등의 네트워크 정보 뿐 아니라, 다양한 하드웨어 및 소프트웨어 정보도 수집합니다.

1.2.4.2. 기대 효과

단말기 정보 수집을 바탕으로 고객의 비정상 거래 발생 시 즉각적인 통보로 전자금융 사기 및 사고 사전 예방과 사고 발생시 사고 추적을 가능하게 합니다.

1.2.4.3. 수집 항목

- 하드웨어 : 하드디스크 시리얼, 모델번호 및 볼륨시리얼, 키보드타입, 메인보드 시리얼, ManufactureID 및 제품 ID, 연결된 USB 메모리갯수
- 소프트웨어 : 윈도우 방화벽 사용여부, 원격데스크탑 사용여부, 보안제품설치여부(CKKP, KCRYPT, KD, KSID, SCSK, AOS), IE 브라우저 버전, 운영체제 버전, 서비스팩 버전, 시스템로케일
- 네트워크 : IP 주소, MacAddress(물리/논리), NatIP, ProxyIP, ProxyServerIP, VpnIP, VpnServerIP, ProxyServer 사용여부, 원격데스크탑으로 접속시, 원격지 IP

1.2.5. 피싱방지(Anti-Phishing)

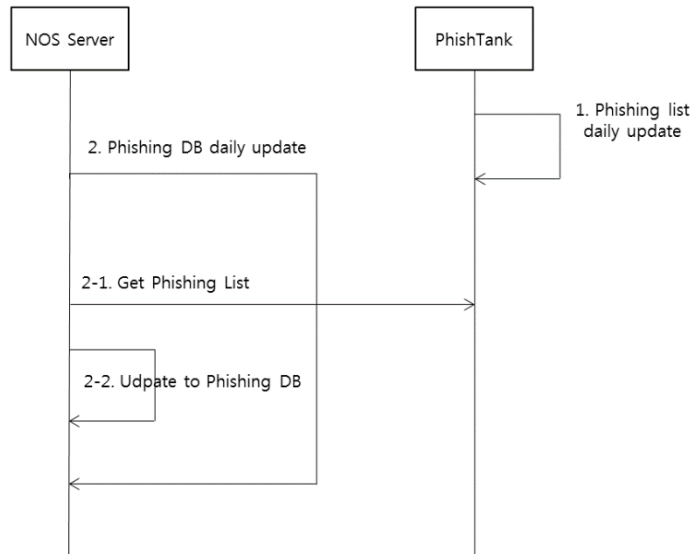
1.2.5.1. 기능 설명

피싱은 주로 금융기관, 정부 등의 이름으로 이메일을 발송, 변조된 웹사이트로 접속을 유도하여 개인정보를 탈취하는 사기수법 입니다. NOS 에서는 변조된 것으로 확인된 사이트 정보를 기준으로 해당 사이트 접속 시 경고 메시지를 발생시키는 방법으로 피싱을 방지합니다.

1.2.5.2. 기대 효과

변조된 사이트로 접속하는 것을 탐지하여 사용자에게 알리고 정상적인 사이트에 접속할 수 있도록 함으로써, 사용자는 안정적인 서비스를 제공받을 수 있습니다.

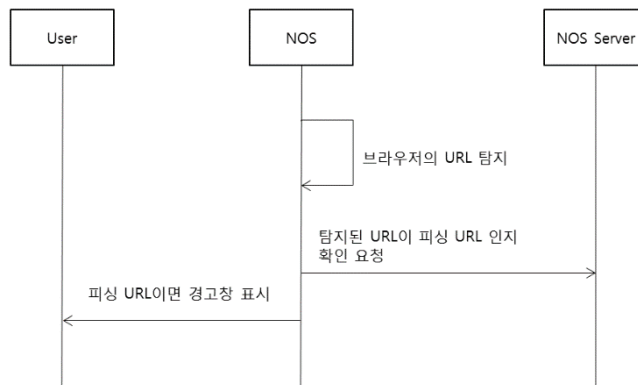
1.2.5.3. 처리 도식도(피싱 사이트 관리서버)



[그림 1-9] Phishing DB 갱신 흐름도

1. PhishTank 는 PhishTank 의 피싱 리스트를 매일 갱신하며 이를 파일 형태로 제공함
2. NOS Server 는 스케줄러에 의해 매일 정해진 시간에 PhishTank 의 피싱 리스트 파일 가져옴
3. 가져온 피싱 리스트 파일에서 유효 정보를 추출하여 NOS Server 의 Phishing DB 를 갱신함

1.2.5.4. 처리 도식도(피싱 방지 기능)



[그림 1-10] NOS 피싱 방지 기능 동작 흐름도

1. NOS 는 실행 중인 브라우저의 URL 을 탐지함
2. 탐지된 URL 이 피싱 URL 인지 NOS 서버에 확인 요청함
3. 확인 요청 결과가 피싱 URL 이면 사용자에게 경고창 표시

1.2.6. 파밍방지(Anti-Pharming)

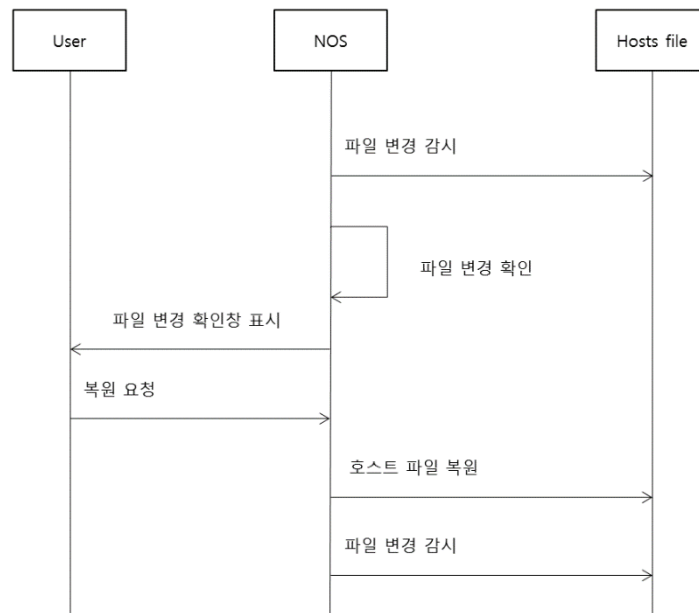
1.2.6.1. 기능 설명

파밍은 PC 이용자가 정확한 도메인 주소를 입력했는데도 불구하고, 변조된 사이트로 접속되게 하는 수법으로, 호스트 파일 변조, 도메인 탈취, DNS 서버 해킹 등의 파밍 유형이 있습니다. 그 중, 사용자 PC 해킹을 통한 Hosts File을 변경하여 변조된 사이트로 유도하는 것을 차단하는 기능을 제공합니다.

1.2.6.2. 기대 효과

사용자 PC에서 발생하는 도메인 변조를 미리 차단하여, 정상적인 사이트에 접속할 수 있도록 함으로써, 사용자는 안정적인 서비스를 제공받을 수 있습니다.

1.2.6.3. 처리 도식도



[그림 1-11] 호스트 파일 감시 흐름도

1. NOS 프로그램 시작 시 호스트 파일 감시 시작
2. 호스트 파일 변경이 발생한 경우, 사용자에게 파일 변경 확인 창을 표시
3. 사용자가 복원 요청 시 호스트 파일 복원
4. 호스트 파일 복원 후 파일 변경 감시 시작

1.2.7. 화면 캡처 방지

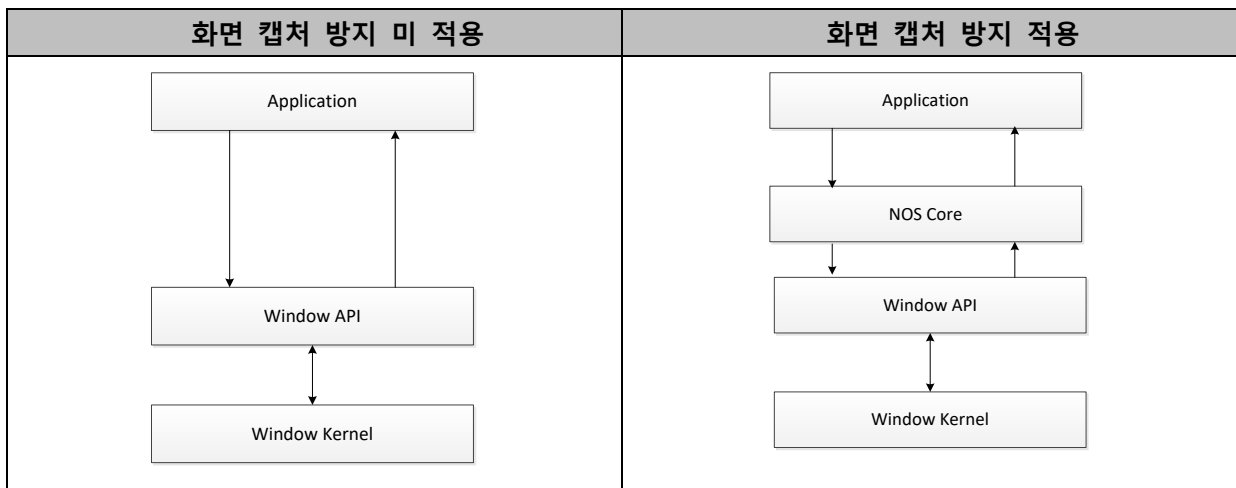
1.2.7.1. 기능 설명

사용자의 개인정보 및 암호화 되지 않는 정보를 표시하는 웹페이지 및 어플리케이션에서 사용자 정보보다 불법적인 스크린샷 및 윈도우의 특정 API를 통하여 화면 전체가 유출되어 개인정보를 유출 하는 악성 프로그램에 대하여 저항성을 높이고 차단 함으로써 사용자의 개인정보 보호를 할 수 있습니다.

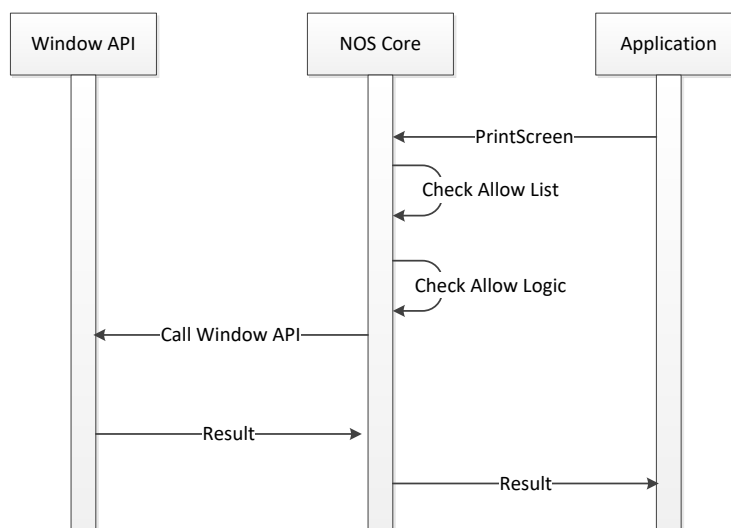
1.2.7.2. 기대 효과

- Windows API 이용 스크린 캡처 시도 차단
- PrintScreenKey 를 통한 스크린 캡처 시도 차단
- 스크린 캡처 동작 차단 시 이미지 교체 기능 지원
- 신종 캡처 프로그램에 대한 대응력 강화

1.2.7.3. 상세 기능 설명



[그림 1-12] 화면 캡처 방지 로직



[그림 1-13] 화면 캡처에 대한 방어 로직

1. NOS 는 모든 프로세스에 NOS Core 모듈을 인젝션한다.
2. 해당 프로세스의 window API 인 함수 사용을 모니터링 한다.
3. 해당 API 를 어플리케이션이 사용 할 때 해당 프로세스가 해당 API 를 사용여부를 판단한다.
4. 사용자 캡처 하려는 영역이 보호 영역과의 일치 여부 및 부분 겹침 등 내부 로직 확인 한다.
5. 보호 영역이 존재 할 경우 API 호출 차단

1.2.8. 마우스 입력기

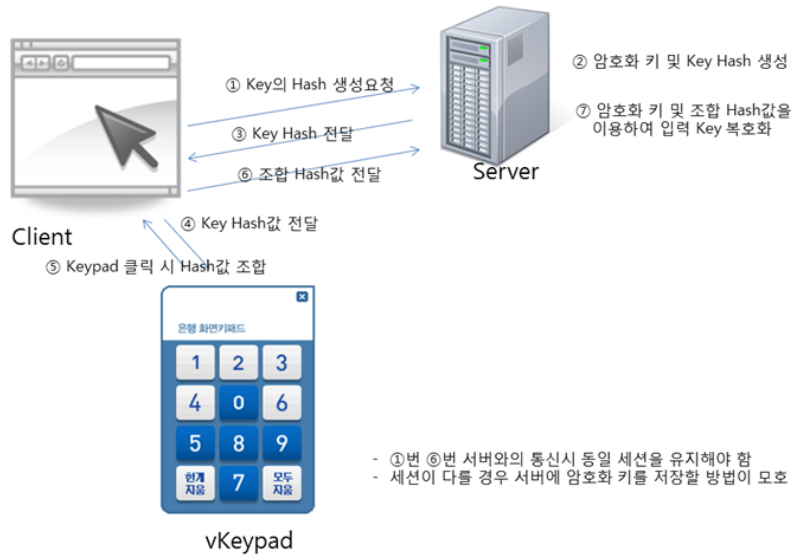
1.2.8.1. 기능 설명

인터넷 이체서비스(인터넷뱅킹, 사이버증권거래, 전자결재 등) 이용시 해커의 이체계좌 피싱기법에 의한 비 의도 계좌로의 이체를 방지하고 고객의 계좌 비밀번호 및 보안카드 비밀번호, 금융거래상 주요정보 유출을 해킹으로부터 보호의 하는 보안솔루션으로 개인, 인터넷 뱅킹 제공 은행, 전자결재서비스 제공업체 등에 필수적으로 적용해야 할 이체계좌, 계좌비밀번호, 보안카드, 로그인 암호의 변조를 방지합니다.

1.2.8.2. 기대 효과

- 주요 거래정보 키보드 보안 강화(숫자 및 모든 영문자 입력)
- 패스워드 입력 보안성 무결성
- 장애 없는 비밀번호 및 계좌번호 입력 체계
- OS 버전 및 IE 버전에 영향을 받지 않음
- 표준 인터넷 개발 플랫폼 준수(대부분의 웹브라우저에서 동작)
- Active-X 및 기타 Plug-In 모듈 불필요
- 드라이버 모듈 등 시스템에 설치되는 파일 없음
- 솔루션 도입에 따른 추가적인 비용 최소화(기존 시스템 사용, 하드웨어 추가 불필요)

1.2.8.3. 처리 도식도



[그림 1-14] 마우스입력기 처리 도식도

1. 클라이언트에서 마우스입력기를 사용하는 화면 생성할 때 Server 로 키발급을 요청합니다. 서버는 임의의 암호화키를 생성하고 암호화키를 HttpSession 에 저장합니다. 암호화키에 따라 각각의 키에 매핑되는 해쉬문자열(SHA-1)을 생성합니다. 생성된 해쉬문자열을 클라이언트에 응답값으로 전달합니다.
2. 클라이언트의 마우스입력기는 해당 해쉬문자열을 파싱하여 각 키값과 해쉬값을 매핑합니다. 사용자의 입력에 따라 해쉬값을 조합합니다.
3. 조합된 해쉬값을 서버에 다시 전달하여 값을 검증합니다.
4. 서버는 세션에 저장된 암호화키값을 얻어와서 다시 각각의 키에 매핑되는 해쉬문자열을 생성하고 전달받은 조합 해쉬값을 복호화 처리합니다.

- WAS 서버의 마우스입력기 데이터 검증

아래의 조건에 1 개라도 부합되지 않으면 WAS 서버에서 예외발생

1. 서버의 Http Session 에 암호화키의 존재여부 확인
2. 마우스입력기 정보(입력양식명, 폼 이름명, 마우스입력기 구분)의 전달여부 확인
3. 클라이언트로부터 전달된 조합된 해쉬값의 존재여부 확인
4. 입력양식의 데이터길이와 복호화된 마우스입력기 데이터의 데이터길이의 일치여부 확인

1.2.9. 구간암호화/보안전송

1.2.9.1. 기능 설명

웹 기반 프로토콜인 Http/Https 는 다양한 프록시 도구를 통하여 데이터가 노출될 수 있는 한계를 지니고 있으며 일부 프록시 도구는 Http를 통해 데이터가 송수신될 때 데이터의 가공을 가능하게 합니다. 이러한 데이터 가공은 원하지 않는 데이터를 수신하여 브라우저에 악성코드를 삽입하거나 서버에 침투할 수 있는 원인으로 작용할 수 있습니다. 본 기능은 네트워크 구간의 데이터를 암호화하여 악성 프록시 도구를 통해 데이터가 변조되는 것을 방지합니다. 전체적인 데이터는 서버에서 발급한 공개키 기반 알고리즘을 사용하여 서버 외에는 복호화가 불가능하도록 처리되어 암호화되어 전달되는 내용은 볼 수 있으나 변조는 불가능합니다.

1.2.9.2. 기대 효과

- 웹 브라우저와 서버간 종단간 암호화 방식(E2E)를 사용하여 중간단계에서의 복호화 무력화
- 웹 표준 규격에 따라 제작되어 웹 표준을 지원하는 다양한 브라우저와 운영체제에서 사용이 가능
- 표준 인터넷 개발 플랫폼 준수(대부분의 웹브라우저에서 동작)
- Active-X 및 기타 Plug-In 모듈 불필요
- 드라이버 모듈 등 시스템에 설치되는 파일 없음
- 솔루션 도입에 따른 추가적인 비용 최소화(기존 시스템 사용, 하드웨어 추가 불필요)

1.2.9.3. 처리 도식도



[그림 1-15] 구간암호화/보안전송 처리 도식도

1. 브라우저에서 암호화할 데이터가 있는 특정 웹페이지에 접속을 하면 자바스크립트의 Ajax 통신을 이용해서 서버에 공개키 발급을 요청합니다.

2. 서버는 서버 기동 시에 공개키/개인키 쌍을 생성하여 메모리에 저장하고 있다가 공개키 발급 요청이 오면 공개키를 암호화하여 브라우저에 응답합니다.
3. 브라우저는 수신된 공개키를 복호화하고 실제 송수신을 위한 암호화키를 발급합니다. 발급된 암호화키는 서버의 공개키를 암호화하여 웹페이지의 특정영역(form 의 input)에 저장합니다.
4. 브라우저에서 전달될 데이터는 발급된 암호화키를 가지고 암호화합니다.
5. 브라우저에서 서버로 데이터가 전송될 때 암호화된 발급 암호화키와 암호화 데이터를 전달합니다.
6. 서버에서는 개인키를 가지고 암호화된 발급 암호화키를 복호화하여 암호화키를 추출하고 추출된 암호화키를 이용하여 암호화 데이터를 복호화합니다. 이때 복호화된 데이터는 다시 Http Request 규격으로 재조립하여 서버에서의 다른 동작이 정상적으로 이루어지도록 처리합니다.

1.2.10. 무설치 단말정보 수집

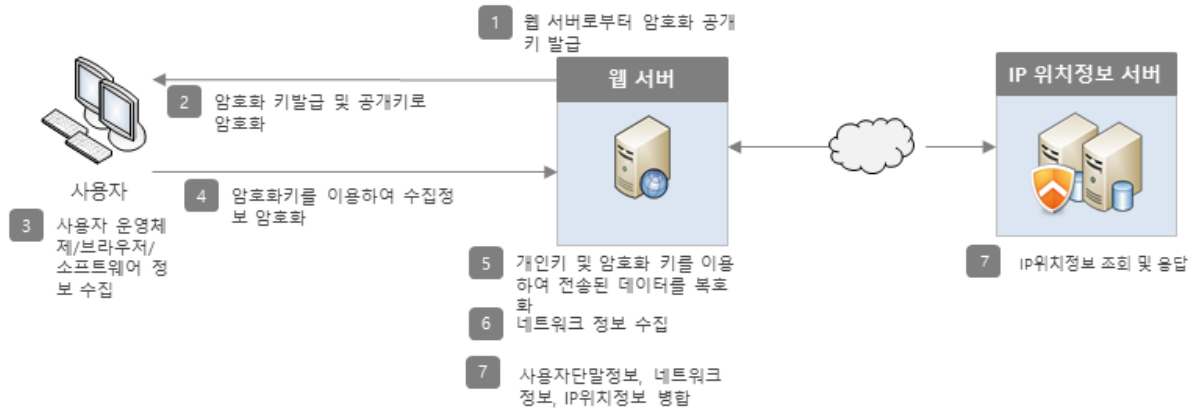
1.2.10.1. 기능 설명

인터넷 웹 이용 환경은 추가되는 프로그램(ActiveX, Plugin 등)없이 순수 브라우저 프로그램만을 통하여 이용될 수 있도록 변화되고 있습니다. 이런 이용환경에서 이상거래탐지를 위해 사용자 단말기를 식별하여 관련 정보를 수집할 수 있는 기술을 필요로 하게 됩니다. 본 기능은 추가프로그램 없이 사용자 단말장치를 식별하고 정보를 수집할 수 있는 기능을 제공합니다.

1.2.10.2. 기대 효과

- 웹 브라우저와 서버간 종단간 암호화 방식(E2E)를 사용하여 중간단계에서 데이터 변조 무력화
- 웹 표준 규격에 따라 제작되어 웹 표준을 지원하는 다양한 브라우저와 운영체제에서 사용 가능
- 특정 웹 페이지 또는 특정업무에서 선택적으로 적용하여 서버 부하를 최소화하여 적용 가능
- 표준 인터넷 개발 플랫폼 준수(대부분의 웹브라우저에서 동작)
- Active-X 및 기타 Plug-In 모듈 불필요
- 드라이버 모듈 등 시스템에 설치되는 파일 없음
- 솔루션 도입에 따른 추가적인 비용 최소화(기존 시스템 사용, 하드웨어 추가 불필요)

1.2.10.3. 처리 도식도



[그림 1-16] 무선랜 단말정보 수집 처리 도식도

1. 브라우저에서 이상거래판별을 위해 단말정보의 수집이 필요한 특정 웹페이지에 접속을 하면 자바스크립트의 Ajax 통신을 이용해서 서버에 공개키 발급을 요청합니다.
2. 서버는 서버 기동시에 공개키/개인키 쌍을 생성하여 메모리에 저장하고 있다가 공개키발급 요청이 오면 공개키를 암호화하여 브라우저에 응답합니다.
3. 브라우저는 수신된 공개키를 복호화하고 실제 송수신을 위한 암호화키를 발급합니다. 발급된 암호화키는 서버의 공개키를 암호화하여 웹페이지의 특정영역(form 의 input)에 저장합니다.
4. 공개키의 수신 및 처리가 완료되면 사용자의 브라우저/운영체제/네트워크 정보를 수집합니다.
5. 수집된 단말정보 데이터는 발급된 암호화키를 가지고 암호화합니다.
6. 브라우저에서 서버로 데이터가 전송될 때 암호화된 발급 암호화키와 암호화 데이터를 전달합니다.
7. 서버에서는 개인키를 가지고 암호화된 발급 암호화키를 복호화하여 암호화키를 추출하고 추출된 암호화키를 이용하여 암호화 데이터를 복호화합니다.
8. 추출한 데이터에서 IP 정보가 있을 경우에는 클라우드시스템 중 IP 위치정보 서버를 통하여 해당 IP의 위치정보를 요청합니다.
9. IP 위치정보 서버는 해당 IP 영역에 있는 위치정보를 검색하여 반환합니다.
10. 서버에서 위치정보와 복호화된 단말기정보 데이터를 병합합니다.

1.3. 암호화 알고리즘 및 키 생명주기

1.3.1. 암호화 알고리즘

알고리즘	용도	설명
RSA2048	키 교환용	키 교환을 위해 키 페어를 생성하기 위한 알고리즘 - WAS 기동시 RSA 2048 키페어 쌍 생성각 WAS마다 다른 RSA 키페어 쌍이 존재 - ARIA 대칭키 교환
ARIA 256 CBC	데이터 암/복호화용	블록 암호화 방식. 요청 시마다 Initial Vector를 임의로 생성 - 키보드 보안 정책파일 - 키보드 보안 드라이버 단 키 입력값 - 서버 랜덤키 암호화, 공개키 정보 암호화 - 키보드보안 모듈의 키입력데이터(서버전달용) 암/복호화
AES 128 CBC	데이터 암/복호화용	데이터 암/복호화를 위한 대칭키 암호화 알고리즘 - 접속 도메인 암호화 - 로컬 서버가 키보드 보안 JavaScript에 전송하는 모든 데이터 - 키보드보안 모듈의 키입력데이터(입력양식 노출용) 암/복호화를 위한 대칭키 암호화 알고리즘
RSA/ECB/ OAEPWITHSHA256A NDMGF1PADDING	키 정보 암/복호화	클라이언트 랜덤키 암/복호화하기 위한 알고리즘 - 클라이언트 랜덤키는 공개키로 암호화 - 서버에서 클라이언트 랜덤키를 개인키로 복호화
SHA 256	무결성 확인용	- 무결성 확인

[표 1-1] 암호화 알고리즘

1.3.2. 암호화 키 생명주기

암호화 키	생명주기
RSA 2048 키페어쌍	- WAS 기동 시 키페어쌍 생성 - WAS 재기동 전까지 유지
서버 랜덤키	- 1회성. 각 웹페이지 실행 시 서버에 요청할 때마다 생성
클라이언트 랜덤키	- 1회성. 각 웹페이지 실행 시 클라이언트에 요청할 때마다 생성
AES 128 CBC	- 1회성. 각 웹페이지 실행 시 클라이언트에서 생성

[표 1-2] 키 생명주기

2. 사업수행 계획

2.1. 사업개요

2.1.1. 사업명

- 보안 솔루션 구매 사업

2.1.2. 개요

- 시스템의 보안 신뢰성을 확보하기 위해 보안 서비스를 제공하고 안정적 서비스를 구축하고자 함

2.1.3. 목적

- 내부 직원 및 고객정보에 대한 해킹 위협으로부터 원천적 보호
- 보안 서비스를 통한 대외 이미지 제고
- 내부 직원 정보 유출 및 침해 등으로 인한 내재된 정보자산 손실방지
- 유연한 시스템 연계방안으로 내실 있는 정보보호 기반 마련

2.1.4. 주요 업무내용

- 서비스 제공방법 및 웹 구성 분석 업무
- 보안 솔루션 시스템 적용 및 구축 업무
- 보안 솔루션 시스템 테스트 업무
- 운영결과 분석 지원 업무
- 서비스 확대 구축 및 운영 지원 업무
- 교육 및 기술이전 지원 업무
- 지속적 유비보수 지원 업무

2.1.5. 수행 기간

- 개시일로부터 15 일 이내

2.1.6. 산출물

- 설치 CD 및 라이선스 1 부
- 납품완료보고서 3 부
- 납품장비별 매뉴얼 및 취급설명서 1 부
- 납품장비 및 설치과정 사진첩 1 부
- 시스템 구성도 1 부
- 위의 내용이 포함 된 CD 1 매

2.2. 투입인력 계획

2.2.1. 투입 인력 및 구성 개요 : 총 2명

- 사업총괄 : 1 명
- 시스템 구축 및 지원업무 : 1 명

2.2.2. 인력 구성 및 수행 업무

구분		담당자	수행 업무
키보드 보안	총괄 책임자		* 프로젝트 기술업무 총괄
	기술 지원		* 시스템 구축 방안 정립 및 설계 업무 * 시스템 구축 종합 관리 지원 업무 * 운영결과 분석 지원 업무 * 기술지원 및 교육지원 * 시스템 운영 지원업무 * 장애처리 분석 및 지원 업무

[표 2-1] 인력 구성 및 수행 업무

2.3. 수행일정 계획

2.3.1. 수행일정

구분	수행 기간 : 4주			
	1주	2주	3주	4주
사전검토 - 업무 담당자 구성 - 사전 적용업무 범위 선정 - 추진일정				
시스템 구축 업무협의 - 시스템 적용 대상 협의 - 보안 시스템 이해 및 적용 방법 등				
테스트 장비 및 환경 준비 - 테스트 장비확인 - 테스트 수행 환경 파악 등				
시스템 구축 - 보안 시스템 적용 구축 - 디자인 구성(Tray Icon & 웹 콘텐츠)				
단위 테스트(시범 서비스) - 시나리오 작성 협의 - 체크리스트 확인 결과				
결과 분석 보고 및 교육 지원 - 테스트 결과 분석 및 지원 업무 - 운영담당자 대상 기술 및 교육지원 - 산출물 작성 및 보고 등				
고객 대상 서비스 확대 - 웹 서버 확대 적용 및 서비스 확대 - 비상 대기반 운영 지원 - 검수완료 등				

[표 2-2] 수행일정

※ 해당 일정은 업무 협의 및 적용 페이지에 따라 단축될 수 있습니다.

2.3.2. 세부 계획

구분	내용	
[1단계] 사전검토 모듈준비	수행업무	* 사전 검토 - 보안 적용 대상 웹 페이지 사전 검토 및 정의 - 웹 프로그램 특이 사항 점검 및 사전환경 정의 - 총 프로젝트 일정 및 업무협의 시간 등 정의 - 협의 대상자 선정(웹 서버 운영 책임자) * 모듈 준비 - 사내 자체 모듈 테스트(자체 웹 페이지 적용) - 키 입력 테스트 - 보안기능(해킹방지) 테스트 - 무결성 테스트(입력 키와 서버에 전송된 키의 일치 여부 테스트) * 작업내역 - 고객사 : 홈페이지 소스 일부 수정 필요 - 잉카인터넷 : 고객사 용 모듈 개발
	일정	1주 (고객사의 사정 및 테스트시의 환경에 따라 변경될 수 있음)
[2단계] 시스템 구축 업무협의	수행업무	* 보안 시스템에 대한 이해 및 구축 일정 협의 * 보안 적용 웹 페이지 등 대상 확인 * 보안 서비스 형태 협의(Tray Icon 생성 및 이미지 등) * 서비스 운영방안에 대한 협의(웹 페이지 구성 FAQ, Q&A 등)
	일정	1주
[3단계] 테스트 장비 및 환경 준비	수행업무	* TEST 환경 구축 & 서버분리 작업 - 테스트 장비 및 테스트 서버 구축 - 테스트 페이지 선정 * TEST Preview - 모듈 탑재 예비 작업 - 적용 대상 웹 페이지 파악 & 적용 방법 협의 - 작업 내역 설명 및 보고 - 테스트 적용 * 작업내역 - 고객사 : E2E 테스트 환경 협의 및 구축 - 잉카인터넷 : E2E 테스트 환경 협의 및 구축
	일정	1주
[4단계] 시스템 구축	수행업무	* 웹 페이지 적용방법에 대한 이해 및 사전 교육 * Pilot 테스트 사이트 구축 - 고객사 내에서 Test 계 페이지 선정하여 Pilot 시스템 구축 - Pilot 시스템에서 키 입력 및 암호화 기초 테스트 * 특이 사항 분석 및 적용 * 디자인 구성 * 작업 내역 - 고객사 : 테스트 페이지 확정

		- 잉카인터넷 : 키보드보안 테스트 모듈 준비
	일정	1주
[5단계] 단위테스트	수행업무	* 테스트 방안 협의 * Pilot 시스템 테스트 - 보안 기능(해킹 방지) 테스트 - 무결성 테스트 - 각종 테스트 * 테스트 Checklist 점검 - 테스트 세부 Checklist 제출 - 세부 Checklist 를 기초로 심화 테스트 - 테스트 결과 분석 - 최종 수정 보완 - 수정 보완 내역 반영 후 Checklist 를 기초로 한 2 차 테스트 * 작업 내역 - 고객사 : 테스트 페이지 확장 및 내부 테스트 - 잉카인터넷 : * 키보드보안 테스트 모듈 테스트 적용 및 테스트 지원 * 테스트 세부 Checklist 제출
	일정	1주
[6단계] 결과 분석 보고 및 교육 지원	수행업무	* 최적화 작업 - 모듈 수정 보안사항 점검 - 최적화 작업(고객사 요구 사항 수렴) * Test 적용 - 실제 환경과 동일한 모든 필드에 적용(고객사 테스트) * 작업 가이드 작성 - 테스트 결과 보고서 - 사이트 구축 가이드 작성 * 작업 내역 - 고객사 : * 테스트 진행, 모듈 수정사항 업체 전달 및 확인 * 테스트 결과 분석 및 모듈 최종 점검 - 잉카인터넷 : * 테스트 결과 분석 및 모듈 최종 점검 * 테스트 결과 보고서 제출, 사이트 구축 가이드 제출
	일정	1주 (고객사의 사정 및 테스트시의 환경에 따라 변경될 수 있음)
[7단계] 고객 대상 서비스 확대	수행업무	* 실 서비스 시점 및 지원사항 협의 * 실 운영서버 시스템 적용 - 실제 서버에 적용 시작 - 부분 적용 후 확인 - 전체 적용 확대 - 적용 완료 및 안정화 작업 * 작업 내역 - 고객사 : 실 서버 E2E 적용 및 모니터링

		- 잉카인터넷 : * 모니터링 지원 * 모듈 안정화 / 고객지원
	일정	1주 (실 서버 적용 이후 서비스 안정화 단계까지)

[표 2-3] 인력 구성 및 수행 업무

※ 해당 일정은 업무 협의 및 적용 페이지에 따라 단축될 수 있습니다.

2.4. 교육계획

2.4.1. 설치 및 운영 교육

- 교육 대상자 : 웹 서버 운영자, 웹 개발자
- 교육 내용 : 제품 서버 적용 가이드에 대한 교육 (문서 내 서버 적용 가이드)

2.4.2. 장애유형 및 대응(응급)조치 교육

- 교육 대상자 : 웹 서버 운영, 웹 개발자
- 교육 내용: 제품의 FAQ 내용에 대한 교육 (문서 내 FAQ 항목)

3. 운영 환경

3.1. 사용자 운영 환경

3.1.1. 하드웨어 요구사항

[표 1-1]은 제품을 운영하기 위한 하드웨어 요구사항을 보여줍니다.

구분	권장사양	
하드웨어	CPU	Intel Pentium IV 3.0 GHz
	RAM	1 GB 이상
	Hard Disk	30 GB 이상
필수 입력 장치	PS/2, USB, Bluetooth Keyboard, 무선키보드 등 (HID 표준 스펙 준수)	
기타	MAC Boot Camp 4.1 이상 지원	

[표 3-1] 운영을 위한 하드웨어 요구사항

* 디바이스 드라이버 영역을 거치지 않는 키보드 입력 프로그램은 암호화 대상이 아님

3.1.2. 소프트웨어 요구사항

소프트웨어 요구사항은 크게 지원 운영체제 환경과 지원 브라우저로 나뉩니다.

3.1.2.1. Windows

구분	지원 현황
운영체제 (Windows)	Windows XP SP3 이상 (32bit)
	Windows Vista SP1 이상 (32/64bit)
	Windows 7 (32/64bit)
	Windows 8 (32/64bit)
	Windows 8.1 (32/64bit)
	Windows 10 (32/64bit)

[표 3-2] Windows 지원 운영체제 환경

웹 브라우저 버전		운영체제 구분					
		XP	Vista	7	8	8.1	10
Internet Explorer	7	○	○	해당버전 없음	해당버전 없음	해당버전 없음	해당버전 없음
	8	○	○	○	해당버전 없음	해당버전 없음	해당버전 없음
	9	해당버전 없음	○	○	해당버전 없음	해당버전 없음	해당버전 없음
	10	해당버전 없음	해당버전 없음	○	○	해당버전 없음	해당버전 없음
	11	해당버전 없음	해당버전 없음	○	해당버전 없음	○	○
Edge	20.10240.16384.0 ~ 38.14393.0.0	해당버전 없음	해당버전 없음	해당버전 없음	해당버전 없음	해당버전 없음	○
Safari	5.0	○	○	○	-	-	-
Opera	18.0 ~ 43.0	○	○	○	○	○	○
Firefox	21.0 ~ 52.0	○	○	○	○	○	○
Chrome	30.0 ~ 56.0	○	○	○	○	○	○
360 B	8.7.0	○	○	○	○	○	○
QQ	38.0 ~ 45.0	○	○	○	○	○	○
Yandex	15.2 ~ 16.10	○	○	○	○	○	○
Swing	1.3 ~ 3.4	○	○	○	○	○	○
Srware	46.0 ~ 54.0	○	○	○	○	○	○

[표 3-12] 브라우저 버전

* Internet explorer 7 버전의 경우 개별 웹 페이지에 정적으로 hidden field 생성 필요

* 붉은색으로 표시된 버전은 2017.03.21 기준 최신 버전을 뜻합니다.

* Safari 5.0 브라우저 공식 지원 운영환경은 Windows XP ~ Windows 7 입니다.

* Safari 브라우저 2012.5.9일 이후 윈도우 운영체제에 대해 지원 중단

3.1.2.2. Linux

구분	지원 현황
운영체제 (Linux)	Ubuntu 12.04~16.10
	Fedora 17 ~ 24

[표 3-2] Linux 지원 운영체제 환경

브라우저 버전		운영체제 구분	
		Ubuntu	Fedora
Firefox	21.0 이상	○	○
Chrome	30.0 이상	○	○

[표 3-12] 브라우저 버전

3.1.2.3. MAC

구분	지원 현황
운영체제 (MAC)	MAC OS X 10.8 Mountain Lion
	MAC OS X 10.9 Mavericks
	MAC OS X 10.10 Yosemite
	MAC OS X 10.11 El Capitan

[표 3-2] MAC 지원 운영체제 환경

브라우저 버전		운영체제 구분			
		OS X 10.8	OS X 10.9	OS X 10.10	OS X 10.11
Safari	6.0 이상	○	○	○	○
Opera	18.0 이상	○	○	○	○
Firefox	21.0 이상	○	○	○	○
Chrome	30.0 이상	○	○	○	○

[표 3-12] 브라우저 버전

3.2. 서버 운영 환경

3.2.1. 소프트웨어 요구사항

서버 소프트웨어 요구사항은 [표 1-1]과 같습니다.

구분	지원 현황
운영체제	Windows, Linux, Unix
WAS	Tomcat, WebLogic, Jeus, JBoss 등 (HttpServletRequest ² 기반 WAS)
Java	JDK 1.6 이상

[표 3-13] 서버 운영체제 및 지원환경

² 일반적으로 사용되는 기본 웹 객체입니다. 자체 프레임워크를 사용하거나 다른 웹객체를 사용하는 경우에는 별도의 맞춤형제작해야 하며 추가적인 비용이 발생할 수 있습니다.

4. 서버 적용 가이드

4.1. 라이브러리 설치³

4.1.1. jar 파일 복사

- 모든 jar 파일을 웹사이트 라이브러리 폴더(예, /WEB-INF/lib)에 복사합니다.
(WAS 시스템에 따라 라이브러리 경로가 다를 수 있으니 해당 라이브러리 경로에 복사합니다.)

4.1.2. /pluginfree/ 폴더 복사

- WEB, WAS 서버가 구분되어 있을 경우
 - /pluginfree/js/* 폴더를 WEB 서버의 Root 에 복사합니다.
 - /pluginfree/jsp/*, /pluginfree/examples/* 폴더를 WAS 서버의 Root 에 복사합니다.
- WEB, WAS 서버가 구분되어 있지 않은 경우
 - /pluginfree/ 폴더를 웹사이트 Root 에 복사합니다.

4.1.3. 마우스입력기 리소스 복사⁴

- WAS 서버의 WEB-INF 경로에 resources 폴더를 복사합니다.

4.1.4. 환경설정 파일 복사⁵

- WAS 서버의 WEB-INF/classes 경로에 nprotect.properties 파일을 복사합니다.

³ 일반 키보드보안만을 사용하는 경우에는 **jar 파일복사, jsp 파일복사, 마우스입력기 리소스복사, 환경설정 파일 복사, Filter 설정**의 과정이 필요 없습니다.(단, 일반 키보드보안 구동을 위한 자바스크립트(/pluginfree/js/*) 파일의 복사는 필요합니다.)

⁴ 마우스입력기를 사용할 경우에 이미지 및 이미지 정보를 포함하는 리소스입니다. 마우스입력기를 사용하지 않을 경우에는 복사하지 않아도 됩니다.

⁵ 본 파일은 복사하지 않아도 되며 파일이 없을 경우에는 마우스입력기를 사용할 수 없습니다. 기본값들은 [환경설정 파일 상세설정]을 참고하십시오.

4.1.5. nppfs-x.x.x.js의 수정

1. WAS의 환경에 따라서 WAS의 Root에 복사하였다고 하여도 실제 접속하는 경로는 다를 수 있습니다⁶. 이런 상태일 경우 아래의 구문을 nppfs-xxx.xxx.xxx.js의 하단에 추가합니다.

```
uV.dV.Gf = "<custom.path>/pluginfree/jsp/nppfs.key.jsp"; // 키발급 경로
uV.dV.zf = "<custom.path>/pluginfree/jsp/nppfs.remove.jsp"; // 키삭제 경로
uV.dV.zo = "<custom.path>/pluginfree/jsp/nppfs.keypad.jsp"; // 마우스입력기 페이지
uV.dV.eP = "<custom.path>/pluginfree/jsp/nppfs.ready.jsp"; // 초기화상태 확인경로
uV.dV.Fz = "<custom.path>/pluginfree/jsp/nppfs.install.jsp"; // 설치안내 페이지
```

4.2. 환경설정

4.2.1. Filter 기본 설정

1. WEB-INF/web.xml 파일을 열어 Filter 구문을 아래와 같이 추가합니다.
2. PropertiesPath는 nprotect.properties 파일의 경로이며 기본 /WEB-INF/classes의 경로가 아닐 경우 주석을 제거하고 절대경로로 지정합니다.

```
<filter>
    <filter-name>PluginFreeFilter</filter-name>
    <filter-class>com.nprotect.pluginfree.PluginFreeFilter</filter-class>
    <init-param>
        <param-name>PropertiesPath</param-name>
        <param-value>nprotect.properties</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>PluginFreeFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

4.2.1.1. Filter 고급 설정

1. RequestEncoding은 웹사이트의 페이지의 언어 인코딩을 설정합니다. 기본값을 UTF-8이며 웹사이트의 개발언어 인코딩 방식에 따라 지정해야 합니다. 일반적으로 한국어 페이지의

⁶ WAS의 인스턴스를 여러 개 생성하고 각각 Context 경로에 따라 인스턴스에 접근하는 경우 또는 도메인에 따라서 접근경로가 다를 경우 등입니다. 각각에 맞게 적절히 수정하십시오.

인코딩 값은 euc-kr 입니다. 특정 경로를 인코딩하지 않게 하려면 ExcludeEncoding 속성⁷을 추가하고 제외하려는 경로를 줄 단위로 입력합니다.

2. 복호화 페이지에 해당하는 경로에 필터를 매핑합니다.⁸ 특정 경로에 필터를 사용하지 않게 설정을 원할 경우에는 ExcludePattern 속성을 추가하고 값으로 제외하려는 경로를 줄 단위로 입력합니다.

```
<filter>
  <filter-name>PluginFreeFilter</filter-name>
  <filter-class>com.nprotect.pluginfree.PluginFreeFilter</filter-class>
  <init-param>
    <param-name>PropertiesPath</param-name>
    <param-value>nprotect.properties</param-value>
  </init-param>
  <init-param>
    <param-name>ExcludePattren</param-name>
    <param-value>
      /Appfree/jsp/*
      /appfreesign.do
      /appfreewebage.do
    </param-value>
  </init-param>

  <!-- 페이지 문자열 인코딩 설정 -->
  <init-param>
    <param-name>RequestEncoding</param-name>
    <param-value>UTF-8</param-value>
  </init-param>
  <init-param>
    <param-name>ExcludeEncoding</param-name>
    <param-value>
      </param-value>
    </param-value>
  </init-param>
</filter>
```

⁷ ExcludeEncoding 속성과 ExcludePattern 속성의 값 설정 시 * 문자의 사용이 가능합니다. 단, * 문자는 각 경로당 1 개만 사용가능합니다.

1. 경로가 * 로 시작하는 경우 *문자 뒷부분의 글자로 끝나는 경로가 설정됩니다.
2. 경로가 * 로 끝나는 경우 *문자 앞부분의 글자로 시작하는 경로가 설정됩니다.
3. 경로가 * 로 중간에 위치하는 경우 *문자 앞부분으로 시작하고 *문자 뒷부분의 글자로 끝나는 경루가 설정됩니다.

⁸ <filter-mapping>를 하지 않을 경우에는 PluginFreeRequest 객체를 사용하여 복호화가 가능합니다.

[nProtect Online Security 공통 – 업무페이지]의 수동복호화를 참고하십시오.

```

<!-- 특정한 페이지에 인코딩을 별도로 설정해야 할 경우 -->
<init-param>
    <param-name>PageCharset</param-name>
    <param-value>
        EUC-KR, EUC-JP
    </param-value>
</init-param>
<init-param>
    <param-name>PageCharset-EUC-KR</param-name>
    <param-value>
        /kr/*
    </param-value>
</init-param>
<init-param>
    <param-name>PageCharset-EUC-JP</param-name>
    <param-value>
        /jp/*
        *JP*
    </param-value>
</init-param>
</filter>
<filter-mapping>
    <filter-name>PluginFreeFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

```

● 환경설정(web.xml) 설정 옵션

환경설정 항목	값타입	설명	기본값
PropertiesPath	String	공통정책(nprotect.properties) 파일의 위치를 지정합니다. 값을 지정하지 않을 경우 /WEB-INF/에 있는 파일을 기본 참조합니다. 정상적으로 파일의 위치를 읽지 못할 경우에는 절대경로로 직접 지정합니다.	없음
ExcludePattren	String	특정페이지에 대해 필터를 사용하게 않게 할 경우 해당 페이지의 경로를 줄 단위로 지정합니다.	없음
RequestEncoding	String	웹 페이지에서 사용하는 문자열 인코딩이 있을 경우 문자열 인코딩을 위한 Charset을 지정합니다.	없음
ExcludeEncoding	String	특정 페이지에서 사용하는 문자열 인코딩이 처리를 원하지 않을 경우 해당 페이지의 경로를 줄 단위로 지정합니다.	없음
PageCharset	String	RequestEncoding으로 동작하지 않는 페이지가 있을 경우에는 사용하는 문자열 인코딩을 직접지정합니다.	없음

		사용하려는 Charset을 ,로 구분하여 지정합니다. ExcludeEncoding으로 지정한 페이지가 본 문자열 서정과 중복 될 경우에는 본 설정값은 동작하지 않습니다.	
PageCharset-[EUC-KR]	String	지정한 Charset이 동작하는 페이지의 경로를 줄 단위로 지정합니다. 이 속성의 이름은 PageCharset에서의 각 단위값과 공백을 제외하고 대소문자 모두 일치해야 합니다. 즉, PageCharset의 값이 EUC-KR이라면 속성값은 PageCharset-EUC-KR 입니다.	없음

[표 4-1] 환경설정(web.xml) 항목설명

4.2.1.2. Servlet 설정(필요시 설정)

1. ResponseEncoding 은 웹사이트의 페이지의 언어 인코딩을 설정합니다. 기본값을 UTF-8 이며 웹사이트의 개발언어 인코딩 방식에 따라 지정해야 합니다. 일반적으로 한국어 페이지의 인코딩 값은 euc-kr 입니다
2. PropertiesPath 는 nprotect.properties 파일의 경로이며 기본 /WEB-INF/classes 의 경로가 아닐 경우 주석을 제거하고 절대경로로 지정합니다.
3. Load-on-startup 항목에 서블릿의 적절한 순서를 지정하십시오. 숫자가 작은 순서대로 초기화되면 값이 같을 경우에는 상위에 선언된 서블릿부터 초기화됩니다. (실행되는 순서에 따라서 본 서블릿 또는 다른 서블릿의 동작이 정상적이지 않을 수 있으니 WAS 담당자 또는 서버관리자에게 문의한 후에 순서를 정하도록 합니다.)
4. Servlet-mapp 항목에 사용할 서블릿 URL 경로를 지정합니다. 초기화만을 위해서 사용하는 경우라면 본 항목을 주석 처리하도록 합니다.

```

<!-- nProtect Online Security, Plugin Free Filter : INCA Internet Co., Ltd. START -->
<servlet>
    <servlet-name>PluginFreeServlet</servlet-name>
    <servlet-class>com.nprotect.pluginfree.PluginFreeServlet</servlet-class>
    <init-param>
        <param-name>ResponseEncoding</param-name>
        <param-value>UTF-8</param-value>
    </init-param>
    <init-param>
        <param-name>PropertiesPath</param-name>
        <param-value>Some Path of Properties file\\nprotect.properties</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>PluginFreeServlet</servlet-name>

```

```

<url-pattern>/nppfs.servlet.do</url-pattern>
</servlet-mapping>
<!-- nProtect Online Security, Plugin Free Filter : INCA Internet Co., Ltd. END -->

```

4.2.1.3. Listener 설정(필요시 설정)⁹

1. WAS 에서 Listener 초기화시 PluginFree 를 참조할 경우 초기화 문제로 인해 오류가 발생할 수 있습니다. 아래의 코드를 삽입하여 Listener 에서 오류발생(NullPointerException)을 회피합니다. 먼저 선언된 순서로 초기화되어 다른 Listener 보다 상단에 위치하도록 합니다.¹⁰
2. PropertiesPath 는 nprotect.properties 파일의 경로이며 기본 /WEB-INF/classes 의 경로가 아닐 경우 주석을 제거하고 절대경로로 지정합니다.

```

<listener>
    <listener-class>com.nprotect.pluginfree.PluginFreeListener</listener-class>
</listener>
<!--
<context-param>
    <param-name>PropertiesPath</param-name>
    <param-value>Some Path of Properties file/nprotect.properties</param-value>
</context-param>
-->

```

4.2.2. 환경설정 파일 상세설정

1. WEB-INF/classes/nprotect.properties 파일을 열어 환경에 맞게 설정합니다.

```

# 모듈 검증 모드(true일 경우 로그출력)
verify.mode = false

# Redefine HttpServletRequest.getParameterValues()
# default : false
# true : use decrypt with getParameterValues()
use.parameter.values = true

# if the parameters(array of input name) at PluginFree.verify() function has not been specified, refer the
HttpServletRequest data to retrieve the input value form automatically.
# However, it can not fix the vulnerability, according to HttpServletRequest data modulation.

```

⁹ 서버모듈 1.3.0 버전 2016 년 05 월 25 일 이후의 모듈에서 사용 가능한 기능입니다.

¹⁰ Tomcat 8.0 WAS 기준으로 Listener > Filter > Servlet 순으로 초기화됩니다. Listener 에서 PluginFree 를 참조하는 경우 Filter 설정만 했을 경우에는 Listener 에서 설정값을 얻을 수 없어 예외가 발생합니다.

```

# default : false
# true : use intelligent search
# false : not use
use.intelligent.verify = true

# encryption spec
# default : ARIA256/CBC/PKCS5Padding
# ARIA256/CBC/PKCS5Padding, AES256/CBC/PKCS5Padding
interoperate.with.certificate.spec = AES256/CBC/PKCS5Padding
# encryption key for interoperate with certificate
interoperate.with.certificate.key =
d4219a6be6b839e34e73e52c9b2213c3d254fed02637abce605a887e1c393dde

# DEFAULT(default), HEADER
remote.address.type = DEFAULT
# DEFAULT(default, Plain Mode), 2ND(Encryption Mode)
natip.scribe.mode = 2ND
# DEFAULT(default, UAC Mode), 2ND(Administrator Mode)
hddserial.scribe.mode = DEFAULT
# true, false(default)
country.code.use = true
# country code url
country.code.url = http://report.nprotect.net:9080/nos/nppfs.country.jsp
# Timeout 3000(default)
country.code.timeout = 3000

ipchecker.url = http://ipc.nprotect.net/nppfs.message.jsp

# true, false(default)
location.info.use = false
# location information url
location.info.url = http://ipc.nprotect.net/nppfs.location.jsp

# 마우스입력기 사용여부. 스크립트의 true와 별도로 true 값이 아니면 동작하지 않음.
keypad.use = true
keypad.default.theme = default
keypad.resources.path = /WEB-INF/resources

# Image Synthesizer mode
# default : dynamic
# dynamic : create each request

```

```
# memory : cache in memory
#keypad.synthesizer.mode = dynamic
keypad.synthesizer.mode = memory
# max cache count
# default 0 : unlimited
keypad.synthesizer.max.cache.count = 1325

# Use Plain Alphabet Keypad(Disable replacement keys)
# default : false
# true : use plain key in textfield
# false : use replacement key in textfield
keypad.use.plain.text = true

# Use WAI-ARIA(Web Accessibility initiative-Accessible Rich Internet Applications)
# default : false
# true : enable ARIA
# false : diable ARIA
keypad.use.aria = true

# Use alert message
# default : false
# true : alert message disabled
# false : alert message enabled
keypad.disable.message = false

# Lastest Button Preview
keypad.preview.use = false

# default : match left coordinates of the input and the keypad.
# left : A specific pixel from the left of the window.
# right : A specific pixel from the right of the window.
# center : Place the keypad in the center of the window.
keypad.policy.position.x = center
keypad.policy.position.delta.x = 0

# default : match bottom coordinates of the input and top coordinates of the keypad.
# auto : The position of the keypad does not overlap with input places at the top or bottom.
#       After placing the keypad on the bottom, if the keypad is off the screen, place it on top.
#       Even if it was placed at the top of the outside of the screen and place the keypad on the
bottom.(default mode)
# top : A specific pixel from the top of the window.
# bottom : A specific pixel from the bottom of the window.
# middle : Place the keypad in the middle of the window.
```

```

keypad.policy.position.y = auto
keypad.policy.position.delta.y = 5

# 마우스입력기 엔터 버튼의 기본기능
keypad.policy.enter = HIDESELF

# keypad resize
# true, false(default)
keypad.resize = true
# percent of keypad width related browser width
keypad.resize.percent = 90

# display touch event(display dummy click button)
# default : false
keypad.display.touch = true
# background color
keypad.display.touch.color = #333333
# hide touch button
keypad.display.touch.timeout = 2000
# button opacity 0~100;
keypad.display.touch.opacity = 70

#####
# Block Cipher Options
#####
# <!-- NPPFS CIPHER BEGIN -->
# <!-- NPPFS CIPHER END -->
# default : false
cipher.block.use = false

# single / multi
# single - 1 cipher block
# multi - multi cipher block
# default : single
cipher.block.policy = multi

# max cipher block size, 0 is disable max length
cipher.block.max.size = 10240

```

● 환경설정 설정 옵션

환경설정 항목	값타입	설명	기본값
---------	-----	----	-----

verify.mode	boolean	모듈 검증 모드를 설정합니다. nProtect Online Security의 암호/복호화 및 정합성검증을 위한 설정입니다. true : System.out에 정합성 검증 로그를 출력합니다. false : 로그를 출력하지 않습니다.	false
use.parameter.values	boolean	request.getParameterValues() 함수에서 필터의 복호화로직을 수행할지 설정합니다. true : request.getParameterValues() 함수에서 복호화로직을 수행합니다. false : request.getParameterValues() 함수에서 복호화로직을 수행하지 않습니다.	false
use.intelligent.verify	boolean	PluginFree.verify() 함수에 지정하는 입력양식의 이름을 HttpRequest의 값들의 이름을 자동으로 검색하여 동작합니다. ¹¹ 암호화 값이 없으면 verify 항목에서 제외시킵니다. true : 자동으로 입력양식을 검색합니다. false : 기능을 사용하지 않습니다.	false
interoperate.with.certificate.spec	string	HTML방식의 외부 프로그램에서 마우스입력기 사용시 복호화 값을 특정한 알고리즘으로 재 암호화하기 위한 암호화 알고리즘을 지정합니다. - ARIA256/CBC/PKCS5Padding - AES256/CBC/PKCS5Padding	ARIA256/CBC/PKCS5Padding
interoperate.with.certificate.key	string	HTML방식의 외부 프로그램에서 마우스입력기 사용시 복호화 값을 특정한 알고리즘으로 재 암호화하기 위한 암호화 키를 지정합니다.	없음
keypad.use	string	마우스입력기 사용여부를 지정합니다. true : 마우스입력기를 사용합니다. false : 마우스입력기를 사용하지 않습니다.	false
keypad.default.theme	string	마우스입력기의 기본 테마를 지정합니다.	default
keypad.synthesizer.mode	string	마우스입력기의 동적이미지 생성을 메모리에 추가할지를 설정합니다. dynamic : 매번 동적으로 이미지를 생성합니다. memory : 메모리에 이미지를 저장하도 동일한 이미지 요청이 오면 메모리에서 바로 이미지를 합성하지 않고 바로 반환합니다.	dynamic
keypad.synthesizer.max.cache.count	number	최대로 메모리에 저장할 이미지의 개수를 지정합니다.	0(제한없음)
keypad.use.plain.text	boolean	마우스입력기에서 alpha형 키패드가 text 입력양식에	true

¹¹ 본 기능은 모든 경우의 입력양식을 검색하지 못할 수 있습니다. 즉, 브라우저에서 Request 값을 변조할 경우에는 입력양식의 이름을 검색하지 못할 수 있습니다. 본 기능의 사용은 추천하지 않으며 개별 페이지에 맞게 입력양식이름을 수동으로 지정하는 방식을 추천합니다.

		서 사용되는 경우 평문을 보여줄지 여부를 설정합니다. (보호하려는 값이 HTML DOM 문서에 노출이 되기 때문에 사용을 추천하지는 않습니다.)	
keypad.use.aria	boolean	웹 접근성 관련하여 스크린리더 기능의 프로그램을 통해 마우스입력기 버튼을 선택 시 해당 버튼을 읽어 줍니다. (보호하려는 값이 HTML DOM 문서에 노출이 되기 때문에 사용을 추천하지는 않습니다.)	false
keypad.disable.message	boolean	의도하지 않는 사용자 행동(예, 마우스입력기 강제사용이 필수인 환경에서 마우스입력기 사용을 해제하려 할 경우)에 따른 경고 문구를 보이거나 보이지 않게 합니다. true : 경고 문구를 보이지 않게 합니다. false : 경고 문구를 보이게 합니다.	false
keypad.preview.use	Boolean	마우스입력기로 입력하는 마지막 문자열을 보여줄지 설정합니다. true : 마지막 문자를 보여줍니다. false : 마지막 문자를 보여주지 않습니다.	false
keypad.resources.path	string	마우스입력기 리소스의 경로를 지정합니다.	/WEB-INF/resources
keypad.focus.mode	string	입력양식에 포커스 이벤트가 발생하였을 경우 추가적인 작업을 결정합니다. default : 아무런 작업도 수행하지 않습니다. clear : 기존에 입력되었던 값들을 초기화합니다.	default
keypad.policy.enter	string	마우스입력기 엔터 버튼의 기본기능 SUBMIT(default) – 입력양식 데이터 Submit HIDEALL – 마우스입력기 모두 숨김 HIDESELF – 해당 마우스입력기 숨김	SUBMIT
keypad.policy.position.x	string	마우스입력기 X좌표 위치 정책을 설정합니다. default : 입력양식의 왼쪽에 위치시킵니다. left : 보여지는 창의 왼쪽에 위치시킵니다. right : 보여지는 창의 오른쪽에 위치시킵니다. center : 보여지는 창의 가운데에 위치시킵니다.	default
keypad.policy.position.delta.x	integer	X위치 정책에 따라 각각의 Y좌표를 조정합니다. default : 입력양식으로부터의 상대좌표 left : 창의 왼쪽으로부터 떨어뜨릴 좌표 right : 창의 오른쪽으로부터 떨어뜨릴 좌표 center : 창의 가운데에서 떨어뜨릴 상대좌표	0
keypad.policy.position.y	string	마우스입력기 Y좌표 위치 정책을 설정합니다. default : 입력양식의 하단에 위치시킵니다. top : 보여지는 창의 상단에 위치시킵니다. bottom : 보여지는 창의 하단에 위치시킵니다. middle : 보여지는 창의 중간에 위치시킵니다. auto : 입력양식의 하단에 위치하려고 계산 후 보여	default

		지는 창을 넘어가면 입력양식의 상단 위치시킵니다. 상단에 위치시키려고 할 때 보여지는 창을 넘어가면 강제로 입력양식의 하단에 위치시킵니다.	
keypad.policy.position.delta.x	integer	Y위치 정책에 따라 각각의 Y좌표를 조정합니다. default : 입력양식으로부터의 상대좌표 top : 창의 상단으로부터 떨어뜨릴 좌표 bottom : 창의 하단으로부터 떨어뜨릴 좌표 middle : 창의 가운데에서 떨어뜨릴 상대좌표 auto : 입력양식으로부터의 상대좌표.	0
keypad.resize ¹²	boolean	마우스입력기의 크기조절 기능 활성화 true : 크기조절 기능 활성화(사용) false : 크기조절 기능 비활성화(사용안함)	false
keypad.resize.ppercent	integer	마우스입력기의 폭과 브라우저 폭의 상대적 비율 퍼센트 값. 설정값이 90이고 키패드의 폭이 100px라면 화면 폭이 111px이하일 경우 키패드가 크기조절 됨 (100px / (90/100) = 111px)	90
keypad.display.touch	boolean	임의 버튼 클릭 기능 사용여부	false
keypad.display.touch.color	string	버튼의 배경색상	#888888
keypad.display.touch.timeout	integer	임의 버튼의 노출시간	1000
keypad.display.touch.opacity	integer	임의 버튼의 투명도(0~100) 0 : 완전 투명, 100 : 완전 불투명	50
keypad.display.touch.mode	string	더미버튼 효과 모드 default : 터치한 버튼과 더미버튼에 효과 적용 single : 터치한 버튼만 효과적용	default
remote.address.type	enum	FDS 단말정보 수집시 원격지 IP를 얻어오는 방법을 지정합니다. DEFAULT : request.getRemoteAddress() 함수를 사용하여 IP주소를 얻어옵니다. HEADER : Http 헤더에서 IP가 존재하는지 확인 후 값을 얻어옵니다.	DEFAULT
natip.scribe.mode	string	NAT IP 수집방법을 정의합니다. DEFAULT : 기존방식 호환용으로 비 암호화된 IP를 수집합니다. 2ND : NAT IP 수집 시 수집정보가 암호화되어 처리됩니다.	DEFAULT
hddserial.scribe.mode	string	하드디스크 시리얼 수집방법을 정의합니다. Windows	DEFAULT

¹² 마우스입력기가 크기조절될 경우에는 화면 노출옵션이 가운데(keypad.policy.position.x = center) 및 상대좌표는 0px(keypad.policy.position.delta.x = 0)로 자동 변경됩니다. 모바일 브라우저의 경우 세로보기 일 경우 지정한 비율값으로 무조건 확대/축소되어 처리되며 PC 및 모바일 가로보기 일 경우 해상도 범위를 벗어나는 경우에만 축소되도록 처리됩니다.

		8/Windows 8.1에서는 UAC권한에 따라서 다른 값이 나오는 경우가 발생합니다. DEFAULT : UAC권한에 따라 수집방법을 달리합니다. 기존 배포된 고객사는 호환을 위해 기본값을 사용해야 합니다. 2ND : 관리자권한으로 수집된 정보를 수집합니다.	
country.code.use	boolean	국가코드 판단기능을 사용하는지 지정합니다. 본 기능 사용시 WAS서버에서 국가코드판단 서버 사이의 네트워크 구간에 대한 접속(예, 방화벽 해제)을 확인해야 합니다.	false
country.code.url	string	국가코드 판단 URL경로를 입력합니다.	null
country.code.timeout	Integer	국가코드 판단 URL호출시 대기시간을 지정합니다.	3000
ipchecker.url	string	무설치단말정보 IP 수집 경로를 입력합니다.	null
location.info.use	boolean	위도, 경도 정보를 사용하는지 지정합니다. 본 기능 사용시 WAS서버에서 위도, 경도 정보 서버 사이의 네트워크 구간에 대한 접속(예, 방화벽 해제)을 확인해야 합니다.	false
location.info.url	string	위도, 경도 정보를 읽어오는 URL경로를 입력합니다.	null
cipher.block.use	boolean	HTML의 특정 블록에 응답값 암호화를 적용할지를 설정합니다. 블록은 다음의 키워드로 지정되어야 합니다. ¹³ 블록 시작 : <!-- NPPFS CIPHER BEGIN --> 블록 끝 : <!-- NPPFS CIPHER END --> true : 응답값 암호화를 사용합니다. false : 응답값 암호화를 사용하지 않습니다.	false
cipher.block.policy	string	단일 또는 여러 개의 블록을 암호화 처리할지 지정합니다. single : 처음에 나타나는 단일 블록을 암호화시킵니다. multi : 여러 개의 블록을 암호화시킵니다.	single
cipher.block.max.size	integer	응답처리의 최대 길이를 설정합니다.	0

[표 4-2] 환경설정(nprotect.properties) 항목설명

¹³ HttpServletResponse 응답값 암호화를 사용하는 경우 암호화가 반드시 필요한 경우에만 블록으로 지정합니다. 브라우저(Microsoft Internet Explorer 7,8)의 종류 및 PC 사양에 따라 브라우저에서의 처리 속도가 현저히 낮아질 수 있습니다.

4.3. 페이지 적용

4.3.1. 문서의 HTML 표준 확인¹⁴

1. 브라우저에서는 표준 HTML 규격에 부합하지 않는 페이지에서는 정상적으로 동작하지 않을 수 있습니다.¹⁵ 페이지를 문서버전에 따라서 아래와 같이 표준에 맞게 먼저 설정합니다.

문서 버전	내용
HTML 5	<!DOCTYPE html>
HTML 4.01	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"> <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
XHTML 1.0	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

[표 4-3] 환경설정 항목설명

2. Microsoft Internet Explorer 브라우저에서 최신 엔진으로 문서를 처리할 수 있도록 다음을 구문을 <head></head> 사이에 추가합니다.

```
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
```

3. HTML 문서가 W3C 표준규격에 부합하는지 아래의 웹사이트 또는 도구를 통하여 확인합니다.

¹⁴ 쿼크모드(Quirks mode)와 표준모드(Standard mode) 그리고 준표준모드(Almost standard mode)

브라우저는 선언된 doctype 에 따라 렌더링할 모드를 선택합니다. 브라우저가 출력하고자 하는 문서가 최신이 라고 판단하면 표준모드(standard mode)로 렌더링을 하는데 CSS2 스펙에 따라 CSS 가 적용되었다는 것을 의미하며 브라우저가 예전 문서라고 판단을 하면 쿼크 모드(quirks mode)로 렌더링하고 이전 세대의 브라우저에 맞는 비표준적 방법의 CSS 를 적용합니다. 그리고, *Opera(7.5 이상), Firefox, 그리고 Safari 는 위에 언급된 두 가지 모드 외에 almost standard mode 라고 불리는 하나의 모드를 더 지원하고 있는데, 이 모드일 경우 테이블 셀 안에 있는 이미지 레이아웃은 CSS2 의 스펙을 따르지 않고 퀵스 모드로 구현되어 있다. 이런 부분을 제외하고는 표준모드와 동일합니다.

¹⁵ 쿼크모드로 동작할 경우 nProtect Online Security 보안프로그램이 정상동작하지 않을 수 있습니다.

Microsoft Internet Explorer 의 경우에는 쿼크모드일 경우 문서처리를 하위 브라우저(IE 7.0 미만), 하위 자바스크립트 엔진(Javascript version 5.6 이하)을 사용하여 처리됩니다.

참고 웹사이트 : <http://en.wikipedia.org/wiki/JScript>

사이트/도구명	설명
W3C Markup Validation Service 마크업(문서) 검사 서비스 - W3C	http://validator.w3.org/ (영문사이트) http://validator.kldp.org/ (국내사이트) URL 입력 - 사이트 주소 입력 파일 업로드 - 컴퓨터 내 작업 HTML 파일을 업로드 직접 입력 - HTML문서 내용을 넣어 확인
W3C CSS Validation Service 케스케이딩 스타일시트 검사 서비스 - W3C	http://jigsaw.w3.org/css-validator/ (한글번역사이트) http://css-validator.kldp.org/ (국내사이트) URL입력 - HTML과 CSS 또는 CSS 문서의 주소 입력 파일 업로드 - 컴퓨터 내 작업 CSS파일을 업로드 직접 입력 - CSS문서 내용을 넣어 확인
K-WAH 4.0 카도와 4.0 - 웹 접근성 연구소	프로그램 다운로드 주소 http://www.wah.or.kr/ [지식마당 > 개발자아카이브 > K-WAH 4.0] 으로 이동 페이지에서 [K-WAH 4.0 프로그램 내려받기] 버튼 클릭
N-WAX(NHN Web Accessibility eXtension) NULI / NHN	웹 브라우저 파이어폭스의 부가 기능 플러그인으로 접근성 문제를 진단하고 수정 도구 Firefox → 부가 기능 → 부가 기능 관리자 페이지 내 [N-WAX] 검색 → [N-WAX] 설치

[표 4-4] 웹 접근성 검사 사이트/도구

● DTD Switching 에 따른 웹 브라우저별 렌더링 모드

구분	내용	IE	FF	OP	SF
None	DTD가 없는 경우	쿼크	쿼크	쿼크	쿼크
HTML 5	<!DOCTYPE html>	표준	표준	표준	표준
HTML 4.01	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">	쿼크	쿼크	쿼크	쿼크
	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">	준표준	준표준	준표준	준표준
	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">	준표준	표준	표준	표준
	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">	준표준	표준	표준	표준
XHTML 1.0	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">	준표준	준표준	준표준	준표준
	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">	준표준	표준	표준	표준

[표 4-5] DTD Switching에 따른 웹 브라우저별 렌더링 모드 전환

4.3.2. nProtect Online Security 공통

1. 페이지 상단에 jquery-xx.xx.xx.js 파일을 포함시킵니다. 이미 사용중인 jQuery 라이브러리가 있다면 추가하지 않아도 됩니다. 단, jQuery 1.7.4 이전 라이브러리를 사용한다면 상위버전으로 업그레이드 하십시오. (일부 기능이 지원되지 않아 정상적으로 동작하지 않을 수 있습니다.)
(일반적으로 html 의 `<head>` `</head>` 사이에 위치합니다.)

```
<script type="text/javascript" src="/pluginfree/js/jquery.min-xx.xx.xx.js"> </script>
```

2. 페이지 상단에 nppfs.script.jsp 파일을 포함시킵니다.
(일반적으로 html 의 `<head>` `</head>` 사이에 위치합니다.)
<script> charset 은 웹사이트에 맞게 언어를(utf-8, euc-kr) 선택합니다.¹⁶

```
<script type="text/javascript" src="/pluginfree/jsp/nppfs.script.jsp" charset="utf-8"> </script>
```

3. 페이지 상단에 nppfs-1.0.0.js 파일을 포함시킵니다.
(일반적으로 html 의 `<head>` `</head>` 사이에 위치합니다.)
<script> charset 은 웹사이트에 맞게 언어를(utf-8, euc-kr) 선택합니다.

```
<script type="text/javascript" src="/pluginfree/js/nppfs-1.0.0.js" charset="utf-8"> </script>
```

4. 문서의 <body>, <form>에 div 노드를 추가합니다. 미리 추가해도 상관없으나 고객사 개발자의 작업이 수반되는 부분이므로 페이지의 특성에 따라 오류가 날 때만 추가하도록 합니다.

- <body>에 추가할 항목. 로딩이미지 표시
<div id="**nppfs-loading-modal**" style="display:none;"> </div>
- <form>에 추가할 항목. 키보드 보안 또는 단말수집정보를 사용하는 form
<div class="**nppfs-elements**" style="display:none;"> </div>

¹⁶ UTF-8 문서 EUC-KR 로 변경하기

1. 메모장 또는 텍스트 편집기에서 해당 파일을 엽니다.
2. [다른이름으로 저장]을 선택한 후 파일 포맷을 ANSI 로 선택합니다.
3. html 문서내용에 charset 설정이 utf-8 일 경우 euc-kr 로 변경합니다. (예 content-type="text/html; charset=**utf-8**)
4. jsp 파일일경우 최상단의 content-type 와 request-type, response-type 도 변경합니다.

- <form>에 추가할 항목. 마우스입력기를 사용하는 form
 <div class="**nppfs-keypad-div**" style="display:none;"> </div>

```

<body>

  <form name="form1" action="do.something.jsp" method="post">
    <div class="nppfs-elements" style="display:none;"> </div>
    <div class="nppfs-keypad-div" style="display:none;"> </div>
    .....
  </form>
</div>

<div id="nppfs-loading-modal" style="display:none;"></div>

</body>

```

5. NOS 1.0 은 기본적으로 페이지 로딩 완료 시 자동으로 npPfsStartup()이 수행되도록 합니다.
 window.onload 또는 jQuery(document).ready() 를 이용하여 처리되도록 해야 합니다.

```

jQuery(document).ready(function(){
  <%--
    npPfsStartup(document.form1, false, true, true, "enc", "on");
    1. form 양식 : 기본값 DOM Document 상의 첫번째 form
    2. 개인방화벽 사용여부 : 기본값 false
    3. 키보드보안 사용여부 : 기본값 false
    4. 단말정보수집 사용여부 : 기본값 false
    5. 마우스입력기 사용여부 : 기본값 false
    6. 키보드보안 E2E 필드 설정 속성명 : 기본값 "npkencrypt"
    7. 키보드보안 E2E 필드 설정 속성값: 기본값 "on"
  --%>
    npPfsStartup(document.form, true, true, true, true, "enc", "on");
  });

```

6. 값을 전달하기 위한 submit 하는 부분에 npPfsCtrl.waitSubmit() 함수를 추가합니다.¹⁷

```

npPfsCtrl.waitSubmit(function(){
  // do something..
});

```

¹⁷ FDS 단말정보수집에만 해당하는 부분입니다. FDS의 단말기정보 수집은 비동기식으로 처리됩니다.
 단말정보의 수집이 완료되지 전까지는 대기하다가 수집이 완료되면 정의된 기능을 수행합니다.

● 전체적인 문서 구조 예시

```

<!DOCTYPE      HTML      PUBLIC      "-//W3C//DTD      HTML      4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
추가헤더...
<script type="text/javascript" src="/pluginfree/js/jquery-1.11.0.min.js"></script>
<script type="text/javascript" src="/pluginfree/jsp/nppfs.script.jsp" charset="utf-8"></script>
<script type="text/javascript" src="/pluginfree/js/nppfs-1.x.x.js" charset="utf-8"></script>
<script type="text/javascript">
jQuery(document).ready(function(){
    if(npPfsCtrl.IsSupport()) {
        npPfsStartup(document.form, true, true, true, true, "npkencrypt", "on");
    } else {
        alert("보안프로그램을 지원하지 않는 환경입니다.");
    }

    // 가상운영체제 여부 확인
    npPfsCtrl.isVirtualMachine(function(result){
        if(result == true) {
            alert("가상운영체제 또는 원격으로 접속하셨습니다.");
        }
    });
});
</script>
</head>
<body>

문서내용...

<form name="form">
    <div class="nppfs-elements" style="display:none"></div>
    <div class="nppfs-keypad-div" style="display:none"></div>
    입력양식...
</form>

문서내용...

<div id="nppfs-loading-modal" style="display:none;"></div>
</body>
</html>

```

4.3.3. nProtect Online Security 공통 – 업무페이지

1. 페이지 상단에 PluginFree, PluginFreeException 클래스를 포함시킵니다.¹⁸

```
<%@page import="com.nprotect.pluginfree.PluginFree"%>
<%@page import="com.nprotect.pluginfree.PluginFreeException"%>
또는
import com.nprotect.pluginfree.PluginFree;
import com.nprotect.pluginfree.PluginFreeException;
```

2. 업무 로직의 상단에 다음 구문을 추가하여 넘어온 값을 검증합니다.

```
try {
    PluginFree.verify(request);19 // 또는
    PluginFree.verify(request, new String[]{"입력양식1", "입력양식2"}); // 또는
    PluginFree.verify(request, new String[]{"입력양식1", "입력양식2"}, new boolean[]{true, false});
    PluginFree.verifyFds(request); // 단말정보 수집
    PluginFree.verifySecureSubmit(request); // 구간암호화/보안전송 수집
} catch(PluginFreeException e) {
    // 페이지의 업무를 중단시키고 오류를 표시 코드를 추가하십시오.
}
```

● 값 검증 함수

| 함수명 | 값타입 | 설명 |
|-------------------|--------------------|--|
| PluginFree.verify | HttpServletRequest | Request 객체 |
| | - | 서버에서 발급한 키와 클라이언트에서 생성한 키 값만을 검증합니다. |
| PluginFree.verify | HttpServletRequest | Request 객체 |
| | String[] | E2E 또는 마우스입력기 필드의 양식명 |
| | - | 서버에서 발급한 키와 클라이언트에서 생성한 키 값만을 검증합니다.
- 각각의 입력양식에서 넘어온 값을 추가적으로 검증합니다. |
| PluginFree.verify | HttpServletRequest | Request 객체 |
| | String[] | E2E 또는 마우스입력기 필드의 양식명 |
| | boolean[] | 텍스트 입력양식과 암호입력양식 구분
텍스트 입력양식 : true
암호입력양식 : false |
| | - | 서버에서 발급한 키와 클라이언트에서 생성한 키 값만을 검증합니다. |

¹⁸ JSP구문의 예시입니다. java 객체일 경우에는 아래 코드를 사용하십시오.

¹⁹ 정책파일 nprotect.properties 의 use.intelligent.verify 의 속성값이 true 일 경우에는 입력양식의 이름을 지정하지 않아도 자동으로 입력양식을 식별합니다. 동적으로 생성되는 입력양식의 이름기준으로 검색하는 방식으로 Request 값 변조하여 넘어 올 경우(동적입력양식을 없애는 방법)에는 입력양식의 이름을 정상적으로 식별하지 못할 수 있습니다.

| | | |
|-------------------------------|---|------------|
| | <ul style="list-style-type: none"> - 각각의 입력양식에서 넘어온 값을 추가적으로 검증합니다. - 원본값과 복호화값의 일치 또는 정합성을 추가적으로 검증합니다. | |
| PluginFree.verifyFds | HttpServletRequest | Request 객체 |
| | <ul style="list-style-type: none"> - 단말정보수집의 수집정보가 올바른 형태로 넘어왔는지 검증합니다. - 서버에서 발급한 키와 클라이언트에서 생성한 키 값을 검증합니다. | |
| PluginFree.verifySecureSubmit | HttpServletRequest | Request 객체 |
| | <ul style="list-style-type: none"> - 구간암호화 요청이 올바른 형태로 넘어왔는지 검증합니다. - 구간암호화 복호화가 정상적으로 이루어졌는지 검증합니다. | |

[표 4-6] 값 검증 함수

3. PluginFreeFilter 의 설정에서 <filter-mapping>을 하지 않았을 경우에는 HttpServletRequest 객체를 참조하여 다음과 같이 복호화합니다.

```
// 페이지 상단에 PluginFreeRequest 객체를 포함시킵니다.
<%@page import="com.nprotect.pluginfree.modules.PluginFreeRequest"%> 또는
import com.nprotect.pluginfree.modules.PluginFreeRequest;

// HttpServletRequest를 인자값으로 하는 PluginFreeRequest객체를 생성합니다.
ServletRequest pluginfreeRequest = new PluginFreeRequest(request);
String cardNo1 = pluginfreeRequest.getParameter("cardNo1");
String cardNo2 = pluginfreeRequest.getParameter("cardNo2");
String cardNo3 = pluginfreeRequest.getParameter("cardNo3");
String cardNo4 = pluginfreeRequest.getParameter("cardNo4");
out.println("수동복호화(cardNo1) : " + cardNo1 + "<br />");
out.println("수동복호화(cardNo2) : " + cardNo2 + "<br />");
out.println("수동복호화(cardNo3) : " + cardNo3 + "<br />");
out.println("수동복호화(cardNo4) : " + cardNo4 + "<br />");
```

4. PluginFreeFilter 에서 복호화하기 위한 규격²⁰이 아니거나 <filter-mapping>을 하지 않았을 경우에는 HttpServletRequest 객체에 직접 데이터를 java.util.Map 형태로 변환하여 복호화가 가능합니다. 아래의 코드를 참조하여 프로그램을 작성합니다.

```
// npPfsCtrl.toJson() 함수를 사용하여 얻은 Json 문자열을 사용하는 경우
```

²⁰ PluginFreeFilter 는 HttpServletRequest, HttpServletResponse 기반의 form submit 형식으로 전달되는 데이터 기준으로 동작을 합니다. 사용자 정의에 따라 변형된 데이터 형식(예, JSON, 전문통신, 구간암호화 등)은 처리할 수 없습니다. PluginFreeRequest(HttpServletRequest, java.util.Map)형식의 복호화방법은 변형된 데이터 형식을 지원하기 위한 방법으로 다양한 형식으로 넘어온 데이터를 <key, value>형태의 Map 형태로 재가공 후에 전달되어야 합니다.

```
String json = "some of json string by npPfsCtrl.toJson()";
Map<String, String> map = RequestUtil.parseJson(json); // npPfsCtrl.toJson()
ServletRequest req = new PluginFreeRequest(request, map);

// HttpServletRequest를 Map으로 변환하는 경우
Map<String, String> map = RequestUtil.getRequestToMap(request);
ServletRequest req = new PluginFreeRequest(request, map);

// 모든 값 출력
Enumeration<String> names1 = (Enumeration<String>)request.getParameterNames();
while (names1.hasMoreElements()) {
    String name = (String) names1.nextElement();
    String values[] = request.getParameterValues(name);
    if(values != null && values.length > 0) {
        out.println("입력양식명 : " + name + " => ");
        for(String value : values) {
            out.println("[ " + value + " ]");
        }
        out.println("<br />");
    }
}
```

● 값 검증 오류

| 오류코드 | 발생 영역 | 오류 내용 / 오류 원인 |
|-------|------------------|--|
| E0001 | 키보드보안 | 세션에 저장된 키객체가 없습니다. |
| | 마우스입력기
단말정보수집 | 1.서버에서 세션을 사용할 수 없을 경우
2.사용자가 키발급을 하지 않았을 경우 |
| E0002 | 키보드보안 | 클라이언트 식별값이 없습니다. |
| | 마우스입력기
단말정보수집 | 1.키보드보안 스크립트가 실행되지 않았을 경우 |
| E0003 | 키보드보안 | 서버에 저장된 RSA 개인키를 찾을 수 없습니다. |
| | 마우스입력기
단말정보수집 | 1.키보드보안 스크립트가 실행되지 않았을 경우
2.서버에서 암호화 라이브러리가 정상적으로 로딩되지 않았을 경우 |
| E0004 | 키보드보안 | 클라이언트 암호화키값이 없습니다. |
| | 단말정보수집 | 1.키보드보안 스크립트가 실행되지 않았을 경우
2.서버에서 암호화키가 제대로 넘어오지 않았을 경우 |
| E0005 | 키보드보안 | 서버에 저장된 암호화키값을 찾을 수 없습니다. |
| | 단말정보수집 | 1.키보드보안 스크립트가 실행되지 않았을 경우
2.브라우저에서 서버로 암호화키 발급을 요청하지 않았을 경우 |
| E0006 | 키보드보안 | 서버에 저장된 암호화키값이 올바르지 않습니다. |
| | 단말정보수집 | 1.키보드보안 스크립트가 실행되지 않았을 경우 |

| | | |
|-------|---------------------------|--|
| | | 2.서버에 저장된 암호화키값을 정상적으로 읽을 수 없을 경우 |
| E0007 | 키보드보안
단말정보수집 | 클라이언트 암호화키값이 올바르지 않습니다. |
| | | 1.키보드보안 스크립트가 실행되지 않았을 경우
2.클라이언트 암호화키값을 서버에게 해석할 수 없을 경우 |
| E0008 | 키보드보안
마우스입력기
단말정보수집 | 서버의 RSA 개인키가 올바른 개인키가 아닙니다. |
| | | 1.키보드보안 스크립트가 실행되지 않았을 경우
2.서버에 저장된 RSA 개인키를 얻어올 수 없을 경우 |
| E0009 | 키보드보안
단말정보수집 | 서버키값과 클라이언트 키값으로 복호화 객체를 생성할 수 없습니다. |
| | | 1.클라이언트 암호화키값을 서버에서 분석할 수 없을 경우 |
| E0010 | 키보드보안 | 입력양식(양식명)의 E2E입력값이 없습니다. |
| | | 1.일반 입력양식은 값이 들어왔으나 E2E 필드의 값이 없을 경우 |
| E0011 | 키보드보안 | 입력양식(양식명)의 E2E입력값의 길이가 규격에 부합하지 않습니다. |
| | | 1.E2E 필드의 값의 길이 규격이 맞지 않을 경우
(입력 1 글자당 64 글자의 암호화 데이터 길이로 와야 함) |
| E0012 | 키보드보안 | 입력양식(양식명)의 입력값의 길이과 E2E입력값의 길이가 일치하지 않습니다. |
| | | 1.E2E 필드의 길이가 일반 입력양식의 입력길이와 일치하지 않을 경우 |
| E0013 | 마우스입력기 | 입력양식(양식명)의 키패드 정보값이 없습니다. |
| | | 1.마우스입력기의 정보가 없을 경우 |
| E0014 | 마우스입력기 | 입력양식(양식명)의 세션키를 찾을 수 없습니다. 세션사용이 가능한지 확인하십시오. |
| | | 1.마우스입력기의 암호화 키값을 얻어올 수 없을 경우 |
| E0015 | 마우스입력기 | 입력양식(양식명)의 키패드 입력값이 없습니다. |
| | | 1.마우스입력기의 입력 값이 없을 경우 |
| E0016 | 마우스입력기 | 입력양식(양식명)의 키패드정보가 올바르지 않습니다. |
| | | 1.마우스입력기의 정보가 올바르지 않을 경우 |
| E0017 | 마우스입력기 | 입력양식(양식명)의 마우스입력기 입력값의 길이가 규격에 부합하지 않습니다. |
| | | 1.마우스입력기의 길이가 규격에 맞지 않을 경우 |
| E0018 | 마우스입력기 | 입력양식(양식명)의 입력값의 길이과 마우스입력기 입력값의 길이가 일치하지 않습니다. |
| | | 1.마우스입력기 입력값 길이와 일반 입력양식 입력길이가 일치하지 않을 경우 |
| E0019 | 키보드보안
마우스입력기 | 암호화된 입력값의 규격이 올바르지 않습니다. |
| | | 1. 암호화된 데이터의 규격이 복호화가 가능한 값이 아닐 경우 |
| E0020 | 키보드보안
마우스입력기
단말정보수집 | 입력양식(양식명)의 복호화값이 입력가능한 값의 범위에 포함되지 않습니다. |
| | | 1. 복호화된 데이터가 아스키 이 외의 문자가 있는 경우 |
| E0021 | 구간암호화 | Request 객체가 NOS 모듈이 아닙니다. |
| | | 1. PluginFreeFilter 를 통해서 HttpServletRequest 가 처리되지 않은 경우 |
| E0022 | 구간암호화 | 구간암호화/보안전송으로 전송되지 않은 요청입니다. |
| | | 1. 구간암호화/보안전송으로 전달된 Request 요청이 아닌 경우 |

| | | |
|-------|-----------------|---|
| E0023 | 키보드보안
마우스입력기 | 원본값과 복호화값이 일치하지 않습니다. |
| | | 1. 원본값과 복호화된 값이 일치하지 않을 경우(테스트 필드) |
| E0024 | 키보드보안
마우스입력기 | 치환값에 부합하지 않는 값이 포함되어 있습니다. |
| | | 1. 원본 값이 정상적인 치환데이터 범위에 있지 않을 경우(패스워드 필드) |

[표 4-7] 값 검증 오류 표

4.3.4. 키보드보안(E2E) 상세설정

1. 입력양식은 자동으로 검색하여 일반 키보드보안이 적용됩니다.
2. 입력양식에 키보드보안을 적용하지 않게 할 경우에는 속성값을 off 로 설정합니다. 일부 브라우저에서는 키보드보안이 적용되면 IME 가 변경되지 않는 버그가 있습니다. 키보드보안이 해당 필드에서는 동작하지 않도록 처리합니다.
3. E2E 를 적용하려면 해당 입력양식에 "npkencrypt=on" attribute 속성을 추가합니다. 단, 기존의 키보드보안이 적용되어 있을 경우에는 npPfsCtrl.init() 함수의 옵션 중 AutoScanAttrName, AutoScanAttrValue 에 attribute 이름 및 값을 지정합니다.

```
// 텍스트 필드 예
<input type="text"      name="E2E_TEXT_1" npkencrypt="on" value=""/>
// 암호 필드 예
<input type="password" name="E2E_PASS_1" npkencrypt="on" value=""/>
// 기존 라온시큐어 키보드보안 텍스트 필드 예
<input type="text"      name="E2E_TEXT_2" enc="on" value=""/>
// 기존 라온시큐어 키보드보안 암호 필드 예
<input type="password" name="E2E_PASS_2" enc="on" value=""/>

// 키보드보안 사용안함
<input type="text"      name="NONE_TEXT_1" npkencrypt="off" value=""/>
// 키보드보안 사용안함(라온시큐어)
<input type="text"      name="NONE_TEXT_2" enc="off" value=""/>
```

4.3.5. 키보드보안(일반²¹) 상세설정

1. 입력양식은 자동으로 검색하여 일반 키보드보안이 적용됩니다. 일반적인 경우 특별한 설정이 필요없습니다.

²¹ 일반 키보드보안은 WAS 서버와 독립적으로 동작하는 키보드보안기능으로 키보드보안으로 입력되는 값만을 보호하는 기능입니다. 일반 키보드보안만을 사용하는 경우 자바스크립트는 스크립트 제공 시에 고정되며 E2E 와 일반키보드보안을 선택적으로 사용할 수 없습니다. 브라우저와 WAS 서버와의 네트워크 구간은 보호되지 않습니다. 일반 키보드보안 기능만을 사용하는 경우 WAS 서버의 설정(jar/jsp 복사 및 Filter 설정)이 필요없습니다.

2. 입력양식에 키보드보안을 적용하지 않게 할 경우에는 속성값을 off 로 설정합니다. 일부 브라우저에서는 키보드보안이 적용되면 IME 가 변경되지 않는 버그가 있습니다. 키보드보안이 해당 필드에서는 동작하지 않도록 처리합니다.

```
// 텍스트 필드 예
<input type="text"      name="KEY_TEXT_1" value=""/>
// 암호 필드 예
<input type="password" name="KEY_PASS_1" value=""/>
// 키보드보안 사용안함
<input type="text"      name="NONE_TEXT_1" npkencrypt="off" value=""/>
```

4.3.6. 마우스입력기 상세설정

1. 입력양식은 자동으로 검색하여 마우스입력기 보안이 적용됩니다.
2. 마우스입력기를 적용하려면 해당 입력양식에 "npkencrypt=on" attribute 속성을 추가합니다. 단, 기존의 키보드보안이 적용되어 있을 경우에는 npPfsCtrl.init() 함수의 옵션 중 AutoScanAttrName, AutoScanAttrValue 에 attribute 이름 및 값을 지정합니다.

```
// 텍스트 필드 예
<input type="text"      name="E2E_TEXT_1" npkencrypt="on" value=""/>
// 암호 필드 예
<input type="password" name="E2E_PASS_1" npkencrypt="on" value=""/>
// 기존 라온시큐어 키보드보안 텍스트 필드 예
<input type="text"      name="E2E_TEXT_2" enc="on" value=""/>
// 기존 라온시큐어 키보드보안 암호 필드 예
<input type="password" name="E2E_PASS_2" enc="on" value=""/>

// 기본(특별한 지정 필요 없음)
var options = { };
npPfsCtrl.init(options);

// 기존 라온시큐어 키보드보안 적용 예
var options = {
    AutoScanAttrName : "enc",
    AutoScanAttrValue : "on",
};

npPfsCtrl.init(options);
```

● 마우스입력기 구동 관련 입력양식 옵션

| 입력양식 옵션명 | 값타입 | 설명 | 기본값 |
|------------------|--------|---------------------------------|-----|
| npkencrypt | string | 입력양식에 마우스입력기 사용여부 값이 on 이어야 동작함 | |
| maxlength | number | 최대 입력길이 | 없음 |
| data-keypad-type | string | 마우스입력기 키패드 종류 | num |

| | | | |
|--------------------------------|---------|---|--|
| | | num : 숫자형 키패드
alpha : 소문자, 대문자, 특수문자, 숫자 ²²
korean : 한글 입력 키패드 ²³ | |
| data-keypad-action
(숫자형 전용) | string | 마우스입력기 동작 방식
password : 암호 입력(입력양식에 *로 표시)
replace : 대치코드 입력(입력양식에 1로 표시)
number : 숫자입력(입력양식에 해당 숫자표시)
account : 계좌번호(57890 값이면 입력양식에 * 표시)
amount ²⁴ : 입금액 입력(3자리마다 ,를 표시, Text입력양식에서만 사용가능) | replace |
| data-keypad-show | string | 마우스입력기 노출 형태
layer : 숨겨져 있다가 활성화시에 레이어로 나타나는 형태(내용 이동하지 않음)
div : 특정영역에 고정된 형태
block : 숨겨져 있다가 활성화시에 특정영역에 나타나는 형태(블록 아래의 내용은 아래로 이동) | layer |
| data-keypad-next | string | 최대길이입력 후 이동할 다음 입력양식
입력양식의 값이 __hide__ 로 설정할 경우 최대입력길이 입력되었을 때 마우스입력기를 자동으로 보이지 않게 합니다. | 없음 |
| data-keypad-theme | string | 마우스입력기 테마 | keypad.default.theme ²⁵ |
| data-keypad-enter | string | alpha, Korean 키패드일 경우 [Enter] 버튼의 동작 기능 | submit() |
| data-keypad-input-range | string | 문자형 키패드의 경우 입력범위를 설정합니다.
lower : 소문자 입력 가능하게 설정
upper : 대문자 입력 가능하게 설정
special : 특수문자 입력 가능하게 설정
입력예시 : lower,upper,special 구분은 , 문자 | 모든 버튼 사용 가능 |
| data-keypad-preview | boolean | 마지막 문자 보여지는 기능을 사용할지 설정합니다.
없음 : 정책설정파일의 설정을 따라갑니다.
on : 정책에 상관없이 마지막 문자를 보여줍니다. | nprotect.properties 파일의 keypad.preview |

²² 문자형키패드는 사용자가 입력한 값을 치환하여 텍스트입력양식에 입력합니다. 따라서 출력되는 값은 사용자가 입력하는 값과 다르게 표시됩니다. 출력 값의 치환은 숫자 => 1, 소문자 => a, 대문자 => A, 특수문자 => _ 로 각각 대치된 문자로 표시됩니다. (정책파일 nprotect.properties 의 keypad.use.plain.text 속성의 값이 true 이면 텍스트입력양식에서도 치환되지 않은 값으로 입력됩니다.)

²³ 암호입력양식에서 한글의 초성,중성,종성을 조합할 수 없어 한글키패드는 암호입력양식에서 사용할 수 없습니다.

²⁴ 자바스크립트(nppfs-1.x.x.js) 1.8.0 이후 버전에서만 지원합니다.

²⁵ nprotect.properties 의 keypad.default.theme 값. 지정하지 않으면 "default"로 동작합니다.

| | | | |
|---------------------------|--------|---|--|
| | | off : 정책에 상관없이 마지막 문자를 보이지 않습니다. | .use 값 |
| data-keypad-type-x | string | 마우스입력기 X좌표 위치 정책을 설정합니다.
default : 입력양식의 왼쪽에 위치시킵니다.
left : 보여지는 창의 왼쪽에 위치시킵니다.
right : 보여지는 창의 오른쪽에 위치시킵니다.
center : 보여지는 창의 가운데에 위치시킵니다. | nprotect.properties 파일의 keypad.policy.position.x 값 |
| data-keypad-x | number | X위치 정책에 따라 각각의 X좌표를 조정합니다.
default : 입력양식으로부터의 상대좌표
left : 창의 왼쪽으로부터 떨어뜨릴 좌표
right : 창의 오른쪽으로부터 떨어뜨릴 좌표
center : 창의 가운데에서 떨어뜨릴 상대좌표 | nprotect.properties 파일의 keypad.policy.position.delta.x 값 |
| data-keypad-type-y | string | 마우스입력기 Y좌표 위치 정책을 설정합니다.
default : 입력양식의 하단에 위치시킵니다.
top : 보여지는 창의 상단에 위치시킵니다.
bottom : 보여지는 창의 하단에 위치시킵니다.
middle : 보여지는 창의 중간에 위치시킵니다.
auto : 입력양식의 하단에 위치하려고 계산 후 보여지는 창을 넘어가면 입력양식의 상단 위치시킵니다. 상단에 위치시키려고 할 때 보여지는 창을 넘어가면 강제로 입력양식의 하단에 위치시킵니다. | nprotect.properties 파일의 keypad.policy.position.y 값 |
| data-keypad-y | number | Y위치 정책에 따라 각각의 Y좌표를 조정합니다.
default : 입력양식으로부터의 상대좌표
top : 창의 상단으로부터 떨어뜨릴 좌표
bottom : 창의 하단으로부터 떨어뜨릴 좌표
middle : 창의 가운데에서 떨어뜨릴 상대좌표
auto : 입력양식으로부터의 상대좌표. | nprotect.properties 파일의 keypad.policy.position.delta.y 값 |
| data-keypad-useyn-type | string | 키보드보안과 마우스입력기를 동시 사용 ²⁶ 하려고 하는 경우 활성화 방법
checkbox : 체크박스로 마우스입력기를 활성화합니다.
radio : 라디오버튼으로 마우스입력기를 활성화합니다.
toggle : 토글이미지로 마우스입력기를 활성화합니다. | 없음 |
| data-keypad-useyn-input | string | 키보드보안과 마우스입력기를 동시 사용하려고 하는 경우 활성화 입력양식. "data-keypad-useyn-type"에 입력한 형태의 입력양식이어야 함
data-keypad-useyn-type이 toggle일 경우에는 이미지 태그의 id정보값을 입력합니다. | 없음 |
| data-keypad-toggle-active | string | 마우스입력기 로딩 완료 후 초기화시에 토글버튼을 활성화여부를 지정합니다. | false |

²⁶ 키보드보안과 마우스입력기를 동시에 사용하는 경우 키보드보안이 우선적으로 실행되며 마우스입력기는 선택적으로 사용이 가능합니다. 키보드보안이 실행되지 않은 상태에서는 마우스입력기가 강제적으로 활성화되며 선택적으로 마우스입력기의 사용을 해제 할 수 없습니다.

| | | | |
|------------------------|--------|--|------------|
| | | true : 마우스입력기가 활성화상태로 시작합니다.
false : 마우스입력기가 비활성화상태로 시작합니다.(단, 강제사용정책이 설정되어 있을 경우 활성화상태로 시작될 수 있습니다.) | |
| data-keypad-toggle-on | string | 마우스입력기 활성화상태의 이미지경로를 지정합니다. | 자바스크립트 기본값 |
| data-keypad-toggle-off | string | 마우스입력기 비활성화상태의 이미지경로를 지정합니다. | 자바스크립트 기본값 |

[표 4-8] 마우스입력기 구동관련 입력양식 옵션

4.3.7. 키보드보안, 마우스입력기 치환기능 사용²⁷

1. 입력양식은 자동으로 검색하여 키보드보안/마우스입력기 보안이 적용됩니다.
2. 해당 입력양식에 "npkencrypt=re" 속성을 추가합니다. (속성의 이름은 options.AutoScanAttrName 으로 값은 대소문자 구분없이 "re"로 지정해야 합니다.)

```
// 텍스트 필드 예
<input type="text" name="E2E_TEXT_1" npkencrypt="re" value=""/>
// 암호 필드 예
<input type="password" name="E2E_PASS_1" npkencrypt="re" value=""/>
// 기존 라온시큐어 키보드보안 텍스트 필드 예
<input type="text" name="E2E_TEXT_2" enc="re" value=""/>
// 기존 라온시큐어 키보드보안 암호 필드 예
<input type="password" name="E2E_PASS_2" enc="re" value=""/>
```

3. 자바스크립트의 다음 함수를 호출하여 치환테이블과 치환데이터를 얻어옵니다.

```
// 치환테이블 얻기
var replace = npPfsCtrl.GetResultField(폼이름, 필드이름);

// 치환데이터 얻기
var table = npPfsCtrl.GetReplaceField(폼이름, 필드이름);
```

4. 외부 프로그램에서 복호화 SDK 와 연동하지 않고 마우스입력기만을 사용하는 경우에는 자바스크립트의 다음 함수를 호출하여 재 암호화된 문자열을 얻어옵니다.

²⁷ 본 기능은 외부 모듈(예, 공인인증서)과의 연동을 위한 인터페이스로 웹사이트에서 입력받은 데이터를 타 모듈에 전달하기 위한 방법입니다. 타 모듈(또는 업체)에는 인터페이스의 적용방법 및 연동을 위한 SDK 가 제공됩니다.

```
// 재암호화된 마우스입력기 입력값 얻기
var value = npPfsCtrl.GetEncryptResult(폼이름, 필드이름);
```

4.3.8. 단말정보수집 적용방법

1. PluginFree 객체를 생성한 후에 get()함수를 호출하여 단말정보 수집 결과 객체인 com.nprotect.pluginfree.PluginFreeDTO 객체를 얻습니다.
2. 수집항목은 [별첨 1.1]을 참고하십시오.

```
// 단말정보수집 데이터 검증
try{
    PluginFree.verifyFds(request);
} catch(PluginFreeException e) {
    out.println("FDS값 복호화를 위한 올바른 암호화 키값이 아닙니다.");
    return;
}

// 단말정보수집 데이터 얻기
PluginFree pf = new PluginFree(request); // 단말정보수집 컨트롤러 인스턴스 객체 생성
PluginFreeDTO dto = pf.get(); // 단말정보수집정보 객체 얻기

// 단말정보 출력
if(dto != null){
    out.println(dto.toString().replace("\n", "<br />") + "<br />");
} else {
    out.println("값을 파싱할 수 없습니다.<br />");
}
```

4.3.9. 무설치 단말정보수집 적용방법

1. PluginFree 객체를 생성한 후에 getDevice()함수를 호출하여 단말정보 수집 결과 객체인 com.nprotect.pluginfree.PluginFreeDeviceDTO 객체를 얻습니다.
2. 수집항목은 [별첨 1.2]를 참고하십시오.

```
try{
    // 단말정보수집 데이터 얻기
    PluginFreeDeviceDTO dto = new PluginFree(request).getDevice();

    // 단말정보 출력
```

```

        if(dto != null){
            out.println(dto.toString().replace("\n", "<br />\n") + "<br />");
        }
    } catch(PluginFreeException e) {
        out.println("무설치단말정보 수집 중 오류가 발생하였습니다.");
        return;
    }
}

```

4.3.10. 구간암호화/보안전송²⁸ 적용방법

1. form submit 를 통해 서버에 데이터를 전송할 경우 submit 처리로직의 마지막에 npPfsCtrl.submit(폼이름) 함수를 호출하도록 코드를 삽입합니다. ²⁹

```

// Form Submit시에 데이터 암호화
...
npPfsCtrl.submit(formname);
...

```

2. Ajax 를 통해 서버에 데이터를 전송할 경우 npPfsCtrl.serialize(데이터) 함수를 호출한 다음 반환된 데이터를 Ajax 요청으로 전송합니다. 이때 데이터는 문자열로 구성된 쿼리문이나 JSON 데이터이어야 합니다.

```

// Ajax 사용시의 데이터 암호화
$.ajax({
    data : npPfsCtrl.serialize(데이터);
});

```

3. 서버로부터 웹 브라우저로 전달되는 내용을 암호화하기 위해서는 암호화하려는 부분에 <!-- NPPFS CIPHER BEGIN --> 과 <!-- NPPFS CIPHER END -->를 추가합니다.³⁰

²⁸ PluginFreeFilter 를 사용하지 않는 환경에서는 전송된 데이터를 중간에서 요청/응답 데이터를 재처리할 수 없어 구간암호화/보안전송을 지원하지 않습니다.

²⁹ 구간암호화/보안전송에서는 일반적인 form submit 을 지원합니다. 즉, 첨부파일을 올리기 위한 multipart/form-data 는 지원하지 않습니다.

³⁰ 응답 데이터 암호화는 웹 브라우저를 통해 자바스크립트가 실행할 수 있는 환경에서만 처리가 가능합니다.

<!-- NPPFS CIPHER BEGIN --> : 암호화되는 블록의 시작을 지정합니다.³¹

<!-- NPPFS CIPHER END --> : 암호화되는 블록의 끝을 지정합니다.

```
// 응답데이터의 암호화
<!-- NPPFS CIPHER BEGIN -->
응답시 암호화되는 데이터
<!-- NPPFS CIPHER END -->
```

4.3.11. 설치 안내 페이지



[그림 4-1] 설치안내 페이지 예시

1. 기본적인 코드는 /pluginfree/jsp/nppfs.install.jsp 에 정의되어 있습니다. 해당 페이지를 참고하여 작업을 진행합니다.
2. npPfsCtrl.checkInstall() 함수는 모듈의 설치 완료여부를 확인할 수 있도록 주기적으로 설치완료여부를 확인합니다.
 - before : 완료처리 로직 수행 전에 사용자에게 고지할 사항이 있거나 작업해야 할 사항이 있을 경우 함수로 지정합니다.
 - after : 처리완료 확인 후 처리될 작업을 함수로 지정합니다.

³¹ 블록으로 지정되는 내용은 많을수록 웹브라우저의 부하를 증가시킬 수 있습니다. 보호를 원하는 블록에만 추가하는 것이 성능상 유리합니다. 블록은 한 응답에 여러 개의 블록을 지정할 수 있으나 nprotect.properties 의 cipher.block.policy 속성이 single 일 경우에는 첫 번째 나타나는 블록에만 암호화가 적용됩니다.

3. npPfsCtrl.isInstall() : 본 페이지는 설치확인이 필요할 시점에 호출되어야 합니다.(페이지 로딩완료시 수행된다면 onload 이벤트 처리시에 수행되어야 합니다.) 설치여부에 따라 수행할 작업을 명시합니다.
- success : 설치가 되어있는 상태일 경우에 처리될 함수를 지정합니다.
 - fail : 설치가 안되어 있을 경우에 처리될 함수를 지정합니다.

```

<script type="text/javascript" src="/pluginfree/js/jquery-1.11.0.min.js"></script>
<script type="text/javascript" src="/pluginfree/js/npPfs-1.0.0.js"></script>
<script type="text/javascript">
function doStartInstall() {
    npPfsCtrl.checkInstall({
        before : function() {
            alert("정상적인 설치가 되었는지 확인합니다. 설치가 완료되면 자동으로 첫 페이지로 이동합니다.");
        },
        after : function() {
            alert("설치가 완료되었습니다.");
            $("#nos-install").html("설치됨");
            location.href = "./index.jsp";
        }
    });
}
</script>

<script type="text/javascript">
jQuery(document).ready(function() {
    npPfsCtrl.isInstall({
        success : function() {
            $("#nos-install").html("설치됨");
        },
        fail : function() {
            var html = [];
            html.push("설치안됨<br />");
            if(npPfsDefine.win) {
                html.push("<a href=\"<다운로드경로>\" onclick=\"doStartInstall();\">다운로드</a>");
            } else if(npPfsDefine.mac) {
                html.push("<a href=\"<다운로드경로>\" onclick=\"doStartInstall();\">다운로드</a>");
            } else if(npPfsDefine.lnx) {
                if(npPfsDefine.lnx64) {
                    html.push("<a href=\"<다운로드경로>\" onclick=\"doStartInstall();\">페도라</a>");
                    html.push("<a href=\"<다운로드경로>\" onclick=\"doStartInstall();\">우분투</a>");
                } else {
                    html.push("<a href=\"<다운로드경로>\" onclick=\"doStartInstall();\">페도라</a>");
                    html.push("<a href=\"<다운로드경로>\" onclick=\"doStartInstall();\">우분투</a>");
                }
            }
            $("#nos-install").html(html.join("\n"));
        }
    });
});
</script>

<table border="0">

```

```

    <tr>
        <td>NOS 1.0</td>
        <td>키보드를 통한 값의 입력을.....</td>
        <td><span id="nos-install"></span></td>
    </tr>
</table>

```

4.4. 고급 페이지 적용

4.4.1. 쿠키 차단 환경³²에서 운용 방법

4.4.1.1. 쿠키 미지원 환경 설명

- 기본적으로 NOS는 서버에서 생성한 각종 값들을 세션에 저장합니다. 쿠키를 사용할 수 없는 환경에서는 세션을 위한 세션키 값을 저장할 수 없게 됩니다. 따라서, 다음페이지에서는 세션키가 없어 서버에 저장된 세션값들을 가져올 수 없게 됩니다.
- nProtect Online Security에서는 URL Rewriting 기법으로 쿠키 미지원 환경에서도 처리가 가능합니다. URL Rewriting 기법은 매번 페이지 이동 시마다 세션키 값을 계속적으로 전달하는 방법입니다.
 - i. 장점 : 쿠키를 사용할 수 없는 환경에서도 세션사용이 가능합니다.
 - ii. 단점 :
 1. 세션에 관련된 모든 페이지링크요청(페이지이동, AJAX)에 대하여 코드수정이 필요합니다.
 2. 자바스크립트로는 URL Rewriting이 불가능하여 서버(jsp, servlet)단의 코드수정이 필요합니다.
 3. 세션키가 페이지의 호출경로에 직접적으로 노출됩니다.

4.4.1.2. 쿠키 미지원 환경 지원

- ① NOS에서 사용하는 URL경로 변경 : npPfsStartup() 자바스크립 함수를 호출하기 전에 아래 코드를 삽입하여 호출 경로를 변경합니다. 아래 코드는 WAS서버 페이지(jsp페이지)에서 해야 합니다.

```
// URL Rewriting : Cookie가 허용되지 않는 브라우저에서는 다음 구문을 추가하여
// 쿠키없이도 동작이 가능하도록 처리
// 키발급 경로
uV.dV.Gf = "<%= response.encodeURL(\"/pluginfree/jsp/nppfs.key.jsp\") %>";
// 키삭제 경로
uV.dV.zf = "<%= response.encodeURL(\"/pluginfree/jsp/nppfs.remove.jsp\") %>";
// 마우스입력기 페이지
uV.dV.zo = "<%= response.encodeURL(\"/pluginfree/jsp/nppfs.keypad.jsp\") %>";
// 초기화상태 확인경로
uV.dV.eP = "<%= response.encodeURL(\"/pluginfree/jsp/nppfs.ready.jsp\") %>";
// 설치안내 페이지
```

³² 쿠키 미지원 환경

1. 모바일 브라우저 쿠키 미설정
2. PC환경의 브라우저 쿠키 미설정
3. iframe을 사용하는 페이지에서 제3자 쿠키 제공 미동의
4. 광고제공용 정보 전송 방지(Do Not Track)설정

```
uV.dV.Fz = "<%= response.encodeURL("/pluginfree/jsp/nppfs.install.jsp") %>";
```

- ② com.inca.nprotect.pluginfree.xxx.xxx.xxx.jar³³ 파일을 교체하여 마우스입력기에서 참고하는 경로들을 변경합니다.
- ③ 데이터를 넘기는 form의 action 경로를 변경합니다.

| |
|---|
| <p>변경 전</p> <pre><form action="/next.jsp" method="post"></pre> |
| <p>변경 후</p> <p>// URL Rewriting : Cookie가 허용되지 않는 브라우저에서는 다음 구문으로 변경하여 쿠키없이도 동작이 가능하도록 처리</p> <pre><form action="<%= response.encodeURL("/next.jsp") %>" method="post"></pre> |

4.4.2. 마우스입력기 PKI 연동가이드

- 웹 브라우저 컨트롤을 포함하는 다이얼로그에서 마우스입력기를 연동을 위한 인터페이스로 다이얼로그의 웹 브라우저 컨트롤이 초기에 호출되는 페이지는 제공된 PKI연동 웹 페이지(/pluginfree/jsp/nppfs.pki.jsp)를 호출하도록 설정합니다.
- 기본적인 경로에 NOS 서버모듈을 설치하지 않았을 경우에는 nppfs.pki.jsp 페이지 내에 nppfs-1.x.x.js와 nppfs.script.jsp, jquery-1.x.x.js의 경로를 수정합니다.
- 웹 브라우저 컨트롤은 웹페이지 이동 이벤트를 수신하고 있어야 합니다. 마우스입력 후 확인 버튼을 누른 경우 [transkey://localhost/?rand=키값&result=암호화값] 으로 페이지를 이동하도록 되어 있습니다.
- 연동하는 프로그램에서는 transkey:// 로 시작하는 URL을 확인하여 키값과 암호화값을 받아서 처리합니다. 제공된 복호화 함수에 전달하면 입력한 값이 반환됩니다.

4.4.3. 자바 도구

4.4.3.1. 데이터 재전송 API³⁴

- 복호화를 WAS 서버에게 처리하지 않고 다른 서버 또는 서비스에 암호화된 데이터를 전달하기 위한 API 입니다.

| 재전송 방법 | 암호화 알고리즘 |
|-------------------------------|--|
| Transfer.DEFAULT | 키교환 암호/복호화 : RSA/ECB/PKCS1Padding
데이터 암호/복호화 : ARIA/CBC/PKCS5Padding |
| Transfer.KCP_QUICK_PAY | 키교환 암호/복호화 : RSA/ECB/PKCS1Padding
데이터 암호/복호화 : SEED-128/CFB/NoPadding |

³³ 2015 년 06 월 12 일 이후의 서버 모듈부터 지원됩니다.

³⁴ 사전 협의를 통해 개발되는 API 입니다. 원하는 데이터전송 API 가 없을 경우에는 지원가능여부를 먼저 문의하십시오.

| 재전송 방법 | 암호화 알고리즘 |
|------------------------------|--|
| Transfer.HANA_SAVINGS | 키교환 암호/복호화 : RSA/ECB/PKCS1Padding
데이터 암호/복호화 : ARIA/CBC/PKCS5Padding |
| Transfer.KFTC | 키교환 암호/복호화 : RSA/ECB/PKCS1Padding
데이터 암호/복호화 : SEED128/CBC/PKCS5Padding |

[표 4-9] 데이터 재전송 방법

● 데이터재전송 API

| Transfer API | 인자값 | 반환값 |
|---|---|------------------|
| Transfer.getInstance
(int, Object) | int 데이터전송방법
Object : 인스턴스 생성용 추가정보 | Transfer 객체 인스턴스 |
| Transfer.getInstance
(int, Object, request) ³⁵ | int 데이터전송방법
Object : 인스턴스 생성용 추가정보
HttpServletRequest 객체 인스턴스 | Transfer 객체 인스턴스 |
| getKey | 없음 | 암호화 |
| setKey | 암호화된 [암/복호화키] 데이터 | 없음 |
| get : 송신모드 | 암호화할 데이터 또는 입력양식명 | 암호화된 데이터 |
| set : 수신모드 | 복호화할 데이터 | 복호화된 데이터 |

[표 4-10] 데이터 재전송 API

4.4.3.2. 재전송 API

● 재전송 방법(송신부)

1. [데이터 암호/복호화키]의 암호화를 위한 [공개키]를 준비합니다.

2. 데이터전송객체를 아래와 같이 생성합니다.

데이터를 직접 가공할 경우

Transfer sender = PluginFreeTransfer.getInstance(**재전송 방법**, publicKey);

HttpServletRequest 객체를 참조하여 가공할 경우

Transfer sender = PluginFreeTransfer.getInstance(**재전송 방법**, publicKey, request);

³⁵ nProtect Online Security의 암호화 필터기능에서 수행하는 작업을 수동으로 작업한 다음 데이터 재전송을 위한 객체를 생성하는 방법입니다. 각 기능의 설정방법은 앞 장의 키보드보안 / 마우스입력기 / 단말정보수집 적용방법을 참고하십시오.

3. 데이터전송객체에서 암호화된 [데이터암/복호화키]를 얻습니다.
String cipherKey = sender.getKey();
4. 데이터전송객체에서 암호화된 각각의 데이터를 얻습니다.

```
// 데이터 송신 처리(데이터직접처리)
Transfer sender = PluginFreeTransfer.getInstance(재전송 방법, publicKey);
String cipherKey = sender.getKey();           // 암/복호화키 받기
String cipherCard1 = sender.get("전송할 암호화 데이터");

// 데이터 송신 처리(HttpServletRequest 간접처리)
Transfer sender = PluginFreeTransfer.getInstance(재전송 방법, 공개키, request);
String cipherKey = sender.getKey();           // 암/복호화키 받기
String cipherCard1 = sender.get("입력양식명");
```

● 재전송 방법(수신부)

1. [데이터 암/복호화키]의 복호화를 위한 [개인키]를 준비합니다.
2. 데이터전송객체(Transfer 객체)를 아래와 같이 생성합니다.
Transfer receiver = PluginFreeTransfer.getInstance(재전송 방법, 개인키);
3. 데이터전송객체에 암호화된 암호화키값을 설정합니다.
receiver.setKey(cipherKey);

```
데이터전송객체에 각각의 암호화된 값을 얻어옵니다.
String plainCard1 = receiver.get(cipherCard1);
// 데이터 수신 처리
Transfer receiver = PluginFreeTransfer.getInstance(재전송 방법, privateKey);
receiver.setKey(cipherKey);           // 암/복호화키 설정
String plainCard1 = receiver.get(cipherCard1);
.....
```

4.4.3.3. 키보드보안/마우스입력기 복호화 방식 찾기

- 특정 입력양식이 키보드보안과 마우스입력기를 같이 사용하는 경우 복호화 페이지에서 복호화 모듈의 동작상태를 확인하려면 PluginFree.findDecryptMode()함수를 사용하여 상태를 확인합니다. 인자값으로 HttpServletRequest 와 입력양식이름을 지정합니다.

| 반환값 | 실제값 | 설명 |
|--|-----|-------|
| PluginFreeConst.DECRYPTION_MODE_ERROR | -1 | 예외발생 |
| PluginFreeConst.DECRYPTION_MODE_KEYBOARD | 10 | 키보드보안 |

| 반환값 | 실제값 | 설명 |
|--|-----|---------------------|
| PluginFreeConst.DECRYPTION_MODE_KEYBOARD_ERROR | 11 | 키보드보안(verify 검증오류) |
| PluginFreeConst.DECRYPTION_MODE_KEYPAD | 20 | 마우스입력기 |
| PluginFreeConst.DECRYPTION_MODE_KEYPAD_ERROR | 21 | 마우스입력기(verify 검증오류) |

[표 4-11] 복호화 모듈 동작상태 확인

```

/* PluginFreeConst.DECRYPTION_MODE_ERROR : 예외발생
 * PluginFreeConst.DECRYPTION_MODE_KEYBOARD : 키보드보안
 * PluginFreeConst.DECRYPTION_MODE_KEYBOARD_ERROR : 키보드보안(verify 검증오류)
 * PluginFreeConst.DECRYPTION_MODE_KEYPAD : 마우스입력기
 * PluginFreeConst.DECRYPTION_MODE_KEYPAD_ERROR : 마우스입력기(verify 검증오류) */
int mode = PluginFree.findDecryptionMode(HttpServletRequest request, String name)

```

4.4.4. 자바스크립트 도구

4.4.4.1. 가상운영체제 접속여부 확인

- npPfsCtrl.isVirtualMachine() 함수를 사용하여 현재 접속환경이 가상운영체제인지를 판단합니다. Callback 함수를 사용하여 전달된 인자 값이 true 일 경우 가상운영체제로 판단된 결과입니다.

```

// 가상운영체제 여부 확인
npPfsCtrl.isVirtualMachine(function(result){
    if(result == true) {
        alert("가상운영체제 또는 원격으로 접속하셨습니다. 보안프로그램의 일부 기능을 사용할 수 없습니다. ");
    }
});

```

4.4.4.2. 동적으로 생성되는 필드 및 값 복사

- npPfsCtrl.copy() 함수를 사용하여 동적 생성 필드 및 값을 복사합니다. 동적으로 생성된 필드만을 대상으로 합니다. 즉, 기존에 존재하는 입력양식의 필드들은 복사하지 않습니다. 개발자가 수동으로 등록해야 합니다.
- 대상 입력양식이 이미 존재할 경우 덮어씁니다. 누락되는 데이터가 발생하지 않게 주의하여 사용해야 합니다.

```

// 입력양식의 동적필드 값 복사
npPfsCtrl.copy(As-Is Form 이름, To-Be Form 이름);

```

4.4.4.3. 동적으로 생성되는 필드에 키보드보안/마우스입력기 설정

- npPfsCtrl.RegistDynamicField() 함수를 이용하여 입력양식이 동적으로 추가되는 경우 생성된 입력양식에 키보드보안/마우스입력기를 사용하도록 처리합니다. 입력양식이 생성되고 난 이후에 호출하도록 해야 합니다.

```
// 동적 입력양식의 마우스입력기/키보드보안기능 사용
npPfsCtrl.RegistDynamicField(Form 양식, 동적필드이름);
```

4.4.4.4. Ajax 를 통해 동적으로 생성되는 필드에 키보드보안/마우스입력기 설정

- npPfsCtrl.RescanField() 함수를 이용하여 Ajax 를 통해 입력양식이 동적으로 추가되는 경우 생성된 입력양식에 키보드보안/마우스입력기를 사용하도록 처리합니다. 입력양식이 생성되고 난 이후에 호출하도록 해야 합니다.
- npPfsCtrl.RegistDynamicField() 함수는 각각의 입력양식을 수동으로 등록하는 함수입니다. 생성되는 입력양식이 여러 개라면 npPfsCtrl.RescanField() 함수를 이용을 권장합니다.

```
// 동적 입력양식의 마우스입력기/키보드보안기능 사용
npPfsCtrl.RescanField();
```

4.4.4.5. 키보드보안/마우스입력기 입력값 필터링(웹 페이지 전체)³⁶

- npPfsCtrl.SetGlobalKeyValidation() 함수를 사용하여 필드에 입력되는 키 값을 예외처리합니다. 본 함수는 NOS 가 실행되고 있는 페이지 전체에 일괄적으로 적용됩니다.
- NOS 의 키보드보안이 초기화되기 전(npPfsStartup 함수 호출 전)에 본 함수를 적용해야 정상 동작합니다.
- 키 값을 검증 로직이 구현된 함수를 인자 값으로 전달합니다. 이때 구현되는 함수는 keycode 값을 인자 값으로 받습니다. 로직 구현이 키 값을 사용하려면 true 를 반환하고 취소하려면 false 를 반환되도록 합니다.

```
// 숫자만 입력되도록 구현된 예제
npPfsCtrl.SetGlobalKeyValidation(function(keyCode) {
    // true : 정합성 충족, false : 정합성 미충족
    if(keyCode >= 48 && keyCode <=57) return true;
    return false;
});

function npPfsStartup(.....);
```

³⁶ NOS 의 키보드보안 모듈은 keydown 이벤트를 기반으로 동작합니다. keydown 이벤트로 기존 사이트에서 필터링을 적용하는 경우 정상적으로 이미 정의된 필터링 기능이 동작하지 않을 수 있습니다. keydown 시 특정 키만을 허용하도록 할 경우 본 기능을 사용합니다.

4.4.4.6. 키보드보안/마우스입력기 입력값 필터링(단일 입력양식)³⁷

- 입력양식의 속성에 입력값구분을 위한 속성을 추가한 다음에 npPfsStartup 함수 호출이전에 npPfsExtension.keyValidation() 함수를 구현합니다. 인자값으로 입력양식의 인스턴스와 키코드가 전달됩니다. 입력가능한 키보드면 true 를 반환하고 입력이 불가능한 항목에 대해서는 false 를 반환하도록 함수를 구현합니다.

```
// true : 입력가능문자, false : 정합성불가/입력불가문자
var npPfsExtension = new function() {
    this.keyValidation = function(element, keyCode) {
        var dataType = element.getAttribute("data-type");
        // 0 = 48, 9 = 57, a = 97, z = 122, A = 65, Z = 90
        var key = parseInt("" + keyCode);
        if(dataType == "n") {
            if(key < 48 || key > 57) {
                return false;
            }
        }

        return true;
    },
};
function npPfsStartup(.....);
```

4.4.4.7. NOS 보안 프로그램 지원환경 확인

- npPfsCtrl.IsSupport() 함수를 사용하여 현재의 접속환경이 NOS 지원환경인지를 확인합니다.
- 반환값 : true : 지원환경, false 미지원 환경

```
// 숫자만 입력되도록 구현된 예제
jQuery(document).ready(function(){
    var isSupport = npPfsCtrl.IsSupport();
    if(!isSupport) {
        // 이 부분에 미지원 환경에 대한 가이드 또는 추가 업무 로직을 구현합니다.
        alert("보안프로그램을 지원하지 않는 환경입니다.");
    }
});
```

³⁷ NOS의 키보드보안 모듈은 keydown 이벤트를 기반으로 동작합니다. keydown 이벤트로 기존 사이트에서 필터링을 적용하는 경우 정상적으로 이미 정의된 필터링 기능이 동작하지 않을 수 있습니다. keydown 시 특정 키만을 허용하도록 할 경우 본 기능을 사용합니다.

4.4.4.8. 치환테이블 및 치환데이터 얻기

- 치환기능이 활성화된 상태에서만 정상적인 값을 반환합니다.
- npPfsCtrl.GetReplaceField() 함수를 사용하여 치환테이블 정보를 얻어옵니다.
- npPfsCtrl.GetResultField() 함수를 사용하여 치환데이터 정보를 얻어옵니다.

```
var table = npPfsCtrl.GetReplaceField(폼이름, 필드이름); // 치환테이블 얻기
var replace = npPfsCtrl.GetResultField(폼이름, 필드이름); // 치환데이터 얻기
```

4.4.4.9. 키보드보안 키보호 해제

- 키보드보안의 키보호를 해제처리합니다. 브라우저에서 입력양식의 focusout 이벤트가 정상적으로 발생하지 않을 경우(예, 입력양식에서 Enter 가 입력되었을 때)에 호출하도록 로직을 변경합니다.
- Enter 입력을 통하여 form 의 값을 넘기기 전에 이 함수를 호출해야 합니다.
- 인자값으로 입력양식 및 처리 후 Callback 함수 지정이 가능합니다. 입력양식이 없을 경우에는 마지막으로 키보드보안이 적용된 필드에 자동 키보호 해제가 실행됩니다.

```
npPfsCtrl.doFocusOut(입력양식, callback);
```

4.4.4.10. 재 암호화된 마우스입력기 입력값 얻기³⁸

- 외부 프로그램에서 복호화 SDK 와 연동하지 않고 마우스입력기만을 사용하는 경우에는 자바스크립트의 다음 함수를 호출하여 재 암호화된 문자열을 얻어옵니다.

```
// 재 암호화된 마우스입력기 입력값 얻기
var value = npPfsCtrl.GetEncryptResult(폼이름, 필드이름);
```

4.4.4.11. 데이터 수동 복호화(Filter 기능 사용하지 않기)

- PluginFreeFilter 의 설정에서 <filter-mapping>을 하지 않았을 경우에는 HttpServletRequest 객체를 참조하여 다음과 같이 복호화합니다.

```
// 페이지 상단에 PluginFreeRequest 객체를 포함시킵니다.
<%@page import="com.nprotect.pluginfree.modules.PluginFreeRequest"%> 또는
```

³⁸ 이 때 사용하는 암호화키는 nprotect.properties 의 interoperate.with.certificate.key 값을 사용합니다. ARIA256/CBC/PKCS5Padding 암호화 알고리즘을 사용하며 초기화벡터(IV, 16bytes)는 동적으로 생성되며 암호화데이터의 앞부분에 hex 인코딩된 값을 덧붙입니다.

```
import com.nprotect.pluginfree.modules.PluginFreeRequest;

// HttpServletRequest를 인자값으로 하는 PluginFreeRequest객체를 생성합니다.
ServletRequest req = new PluginFreeRequest(request);
String cardNo1 = req.getParameter("cardNo1");
String cardNo2 = req.getParameter("cardNo2");
String cardNo3 = req.getParameter("cardNo3");
String cardNo4 = req.getParameter("cardNo4");

out.println("수동복호화(cardNo1) : " + cardNo1 + "<br />");
out.println("수동복호화(cardNo2) : " + cardNo2 + "<br />");
out.println("수동복호화(cardNo3) : " + cardNo3 + "<br />");
out.println("수동복호화(cardNo4) : " + cardNo4 + "<br />");
```

4.4.4.12. 키보드보안/마우스입력기를 위한 입력양식 속성확장³⁹

- 키보드보안의 입력양식에 동작을 확장하기 위하여 특정한 속성을 추가하려면 npPfsExtension.secureKeyUiModifier 함수에서 구현하도록 합니다.
- 마우스입력의 입력양식에 동작을 확장하기 위하여 특정한 속성을 추가하려면 npPfsExtension.keypadUiModifier 함수에서 구현하도록 합니다.
- 본 함수는 npPfsStartup() 함수가 호출되기 전에 먼저 구현되어 있어야 합니다.

```
var npPfsExtension = new function() {
    this.secureKeyUiModifier = function(element) {
        var attr = jQuery(element).attr("enc");
        if(typeof(attr) == "undefined" || attr == "") {
            jQuery(element).attr({"enc" : "off"});
        }
    },
    this.keypadUiModifier = function(element) {
        jQuery(element).attr({"data-keypad-type" : "alpha"});
    }
};

function npPfsStartup(.....);
```

³⁹ 신규로 구축되는 홈페이지에서는 본 코드의 확장이 필요없으나 기존의 홈페이지에서 이미 구현된 다양한 속성을 NOS 키보드보안/마우스입력기에서도 사용자가 의도하는 동작을 하기 위한 방법입니다. 예를 들어 data-type 이라는 속성에 따라 마우스입력기의 종류를 선택하여야 할 경우에 본 기능을 사용합니다.

4.4.4.13. 단말정보수집 사용자정의 데이터 추가

- 단말정보수집 시 사용자 정의 데이터가 있을 경우에는 사용자정의 데이터(문자열)를 반환하도록 npPfsExtension.additionalData() 함수를 구현하여 데이터를 추가적으로 전달합니다.
- 전달하고자 하는 데이터가 1 개 이상일 경우에는 파싱가능한 문자열로 변환하여 처리합니다. 즉, JSON 형태로 값을 처리한다음 JSON.stringify() 함수를 사용하여 문자열로 변환하여 반환하도록 합니다.
- 본 함수는 npPfsStartup() 함수가 호출되기 전에 먼저 구현되어 있어야 합니다.

```
var npPfsExtension = new function() {
    this.additionalData = function(event) {
        return "사용자 정의 데이터";
    }
    /*
        var data = {
            korstring : "한글로 인코딩된 추가 데이터"
            , engstring : "this is additional data...."
            , natip : "<%= request.getRemoteAddr()%">"
        }
        return JSON.stringify(data);
    */
};
function npPfsStartup(.....);
```

4.4.4.14. 웹페이지를 벗어날 경우의 확장 처리

- 웹페이지를 벗어날 경우 사용자에게서 특정한 알림 메시지를 출력하게 하려면 아래와 같이 npPfsExtension.beforeFinalize() 함수를 구현하여 처리하도록 합니다.
- 본 함수는 npPfsStartup() 함수가 호출되기 전에 먼저 구현되어 있어야 합니다.

```
var npPfsExtension = new function() {
    this.beforeFinalize = function(event) {
        if(false) {
            event = (event || window.event);
            var m = '작업이 아직 진행중에 있습니다. 저장하지 않은 채로 다른
            페이지로 이동하시겠습니까?'; // a space
            (event || window.event).returnValue = m;
            return m;
        }
        return null;
    }
};
```

```
function npPfsStartup(.....);
```

4.4.4.15. 특정 Form 의 입력값들을 Json 형태의 값으로 변환하기

- 데이터통신 구간에서 Json 형태로 데이터를 교환할 경우 입력한 값과 암호화된 값들을 Json 형태로 반환해주는 함수입니다.
- 함수의 인자값으로 Json 으로 변환하려는 Form 의 이름을 전달합니다.

```
var json = npPfsCtrl.toJson(Form이름);
```

4.4.4.16. 특정 Form 의 입력값들을 <key, value>형태의 값으로 변환하기

- 입력한 값을 가공하기 위하여 <key, value>형태의 값을 반환합니다.
- 함수의 인자값으로 변환하려는 Form 의 이름과 동적으로 생성된 필드들만 변환할지를 전달합니다.
- 동적필드여부 값이 true 이면 동적으로 생성된 필드들만 <key, value>형태로 값을 반환하며 false 이면 모든 필드의 값들을 <key, value>형태로 반환합니다.

```
var keyValues = npPfsCtrl.serializeArray(Form 이름, 동적필드여부);
```

4.4.4.17. 데이터 암호화 및 복호화

- 특정한 값을 암호화하기를 원할 경우에는 npPfsCtrl.encryptValue()함수를 사용합니다. 인자값으로 암호화하려는 값과 값이 암호화된 후 처리될 콜백함수를 전달합니다.
- 본 함수는 비동기로 처리되어 암호화이후의 로직이 있다면 콜백 함수 내에 구현해야 합니다. 즉, form.submit() 이 하단에 구현되어 있었을 경우 기존의 코드를 삭제하고 콜백함수에 구현되어야 합니다.
- 데이터의 복호화는 PluginFree.decrypt("암호화된값") 함수를 사용합니다.

```
var keyValues = npPfsCtrl.encryptValue("암호화하려는값", function(result){
    // alert(result);
});
```

4.4.4.18. 키보드보안/마우스입력기 색상 변경

- npPfsCtrl.setColor()함수를 사용합니다. 인자값으로 JSON 형태의 값을 지정합니다.

```

npPfsCtrl.setColor({
    TextColor : "",           // 키보드보안 글자 색상
    FieldBgColor : "",        // 키보드보안 배경 색상
    ReTextColor : "",         // 키보드보안 치환 글자 색상
    ReFieldBgColor : "",      // 키보드보안 치환 배경 색상
    OnTextColor: "",          // 마우스입력기 포커스 글자 색상
    OnFieldBgColor : "",      // 마우스입력기 포커스 배경 색상
    OffTextColor : "",         // 마우스입력기 글자 색상
    OffFieldBgColor : ""      // 마우스입력기 배경 색상
});

```

5. 고급 적용 가이드

5.1. NOS 이벤트 활용

5.1.1. NOS 발생 이벤트

- 복호화를 WAS 서버에게 처리하지 않고 다른 서버 또는 서비스에 암호화된 데이터를 전달하기 위한 API 입니다.

| 이벤트명 | 이벤트 발생 시점 | 전달값 |
|--------------------------------------|---------------------------------------|---|
| nppfs-before-init | NOS의 초기화 기능이 수행되기 전 | message : 메시지
time : 발생시각 |
| nppfs-after-init | NOS의 초기화 기능이 완료된 후 | message : 메시지
time : 발생시각 |
| nppfs-before-startup | NOS의 실행 기능이 수행되기 전 | message : 메시지
time : 발생시각 |
| nppfs-after-startup | NOS의 실행 기능이 완료된 후 | message : 메시지
time : 발생시각 |
| nppfs-keep-alive | NOS 연결 유지 명령이 실행될 때 | message : 메시지
time : 발생시각 |
| nppfs-npn-before-startup | 개인방화벽 구동이 수행되기 전 | message : 메시지
time : 발생시각 |
| nppfs-npn-after-startup | 개인방화벽 구동이 완료된 후 | message : 메시지
time : 발생시각 |
| nppfs-npn-keep-alive | 개인방화벽 연결 유지 명령이 실행될 때 | message : 메시지
time : 발생시각 |
| nppfs-npn-finalized | 개인방화벽 구동의 중지가 완료된 후
(페이지를 벗어났을 경우) | message : 메시지
time : 발생시각 |
| nppfs-npk-before-startup | 키보드보안 구동이 수행되기 전 | message : 메시지
time : 발생시각 |
| nppfs-npk-after-startup | 키보드보안 구동이 완료된 후 | message : 메시지
time : 발생시각 |
| nppfs-npk-before-regist-field | 키보드보안 입력양식에 이벤트 바인딩 및 초기화를 수행하기 전 | message : 메시지
time : 발생시각
target : 입력양식 인스턴스
form : 해당 form 이름
name : 입력양식 이름 |
| nppfs-npk-after-regist-field | 키보드보안 입력양식에 이벤트 바인딩 및 초기화 완료 후 | message : 메시지
time : 발생시각
target : 입력양식 인스턴스
form : 해당 form 이름
name : 입력양식 이름 |

| 이벤트명 | 이벤트 발생 시점 | 전달값 |
|--------------------------------------|--|---|
| nppfs-npk-focusin | 입력양식에 포커스가 들어왔을 경우 | message : 메시지
time : 발생시각
target : 입력양식 인스턴스
form : 해당 form 이름
name : 입력양식 이름 |
| nppfs-npk-focusout | 입력양식의 포커스를 잃었을 경우 | message : 메시지
time : 발생시각
target : 입력양식 인스턴스
form : 해당 form 이름
name : 입력양식 이름 |
| nppfs-npk-put-complete | 키보드보안 값 입력 후 | message : 메시지
time : 발생시각
target : 입력양식 인스턴스
form : 해당 form 이름
name : 입력양식 이름 |
| nppfs-npk-finalized | 키보드보안 구동의 중지가 완료된 후
(페이지를 벗어났을 경우) | message : 메시지
time : 발생시각 |
| nppfs-npf-before-startup | 단말정보수집의 구동이 수행되기 전 | message : 메시지
time : 발생시각 |
| nppfs-npf-after-startup | 단말정보수집의 구동이 완료된 후 | message : 메시지
time : 발생시각 |
| nppfs-npf-complete | 단말정보수집이 완료된 후 | message : 메시지
time : 발생시각 |
| nppfs-npf-finalized | 단말정보수집의 구동이 중지 완료된 후
(페이지를 벗어났을 경우) | message : 메시지
time : 발생시각 |
| nppfs-npv-before-startup | 마우스입력기의 구동이 수행되기 전 | message : 메시지
time : 발생시각 |
| nppfs-npv-after-startup | 마우스입력기의 구동이 완료된 후 | message : 메시지
time : 발생시각 |
| nppfs-npf-finalized | 마우스입력기의 구동이 중지 완료된 후
(페이지를 벗어났을 경우) | message : 메시지
time : 발생시각 |
| nppfs-npv-before-regist-field | 마우스입력기 입력양식에 이벤트 바인딩
및 초기화를 수행하기 전 | message : 메시지
time : 발생시각
target : 입력양식 인스턴스
form : 해당 form 이름
name : 입력양식 이름 |
| nppfs-npv-after-regist-field | 마우스입력기 입력양식에 이벤트 바인딩
및 초기화 완료 후 | message : 메시지
time : 발생시각
target : 입력양식 인스턴스
form : 해당 form 이름
name : 입력양식 이름 |

| 이벤트명 | 이벤트 발생 시점 | 전달값 |
|---|--|---|
| nppfs-npv-enabled
nppfs.npv.enabled(호환용) | 마우스입력기가 활성화되었을 때
(사용여부 체크가 on상태로 전환되었을 때) | message : 메시지
time : 발생시각
target : 입력양식 인스턴스
form : 해당 form 이름
name : 입력양식 이름 |
| nppfs-npv-disabled
nppfs.npv.disabled(호환용) | 마우스입력기가 비활성화되었을 때
(사용여부 체크가 off상태로 전환되었을 때) | message : 메시지
time : 발생시각
target : 입력양식 인스턴스
form : 해당 form 이름
name : 입력양식 이름 |
| nppfs-npv-closed
nppfs.npv.closed(호환용) | 마우스입력기가 닫혔을 때(오른쪽 상단의 닫기버튼을 클릭했을 경우) | message : 메시지
time : 발생시각
target : 입력양식 인스턴스
form : 해당 form 이름
name : 입력양식 이름 |
| nppfs-npv-before-show | 마우스입력기를 보이게 하기 전 | message : 메시지
time : 발생시각
target : 입력양식 인스턴스
form : 해당 form 이름
name : 입력양식 이름 |
| nppfs-npv-after-show | 마우스입력기를 보이게 한 후 | message : 메시지
time : 발생시각
target : 입력양식 인스턴스
form : 해당 form 이름
name : 입력양식 이름 |
| nppfs-npv-after-hide | 마우스입력기를 숨겼을 때 | message : 메시지
time : 발생시각
target : 입력양식 인스턴스
form : 해당 form 이름
name : 입력양식 이름 |
| nppfs-npl-before-startup | Flash SDK의 구동이 수행되기 전 | message : 메시지
time : 발생시각 |
| nppfs-npl-after-startup | Flash SDK의 구동이 완료된 후 | message : 메시지
time : 발생시각 |
| nppfs-npl-finalized | 마우스입력기의 구동이 중지 완료된 후
(페이지를 벗어났을 경우) | message : 메시지
time : 발생시각 |
| nppfs-npl-before-regist-field | 마우스입력기 입력양식에 이벤트 바인딩 및 초기화를 수행하기 전 | message : 메시지
time : 발생시각
name : Flash 입력양식 ID |
| nppfs-npl-after-regist-field | 마우스입력기 입력양식에 이벤트 바인딩 및 초기화 완료 후 | message : 메시지
time : 발생시각
name : Flash 입력양식 ID |

[표 5-1] NOS 이벤트

5.1.2. NOS 발생 이벤트 처리

- jQuery 의 on 함수를 사용하여 아래의 코드와 같이 이벤트를 받아서 처리합니다.
event 객체에 전달되는 값은 NOS 이벤트 표를 참고하십시오.

```
jQuery(document).on("발생이벤트명", function(event) {
    console.log(event.message);
    var target = event.target;
    if(target.value == "") {
        $("# + event.name).css({"z-index": ""});
    }
});
```

5.2. ASP.NET(C#) 언어 지원

5.2.1. ASP.NET 프로젝트 설정⁴⁰

1. 동적 라이브러리 파일을 라이브러리 경로에 복사합니다. 일반적으로 라이브러리 경로는 프로젝트 루트의 /bin 경로입니다. 다른 경로를 사용하실 경우에는 해당 경로에 파일을 복사하거나 프로젝트 이외의 경로에 설정할 경우에는 [시스템 > 고급 시스템 설정 > 고급 > 환경 변수]에서 Path 항목에 해당 dll의 경로를 추가하십시오.

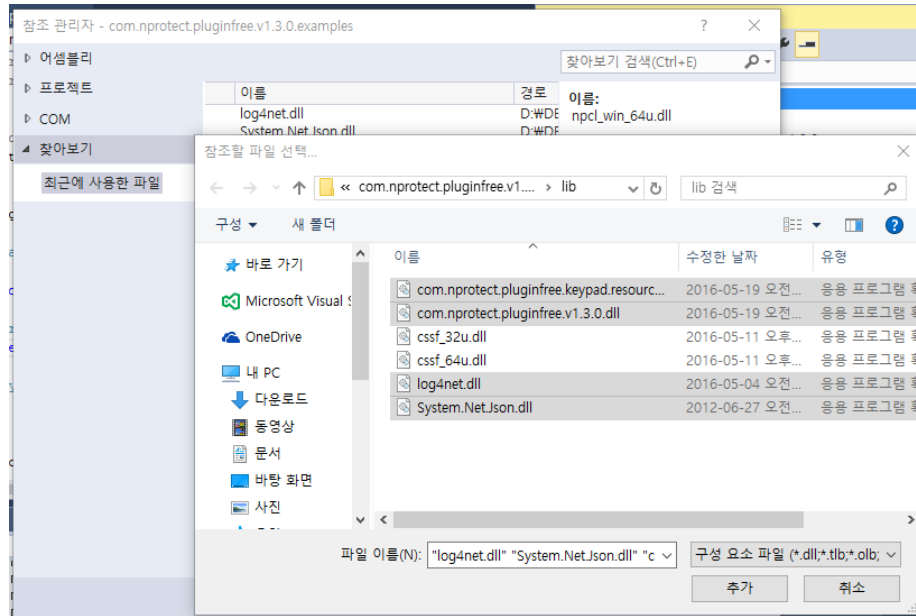
| 파일명 | 설명 |
|--|--|
| cssf_32u.dll, ssf_64u.dll | - 암호화 라이브러리
- 서버의 32bit/64bit 실행여부에 따라 자동으로 링크되므로 2개 모두 복사 필요 |
| com.nprotect.pluginfree.v1.3.0.dll | - NOS 비즈니스 로직 구현 |
| com.nprotect.pluginfree.keypad.resources.dll | - 마우스입력기 관련 그림 및 설정 ⁴¹ |
| log4.net.dll | - 로그 출력 라이브러리 |
| System.net.Json.dll | - JSON 파싱 라이브러리 |

[표 5-2] ASP.NET 라이브러리 파일 목록

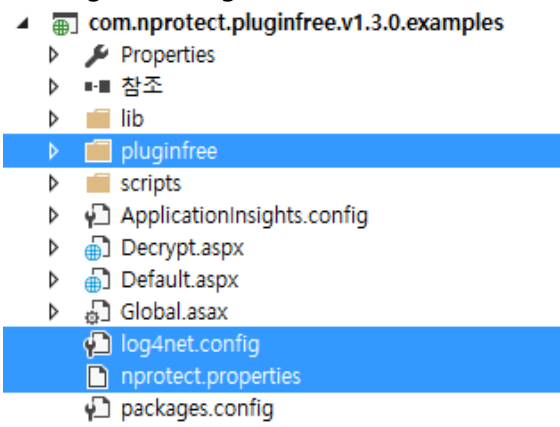
⁴⁰ 2016년 05월 23일 현재 .NET Framework 4.0만 지원합니다.

⁴¹ 마우스입력기의 이미지 및 설정을 변경하기 위해서는 별도의 dll 제작이 필요합니다. 잉카인터넷 기술지원부서과 사전 협의 후 진행하십시오.

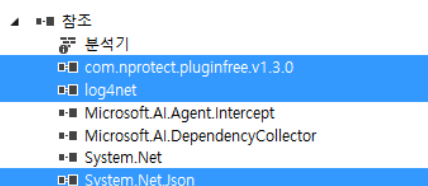
- 프로젝트 파일 목록에서 참조에서 오른쪽 마우스를 클릭 한 후 [참조 추가]를 선택합니다. 참조관리자 창에서 [찾아보기] 탭을 선택한 후 log4net.dll, System.Net.Json.dll, com.nprotect.pluginfree.v1.3.0.dll, com.nprotect.pluginfree.keypad.resources.dll 파일을 찾아 선택합니다.⁴²



- NOS 관련 웹 페이지(pluginfree) 및 정책 파일(nprotect.properties), 로그 설정 파일(log4net.config)을 프로젝트에 추가합니다.



⁴² 참조 목록에서 com.nprotect.pluginfree.v1.x.x, log4net, System.Net.Json 이 정상적으로 추가되었는지 확인합니다.



4. log4net.config 파일을 열어 log4net > appender > param > File 항목의 value 부분에 [로그 저장 경로]와 로그저장 레벨을 지정합니다. (지정하는 경로에서 ₩경로는 [₩₩로](#) 변경하여 지정해야 합니다.)

```
<!-- Logging Configuration -->
<log4net>
  <appender type="log4net.Appender.RollingFileAppender, log4net" name="RollingLogFileAppender">
    <param name="File" value="D:\\DEV - Eclipse\\DEV - NOS Plugin Free Service .NET\\logs\\asp.net.log" />
    <param name="AppendToFile" value="true" />
    <param name="MaxSizeRollBackups" value="10" />
    <param name="MaximumFileSize" value="5MB" />

  </appender>
  <root>
    <level value="DEBUG" />
    <appender-ref ref="RollingLogFileAppender" />
  </root>
```

5. 프로젝트의 Web.config 파일을 열어 appSettings 항목에 아래의 설정을 추가합니다. PropertiesPath 에서 ₩경로는 [₩₩로](#) 치환할 필요가 없습니다.

```
<appSettings>

  <add key="log4net.Config" value="log4net.config"/>
  <add key="log4net.Config.Watch" value="True"/>
  <add key="log4net.Internal.Debug" value="true"/>
  <add key="PropertiesPath" value="[프로젝트 경로]₩lib₩nprotect.properties"/>

</appSettings>
```

6. Global.aspx 파일을 열어 Application_Start 함수에 아래의 코드를 추가합니다.

```
protected void Application_Start(object sender, EventArgs e)
{
    if (WebConfigurationManager.AppSettings.Count > 0) {
        String propertiesPath = WebConfigurationManager.AppSettings["PropertiesPath"];
        if (null != propertiesPath && propertiesPath != "") {
            HttpContext.Current.Items["PluginFreePropertiesPath"] = propertiesPath;
            PluginFreeConfig config = PluginFreeConfig.GetInstance();
            config.init(PluginFreeProperties.GetInstance(propertiesPath));
        }
    }
}
```

5.2.2. 암호화 페이지

- 자바스크립트 파일(jQuery, nppfs.script.aspx, nppfs-1.x.x.js)이 참조되도록 아래의 구문을 추가합니다. 웹 페이지 로드 시 자동으로 시작될 수 있도록 npPfsStartup 함수 호출 구문을 추가합니다.
- 자세한 사항은 [4.3 페이지 적용](#) 단원을 참고하십시오.

```
<script type="text/javascript" src="/pluginfree/js/jquery-1.11.0.min.js"></script>
<script type="text/javascript" src="/pluginfree/asp/nppfs.script.aspx"></script>
<script type="text/javascript" src="/pluginfree/js/nppfs-1.9.0.js"></script>
<script type="text/javascript">
jQuery(document).ready(function(){
/*
    npPfsStartup(document.form1, false, true, true, "enc", "on");
    1. form 양식 : 기본값 DOM Document 상의 첫번째 form
    2. 개인방화벽 사용여부 : 기본값 false
    3. 키보드보안 사용여부 : 기본값 false
    4. 단말정보수집 사용여부 : 기본값 false
    5. 키보드보안 E2E 필드 설정 속성명 : 기본값 "enc"
    6. 키보드보안 E2E 필드 설정 속성값: 기본값 "on"
*/
    npPfsStartup(document.form, true, true, true, true, "npkencrypt", "on");
});
```

5.2.3. 복호화 페이지⁴³

- HttpRequest 기반 복호화방법.

```
PluginFreeRequest pfr = new PluginFreeRequest(Page.Request);
String cardNo1 = pfr["cardNo1"];
String cardNo2 = pfr["cardNo2"];
String cardNo3 = pfr["cardNo3"];
String cardNo4 = pfr["cardNo4"];
```

- Dictionary 기반 복호화 방법

```
Dictionary<String, String> map = RequestUtil.ToDictionary(Page.Request);
PluginFreeRequest pfr = new PluginFreeRequest(Page.Request, map);
String cardNo1 = pfr["cardNo1"];
```

⁴³ ASP.NET에서는 Java 기반 WAS의 Filter와 같은 자동복호화 방식은 구현되지 않았습니다. 수동 복호화방법을 적용해야 합니다.

```
String cardNo2 = pfr["cardNo2"];
String cardNo3 = pfr["cardNo3"];
String cardNo4 = pfr["cardNo4"];
```

- 단말정보수집 복호화 방법.

```
try {
    PluginFree.verifyFds(Page.Request);
} catch (PluginFreeException) {
    // "단말정보수집 복호화 검증 오류가 발생하였습니다."
    return;
}

PluginFree pf = new PluginFree(Page.Request);
PluginFreeDTO dto = pf.get();
if (dto != null) {
    // 정상적으로 수집되었습니다.
} else {
    // 값을 파싱할 수 없습니다.
}
```

5.3. Node.js 언어 지원

5.3.1. Node.js 프로젝트 설정

1. PluginFree 폴더를 프로젝트의 루트 경로에 복사합니다.

파일/폴더명	설명
PluginFree.js	- Node.js 용 확장 모듈
PluginFreeException.js	- 예외처리 관련 사용자 예외 코드
com.nprotect.pluginfree.v1.3.0.jar	- 키보드 보안 및 마우스입력기, 단말정보수집 관련 자바 모듈
nprotect.properties	- 정책 파일
install.html	- 설치 안내 페이지
idnex.html	- 테스트 예제
decrypt.html	- 복호화 예제
resources	- 마우스입력기 관련 이미지 및 설정 폴더

[표 5-3] Node.js 설치 파일 목록

2. Node.js 의 확장 모듈을 추가합니다.

확장 모듈명	설명
express	- 웹사이트(WAS) 처리관련 모듈
express-session	- 세션 처리관련 모듈
ejs	- 동적 웹페이지 작성 모듈
java	- 자바연동 관련 모듈
body-parser	- POST 데이터 파싱 모듈
cookie-parser	- 쿠키 파싱 모듈

[표 5-4] Node.js 확장 모듈 목록

3. 웹 서버의 정적 콘텐츠 위치에 모듈의 /public/pluginfree/ 폴더를 복사합니다.

4. WAS 의 상단에 확장 모듈을 정의합니다

```
var http = require("http");
var url = require("url");
var fs = require("fs");

var express = require("express");
var ejs = require("ejs");
var session = require("express-session");
var cookieParser = require("cookie-parser");
var bodyParser = require("body-parser");
var nppfs = require(__dirname + "/PluginFree/PluginFree");
```

5. WAS 에 키발급 및 마우스입력기 모듈과 복호화⁴⁴ 모듈을 등록합니다.

```
var app = express();
```

⁴⁴ 복호화 모듈 nppfs.filter(request, response) 를 추가하게 되면 자동으로 request.body 에 복호화된 값이 자동으로 변환되어 처리됩니다.

```
//app.use(app.router);
app.use(express.static(__dirname + "/public"));
app.use(cookieParser());
app.use(bodyParser.urlencoded({extended: true}));

app.use("/pluginfree/:mode", function (request, response, next) {
    var mode = request.params.mode;
    var result = nppfs.process(mode, request, response);
    //    next();
});
app.use(function (request, response, next) {
    nppfs.filter(request, response);
    next();
});
```

6. WAS 가 정상적으로 기동되면 최초 실행시에 아래와 같은 로그를 출력합니다.

```
Protect Plugin Free Service, [/Users/node.js/PluginFree/nprotect.properties] was found.
nProtect Plugin Free Service, Security initialized successfully.
nProtect Plugin Free Service, Public Key.
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAhZsc2aCswN+yvFekgidy
JCsTr4DMDgOdhdztzg0IRVpfpkihKqCDgJtpzVMjdDDHgnx1ljVVMFzFE2iw1M6A
d2UMhTmNijAvDPU18+KJ5amAS8SiqbYCdiY7Y4C6clBkvpAZXqXBq55VusZAF0I1
Um5QQ8qYxcVO7CMnUzoEOYGd7yjjelFQFmQO0xvA0TzFAQDzX0AsGsM7HFmMUHv4
Ybg/Vculq6lmuXitxMmgXoDzwqZvaX0EBZdLwLWvLDzkF27xsiC7ZdV0e2ynl6IZ
og7NNglgkYmkV53HzTtY2BQYa8tRESYSekiWECW6I5/TFA3Qz57PRxGEqSQ1Kyk/
QQIDAQAB
-----END PUBLIC KEY-----
nProtect Plugin Free Service, Disabled Country Code
    Use Country Code Site : http://ipc.nprotect.net/nppfs.country.jsp
nProtect Plugin Free Service, Use KeyPad
    KeyPad Resources Path : /Users/node.js/PluginFree/resources
    Default Theme : mobile
    Default Enter : HIDESELF
    Use Preview : false
    Use Image Resizing : true
    Position Type X : center
```

```
Position Pixel X : 0
Position Type Y : auto
Position Pixel Y : 5
```

5.3.2. 암호화 페이지

- 자바스크립트 파일(jQuery, nppfs-1.x.x.js)이 참조되도록 아래의 구문을 추가합니다. 웹 페이지 로드 시 자동으로 시작될 수 있도록 npPfsStartup 함수 호출 구문을 추가합니다.
- 자세한 사항은 [4.3 페이지 적용](#) 단원을 참고하십시오.

```
<script type="text/javascript" src="/pluginfree/js/jquery-1.11.0.min.js"></script>
<script type="text/javascript" src="/pluginfree/js/nppfs-1.9.0.js"></script>
<script type="text/javascript">
jQuery(document).ready(function(){
/*
    npPfsStartup(document.form1, false, true, true, "enc", "on");
    1. form 양식 : 기본값 DOM Document 상의 첫번째 form
    2. 개인방화벽 사용여부 : 기본값 false
    3. 키보드보안 사용여부 : 기본값 false
    4. 단말정보수집 사용여부 : 기본값 false
    5. 키보드보안 E2E 필드 설정 속성명 : 기본값 "enc"
    6. 키보드보안 E2E 필드 설정 속성값: 기본값 "on"
*/
    npPfsStartup(document.form, true, true, true, true, "npkencrypt", "on");
});
```

5.3.3. 복호화 페이지⁴⁵

- 복호화 데이터는 nppfs.filter 를 통하여 이미 request.body 에 복호화된 상태로 존재합니다. 값을 사용하시려면 request.body["입력양식이름"]을 사용하여 값을 얻을 수 있습니다. 복호화 페이지에 nppfs.verify 함수를 삽입하여 값이 변조되지 않았는지 체크합니다.

```
// 키보드보안/마우스입력기 검증
try {
    nppfs.verify(request, ["cardNo1", "cardNo2", "cardNo3", "cardNo4"]);
} catch(e){
```

⁴⁵ ASP.NET 에서는 Java 기반 WAS 의 Filter 와 같은 자동복호화 방식은 구현되지 않았습니다. 수동 복호화방법을 적용해야 합니다.

```
console.log(e.code, e.message);
}
```

- 단말정보수집 데이터는 nppfs.fds(request)를 통하여 JSON 형태로 반환됩니다. nppfs.verifyFds(request)함수를 통하여 값이 변조되지 않았는지 확인합니다. JSON 의 키 목록은 **[표 4-9] PluginFreeDTO 객체 구조** 표의 변수명입니다. 해당 표를 참고하십시오.

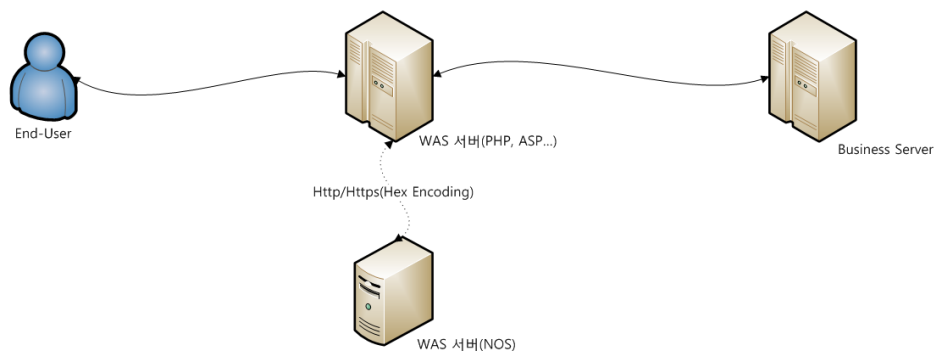
```
// 단말수집정보 검증
try {
    nppfs.verifyFds(request);
} catch(e){
    console.log(e.code, e.message);
}

// 단말수집정보
var deviceinfo = nppfs.fds(request);
```

5.4. 다른 개발 언어 지원

5.4.1. 구조 설명(Java기반 WAS를 통한) API⁴⁶

- 자바 기반의 WAS 서버를 구축한 다음에 기존 WAS 서버에서 자바기반 WAS 서버로 각종 값 및 명령을 호출하는 방식으로 구현됩니다.



서버구분	설명
------	----

⁴⁶ 2015 년 11 월 30 일 현재 PHP 언어만 지원합니다.

서버구분	설명
WAS 서버(PHP, ASP)	- 기존 웹사이트 WAS 서버 - WAS 서버(NOS)를 호출하도록 처리
Business 서버	- 기존 비즈니스 업무 서버 - WAS서버(NOS)에서 암호화된 데이터를 복호화
WAS 서버(NOS)	- Java기반 WAS에서 처리되던 로직은 처리 - 실제 사용자 UI는 없으며 WAS 서버(PHP, ASP)의 요청을 처리

[표 5-3] 데이터 재전송 방법

5.4.2. WAS 서버(NOS) 설정

- 서버 설정은 필터를 등록하는 과정을 제외하고는 자바기반의 WAS 서버와 동일합니다. 하지만, 본 기능은 필터를 사용하지 않고 별도의 자바 Servlet 으로 동작합니다.
- AllowRemoteAddress 값은 WAS 서버(PHP, ASP)서버의 IP 주소가 되며 WAS 서버(NOS)에서 바라보는 IP 주소를 입력합니다. 줄 단위로 입력해야 하며 IP 가 목록에 없을 경우에는 아무런 응답값을 주지 않습니다.
- Servlet-mapp WAS 서버(NOS)에 호출 가능한 사이트 경로로서 적절한 값을 지정합니다.

```

<servlet>
  <servlet-name>PluginFreeBroadcastServlet</servlet-name>
  <servlet-class>com.nprotect.pluginfree.broadcast.BroadcastServlet</servlet-class>
  <init-param>
    <param-name>ResponseEncoding</param-name>
    <param-value>UTF-8</param-value>
  </init-param>
  <init-param>
    <param-name>AllowRemoteAddress</param-name>
    <param-value>
      127.0.0.1
    </param-value>
  </init-param>
<!--
  <init-param>
    <param-name>PropertiesPath</param-name>
    <param-value>Some Path of Properties file\Wnprotect.properties</param-value>
  </init-param>
-->

```

```

        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>PluginFreeBroadcastServlet</servlet-name>
        <url-pattern>/nppfs.broadcast.do</url-pattern>
    </servlet-mapping>

```

5.4.3. WAS 서버(PHP) 설정

5.4.3.1. PHP WAS 서버 설정⁴⁷

1. 파일 nppfs.broadcast.inc 을 열어 최 하단의 코드를 적절한 값으로 변경합니다.

```
new BroadcastClient("http", "192.168.1.111", 8888, "/nppfs.broadcast.do");
```

2. nppfs.broadcast.inc 과 nppfs.broadcast.php 파일을 정적인 곳에 업로드 합니다.
3. 자바스크립트(nppfs-1.x.x.js)파일은 서블릿모드로 동작되도록 설정해야 하며 이때 참고하는 서블릿 경로는 nppfs.broadcast.php 경로이어야 합니다.
4. 데이터의 재 암호화는 다음의 코드를 참고하여 작성합니다.

```

<?php
session_start();

include_once "pluginfree/php/nppfs.broadcast.inc";

try {
    $_REQUEST = array_map('mysql_escape_string', $_REQUEST);
    $value1 = $_nppfs_broadcast_client->decrypt($_REQUEST, "FDS");
    foreach ($_REQUEST as $name => $value) {
        $value1 = $_REQUEST[$name];
        $value2 = $_nppfs_broadcast_client->decrypt($_REQUEST, "E2E", $name);
        echo("입력양식명 : " . $name . " => [" . $value1 . "]" . $value2 . ")<br />");
    }
} catch(Exception $ex) {

```

⁴⁷ PHP 의 session_start() 함수를 통하여 반드시 세션이 사용가능하도록 해야 합니다. WAS 서버(NOS)와의 세션유지를 위해 쿠키값을 PHP 의 세션에 저장합니다.

```

        echo $ex;
    }
?>

```

5.4.3.2. 업무서버에서의 복호화 사용

- 페이지 상단에 com.nprotect.pluginfree.broadcast.BroadcastUtil 을 포함시킵니다.
- 키보드 E2E 정보 복호화는 \$_nppfs_broadcast_client->(\$_REQUEST, "E2E", 입력양식명) 으로 반환 된 값을 BroadcastUtil.decrypt() 함수를 하여 처리합니다.
- 단발정보 복호화는 \$_nppfs_broadcast_client->(\$_REQUEST, "FDS") 으로 반환된 값을 BroadcastUtil.decrypt() 함수를 하여 처리합니다. 단말수집 정보의 경우에는 복호화된 값이 JSON 형태로 반환됩니다. 항목의 key 부분은 PluginFreeDTO 객체의 변수명으로 매핑되어 처리됩니다.

```

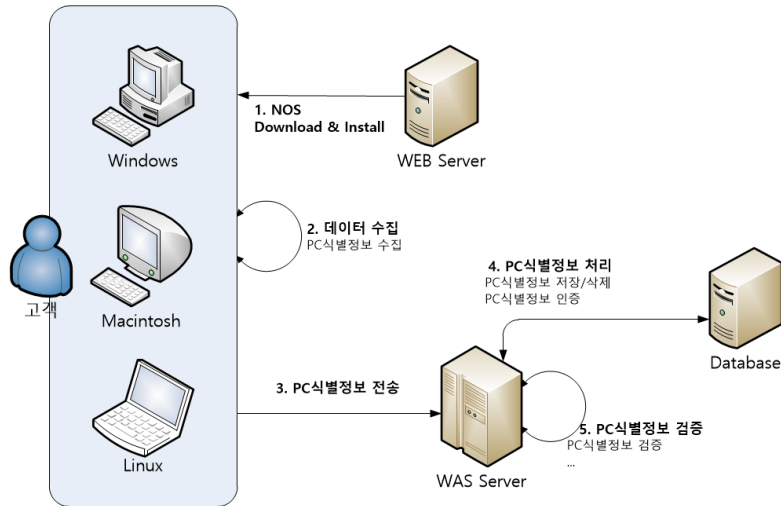
import "com.nprotect.pluginfree.broadcast.BroadcastUtil";
String value = BroadcastUtil.decrypt("암호화데이터");

```

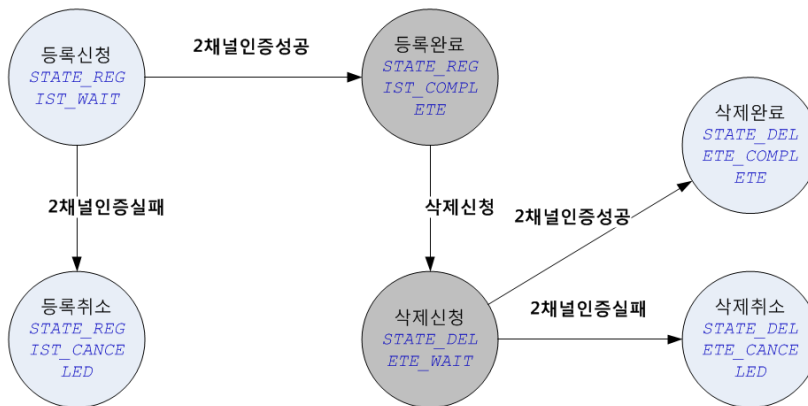
5.5. PC 지정서비스 개발가이드

5.5.1. PC지정서비스 처리 개요

5.5.1.1. PC 지정서비스 흐름도



5.5.1.2. PC 지정서비스 상태 천이도



5.5.1.3. PC 지정서비스 상태

상태	설명	비고
STATE_REGIST_WAIT	등록신청(10)	
STATE_REGIST_COMPLETE	등록완료(11)	PC 지정 인증가능상태
STATE_REGIST_CANCELED	등록취소(12)	
STATE_DELETE_WAIT	삭제신청(20)	PC 지정 인증가능상태
STATE_DELETE_COMPLETE	삭제완료(21)	
STATE_DELETE_CANCELED	삭제취소(22)	

5.5.2. PC지정서비스 식별 정보 및 WAS 서버 인증

5.5.2.1. PC 지정서비스 식별정보(PCSignupDTO 객체)

- PluginFree.getSignup() 함수를 사용하여 수집된 객체를 얻습니다.

변수명	설명	비고
String[] macEth ⁴⁸	Ethernet Mac 주소	
String osGuid	OS GUID 시리얼	
String hddSrial	HDD 시리얼	
String mbSrial	MB 시리얼	
String cpuld	CPU 시리얼	
String bootUuid	BOOT UUID	
Boolean virtualOs ⁴⁹	가상운영체제 여부	
String[] hashMacEth	Ethernet Mac 주소(해쉬)	SHA1 해쉬 자동처리
String hashOsGuid	OS GUID 시리얼(해쉬)	SHA1 해쉬 자동처리
String hashHddSrial	HDD 시리얼(해쉬)	SHA1 해쉬 자동처리
String hashMbSrial	MB 시리얼(해쉬)	SHA1 해쉬 자동처리
String hashCpuld	CPU 시리얼(해쉬)	SHA1 해쉬 자동처리
String hashBootUuid	BOOT UUID(해쉬)	SHA1 해쉬 자동처리

5.5.2.2. PC 지정서비스 WAS 서버 인증

인증조건	타입	설명
OS_GUID + HDD Serial	운영체제식별코드 + 하드디스크 시리얼	고유식별번호(Primary Key)
OS_GUID + MAC Addr 1	운영체제식별코드 + 맥주소 1	
OS_GUID + MAC Addr 2	운영체제식별코드 + 맥주소 2	
OS_GUID + MAC Addr 3	운영체제식별코드 + 맥주소 3	
OS_GUID + MB Serial	운영체제식별코드 + 마더보드 시리얼	

⁴⁸ Ethernet MAC Address 는 3 개가 수집됩니다. 네트워크상황에 따라 순서(네트워크가 직접 연결된 Ethernet 이 첫번째로 수집됨)가 다르게 수집됩니다. 데이터베이스 저장시에는 첫번째 값을 저장하고 비교시에는 3 개를 OR 조건으로 검증하는 로직으로 구현하십시오.

⁴⁹ 가상운영체제를 PC 지정서비스에 부합하는 PC 로 등록할지 말아야 할지는 해당 고객사의 보안 정책에 따라 구현하십시오. 가상운영체제를 통하여 Macintosh 나 Linux 시스템에서 가상운영체제를 Windows 를 설치하여 접속하는 사용자도 있으며 불법적인 사용자 또한 가상운영체제를 사용하는 경우가 있습니다.

인증조건	타입	설명
OS_GUID + CPU ID	운영체제식별코드 + CPU ID	
OS_GUID + BOOT UUID	운영체제식별코드 + 부트고유정보	리눅스환경에서만 지원

- 위 조건들은 OR 조건 또는 일치 대상을 한정(2 개)하여 처리합니다. 하드웨어 특성을 타는 정보는 미 수집이 발생가능하며 이미지 백업 프로그램(예, GHOST 등)으로 운영체제를 설치하는 경우에는 다른 PC 의 경우에도 동일한 PC 로 인식하는 경우가 발생합니다. 이런 경우 OR 조건보다는 2 개이상의 식별정보가 일치할 경우에 PC 의 인증을 하는 방법으로 구현하기를 권고합니다.
- [등록완료], [삭제신청] 상태에서만 인증이 가능한 상태로 처리합니다.

5.5.3. PC지정서비스 데이터베이스 테이블 설계

5.5.3.1. PC 지정 식별정보 테이블

컬럼명	타입	설명
NO	NUMBER NOT NULL	고유식별번호(Primary Key)
USER_ID	VARCHAR2(20) NOT NULL	고객식별정보(예, 사용자 ID)
SITE_CD ⁵⁰	CHAR(4) NOT NULL	사이트구분코드(예, 개인뱅킹 : 1000, 기업뱅킹 : 2000)
DEVICE	CHAR(2) NOT NULL	단말 구분정보(예, PC : 10, 모바일기기 : 20 등)
NICKNAME	VARCHAR2(30) NOT NULL	PC 구분별명(예, 집컴퓨터, 회사컴퓨터, 개인노트북 등)
IS_VIRTUAL_OS	CHAR(1) NOT NULL	가상운영체제 여부
HASH_OS_GUID	CHAR(40)	OS 식별정보(수집정보를 sha1 해쉬처리하여 저장, 40byte)
HASH_MAC	CHAR(40)	Mac 주소 정보(sha1 해쉬처리하여 저장, 40byte)
HASH_HDD	CHAR(40)	HDD Serial 정보(sha1 해쉬처리하여 저장, 40byte)
HASH_MB	CHAR(40)	MB Serial 정보(sha1 해쉬처리하여 저장, 40byte)
HASH_CPU	CHAR(40)	CPU ID 정보(sha1 해쉬처리하여 저장, 40byte)
HASH_UUID	CHAR(40)	Boot UUID(sha1 해쉬처리하여 저장, 40byte) – 리눅스용
STATE	CHAR(2) NOT NULL	등록상태(예, 등록신청대기 : 10, 등록완료 : 20 등)
REG_DATE	CHAR(8) NOT NULL	등록신청일/등록완료일(년월일)
REG_TIME	CHAR(6) NOT NULL	등록신청시각/등록완료시각(시분초)
DEL_DATE	CHAR(8)	삭제신청일/삭제완료일(년월일)

⁵⁰ 사이트구분, 단말정보 구분, PC 구분설명, 가상운영체제여부는 개발시 불필요할 경우 삭제하십시오. 본 문서에서 제공하는 개발 가이드보다는 각 업체별 업무로직 또는 스토리보드를 우선하여 적용하십시오.

컬럼명	타입	설명
DEL_TIME	CHAR(6)	삭제신청시각/삭제완료시각(시분초)

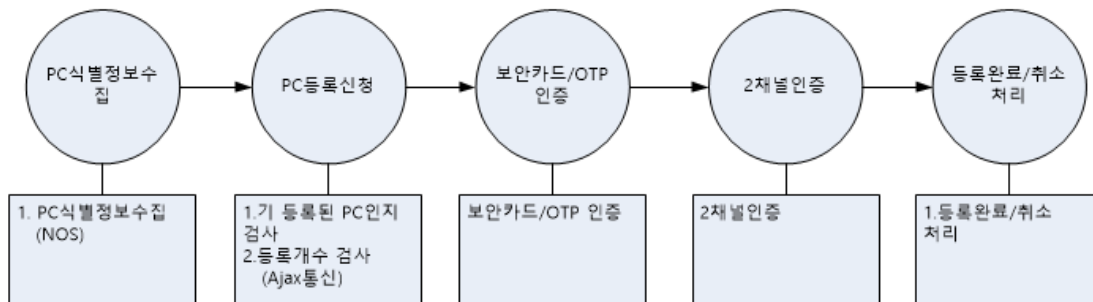
5.5.3.2. PC 지정 상태변경내역 테이블⁵¹

컬럼명	타입	설명
REG_DATE	CHAR(8) NOT NULL	상태변경일(년월일)
REG_TIME	CHAR(9) NOT NULL	상태변경시각(시분초밀리초)
NO	NUMBER NOT NULL	고유식별번호(Foreign Key)
USER_ID	VARCHAR2(20) NOT NULL	고객식별정보(예, 사용자 ID)
SITE_CD	CHAR(4) NOT NULL	사이트구분코드(예, 개인뱅킹 : 1000, 기업뱅킹 : 2000)
DEVICE	CHAR(2) NOT NULL	단말 구분정보(예, PC : 10, 모바일기기 : 20 등)
IS_VIRTUAL_OS	CHAR(1) NOT NULL	가상운영체제 여부
NICKNAME	VARCHAR2(30) NOT NULL	PC 구분별명(예, 집컴퓨터, 회사컴퓨터, 개인노트북 등)
HASH_OS_GUID	CHAR(40)	OS 식별정보(수집정보를 sha1 해쉬처리하여 저장, 40byte)
HASH_MAC	CHAR(40)	Mac 주소 정보(sha1 해쉬처리하여 저장, 40byte)
HASH_HDD	CHAR(40)	HDD Serial 정보(sha1 해쉬처리하여 저장, 40byte)
HASH_MB	CHAR(40)	MB Serial 정보(sha1 해쉬처리하여 저장, 40byte)
HASH_CPU	CHAR(40)	CPU ID 정보(sha1 해쉬처리하여 저장, 40byte)
HASH_UUID	CHAR(40)	Boot UUID(sha1 해쉬처리하여 저장, 40byte) – 리눅스용
STATE	CHAR(2) NOT NULL	등록상태(예, 등록신청대기 : 10, 등록완료 : 20 등)
REG_DATE	CHAR(8) NOT NULL	등록신청일/등록완료일(년월일)
REG_TIME	CHAR(6) NOT NULL	등록신청시각/등록완료시각(시분초)
DEL_DATE	CHAR(8)	삭제신청일/삭제완료일(년월일)
DEL_TIME	CHAR(6)	삭제신청시각/삭제완료시각(시분초)

⁵¹ 본 테이블은 고객문의 시 등록 및 삭제처리를 조회하기 위한 테이블입니다. 불필요할 경우에는 구현하지 않아도 됩니다.

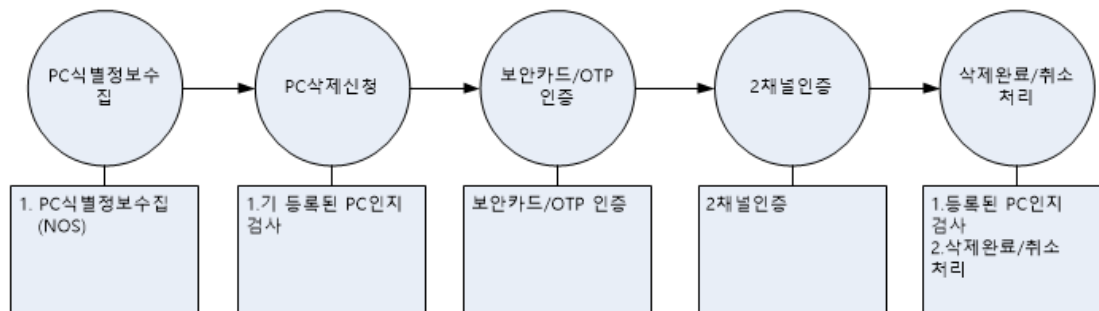
5.5.4. PC지정서비스 화면 절차 가이드

5.5.4.1. PC 등록 페이지 절차 가이드

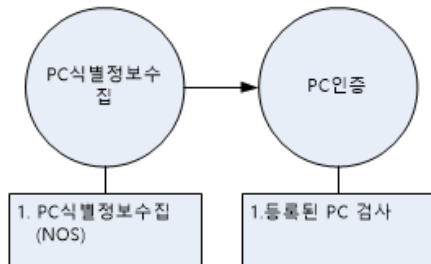


5.5.4.2. PC 삭제 페이지 절차 가이드

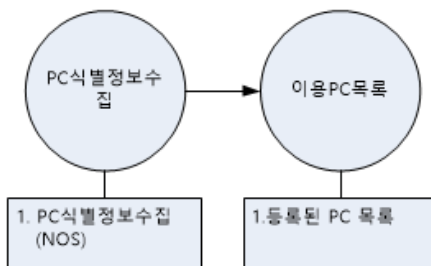
- 고객사 정책에 따라 삭제신청 시 등록된 PC에서만 삭제신청이 가능하도록 처리한 부분은 생략 가능합니다.



5.5.4.3. PC 목록 페이지 절차 가이드



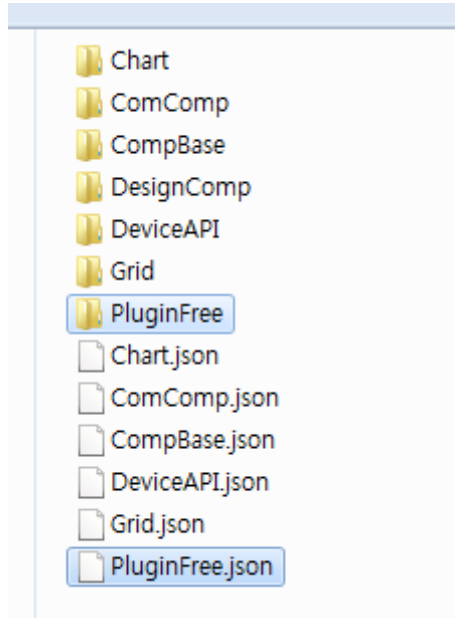
5.5.4.4. PC 인증 절차 가이드



5.6. 넥사크로플랫폼 적용가이드

5.6.1. 넥사크로 스튜디오 확장 라이브러리 추가

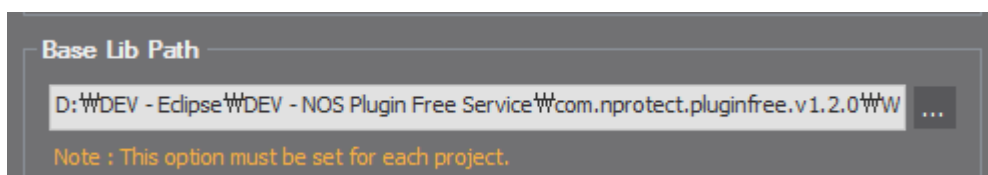
- 제공된 모듈을 다운로드 받아서 넥사크로플랫폼 라이브러리 경로⁵²에 복사합니다.



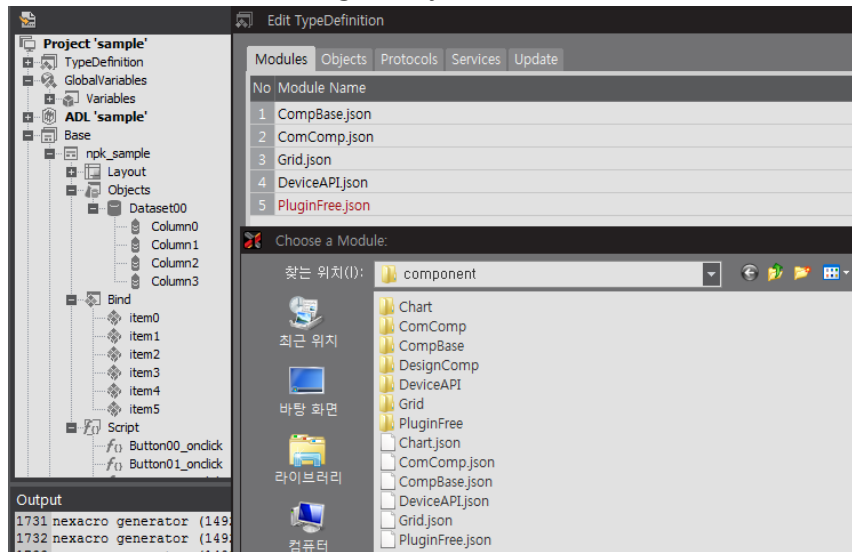
- 자바스크립트 파일 nppfs-1.x.x.js파일을 PluginFree 폴더에 복사합니다. PluginFree.json 파일을 텍스트편집기로 열어 nppfs-1.x.x.js의 파일명을 확인하고 상이할 경우 수정합니다.

```
{
  "name": "PluginFree",
  "version": "14.0.0.900",
  "description": "nProtect Online Security V1.0 for ...",
  "license": "",
  "scripts": [
    "PluginFree/nppfs.query-1.11.3.js",
    "PluginFree/nppfs.script.jsp",
    "PluginFree/nppfs-1.x.x.js",
    "PluginFree/nppfs.nexacro.plugin.js"
  ]
}
```

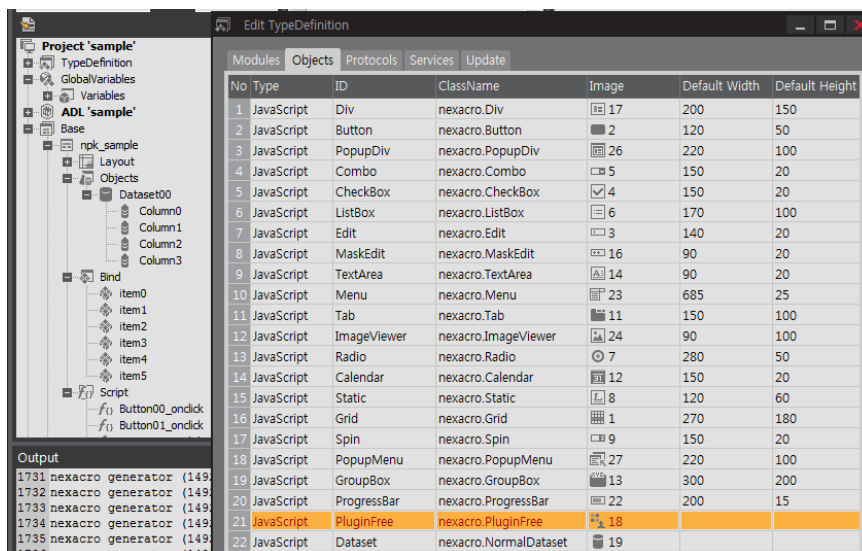
⁵² 넥사크로 스튜디오의 기본경로는 C:\Program Files (x86)\nexacro\14\nexacro14lib\component 입니다. 현재 사용하고 있는 라이브러리경로는 Tools > Options 에서 Base Lib Path 값을 확인하십시오.



- 프로젝트의 TypeDefinition을 더블 클릭하여 Edit Type Definition 창을 열고 Modules 탭을 선택합니다. Add 버튼을 눌러 PluginFree.json을 찾아 등록합니다.



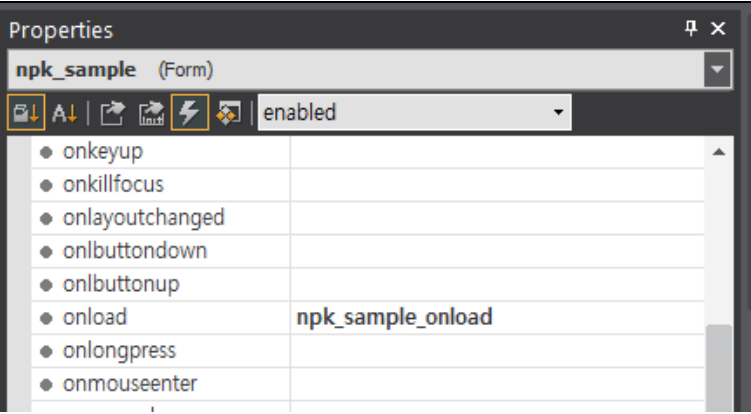
- 프로젝트의 TypeDefinition을 더블 클릭하여 Edit Type Definition 창을 열고 Objects 탭을 선택합니다. Add 버튼을 눌러 Javascript 타입으로 PluginFree, nexacro.PluginFree 값을 입력하여 등록합니다.



5.6.2. 키보드보안 및 마우스입력기 상세 설정⁵³

- 품을 선택하고 Properties의 이벤트에서 onload 이벤트 함수를 구현하고 npPfsStartup() 함수가 호출되도록 합니다.

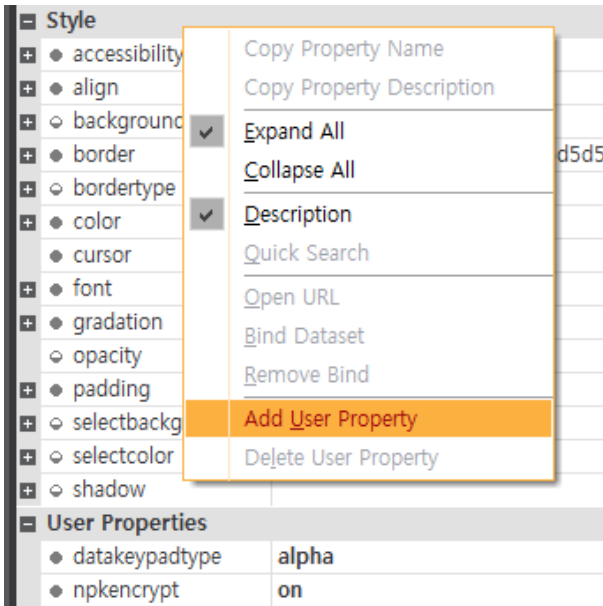
⁵³ 넥스코플랫폼의 MaskEdit 는 입력양식이 일반적인 웹입력양식으로 처리되지 않으며 특정한 규격으로 값을 강제적으로 변경하도록 동작되어 키보드보안 및 마우스입력기는 구동이 불가능합니다.



```

this.npk_sample_onload = function(obj:Form, e:nexacro.LoadEventInfo)
{
    npPfsStartup(null, true, true, true, true, "npkencrypt", "on");
}
    
```

- E2E를 적용하려는 필드를 선택한 뒤 입력양식의 Properties창에서 Add User Property를 선택합니다. npkencrypt 라는 속성이름을 신규로 생성합니다. E2E를 사용하는 필드의 경우에는 본 속성의 값을 on으로 지정하고 치환데이터를 사용하는 경우에는 re 값을 입력합니다.



- 마우스입력기의 노출형태를 변경하기 위해 아래와 같이 추가적인 속성을 추가해야 할 수도 있습니다. 각 속성에 대한 동작방식은 본 적용가이드 문서의 페이지적용 > 마우스입력기 상세설정 부분을 참고하십시오.

사용자정의 속성이름	입력양식 옵션명
datakeypadtype	data-keypad-type
datakeypadaction	data-keypad-action
datakeypadshow	data-keypad-show
datakeypadnext	data-keypad-next

datakeypadtheme	data-keypad-theme
datakeypadenter	data-keypad-enter
datakeypadinputrange	data-keypad-input-range
datakeypadpreview	data-keypad-preview
datakeypadtypex	data-keypad-type-x
datakeypadx	data-keypad-x
datakeypadtypey	data-keypad-type-y
datakeypady	data-keypad-y
datakeypaduseynotype	data-keypad-useyn-type
datakeypaduseyninput	data-keypad-useyn-input
datakeypadtoggleactive	data-keypad-toggle-active
datakeypadtoggleon	data-keypad-toggle-on
datakeypadtoggleoff	data-keypad-toggle-off

[표 5-3] 마우스입력기 구동관련 입력양식 옵션

- 치환테이블 및 치환데이터는 아래의 코드를 참고하여 적절하게 변경하여 처리해야 합니다. npPfsCtrl.findBindColumnName(입력양식의 ID값, 입력양식의 데이터바인딩 정보) 함수를 사용하여 지정한 ID에 대응하는 입력양식을 찾을 수 있습니다. npPfsCtrl.getResultField(공백문자, 입력양식명) 함수를 사용하여 치환 테이블값과 npPfsCtrl.GetReplaceField(공백문자, 입력양식명) 함수를 사용하여 호출하여 치환데이터 값을 얻습니다.

```

this.Button01_onclick = function(obj:Button, e:nexacro.ClickEventInfo)
{
npPfsCtrl.getResultField("", npPfsCtrl.findBindColumnName("Edit00", obj.parent.binds));
}

this.Button02_onclick = function(obj:Button, e:nexacro.ClickEventInfo)
{
npPfsCtrl.GetReplaceField("", npPfsCtrl.findBindColumnName("Edit00", obj.parent.binds));
}

```

5.6.3. WAS 서버⁵⁴로 데이터 전송 처리

5.6.3.1. WAS 서버로 데이터 송신 처리

- WAS서버에 데이터를 보내기 전에 동적으로 생성된 필드들을 DataSet에 등록합니다. 아래의 코드를 페이지에 맞게 수정하여 적용합니다. npPfsCtrl.serializeArray()⁵⁵ 함수를 사

⁵⁴ WAS 서버는 기본적으로 넥사크로플랫폼이 운영가능한 상태이어야 합니다. 자세한 사항은 투비소프트로 문의하십시오. 홈페이지 : <http://www.tobesoft.co.kr/>, 고객센터 : <http://support.tobesoft.co.kr/>

⁵⁵ Serialize 는 다음의 2 가지 방식을 지원합니다.

용하여 현재 페이지의 값들을 얻어옵니다. nosQuery.each함수를 사용하여 각각의 데이터를 DataSet에 추가합니다.

- 넥사크로플랫폼에서 데이터를 전송하는 방법은 Dataset과 DataSet (args) 방식이 있으며, 이중 하나의 방식을 선택하여 사용해야 합니다

● DataSet 방식을 사용하는 경우

```
this.Button00_onclick = function(obj:Button, e:nexacro.ClickEventInfo)
{
    var elements = npPfsCtrl.serializeArray();
    var _this = this;
    nosQuery(elements).each(function() {
        _this.Dataset00.addColumn(this.name, "string", 1024);
    });
    nosQuery(elements).each(function() {
        _this.Dataset00.setColumn(0, this.name, this.value);
    });
    trace(this.Dataset00.saveXML());
    this.transaction("save","SVC::/nexacro/nexaservice.jsp","Dataset00=Dataset00","", "", "fn_callback");
}
```

● DataSet(args) 방식을 사용하는 경우

```
this.Button00_onclick = function(obj:Button, e:nexacro.ClickEventInfo)
{
    var sId = this.Edit00.value;
    var sName = this.Edit01.value;
    var sPw = this.Edit02.value;
    var args = "GUBUN=" + nexacro.wrapQuote("SFA");
```

1. 특정 필드만 Serialize 처리
: 인자값으로 input 의 id 값을 배열로 지정합니다.

```
// 특정 필드만 serialize 할 경우
var elementsArray = [];
elementsArray.push(this.Edit00._input_element._input_handle.id);
elementsArray.push(this.Edit01._input_element._input_handle.id);
var elements = npPfsCtrl.serializeArray(elementsArray);
```

2. 모든 필드 Serialize 처리
: 인자값을 지정하지 않는 경우 모든 필드를 serialize 처리합니다.

```
// 전체 필드를 serialize 할 경우
var elements = npPfsCtrl.serializeArray();
```

```

args += "userid=" + nexacro.wrapQuote(sId);
args += "username=" + nexacro.wrapQuote(sName);
args += "password=" + nexacro.wrapQuote(sPw);

var elements = npPfsCtrl.serializeArray();
nosQuery(elements).each(function() {
    args += " " + this.name + "=" + nexacro.wrapQuote(this.value);
});
this.transaction("MyService01", "SVC::/nexacro/nexaservice.jsp", "input1=DS_SEARCH",
"Dataset00=output", args, "txCallback");
}

```

5.6.3.2. WAS 서버 데이터 수신 처리

- 전달받은 데이터를 java.util.Map으로 변환한 뒤 PluginFreeRequest 객체를 사용하여 복호화합니다. 각각의 값은 PluginFreeRequest.getParameter() 함수를 사용하여 복호화된 값을 얻을 수 있습니다.
- 넥사크로플랫폼에서 지원하는 데이터 Mapping 방식은 DataSetList와 VariableList 방식이 있으며 이중 하나를 선택하여 사용해야 합니다.

● 넥사크로플랫폼 라이브러리 및 복호화 관련 라이브러리 추가

```

// 관련라이브러리를 추가합니다.
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
%> <%@ page import="java.util.*"
%> <%@ page import="com.nexacro.xapi.tx.*"
%> <%@ page import="com.nexacro.xapi.data.*"
%> <%@ page import="com.nexacro.xapi.data.datatype.*"
%> <%@ page import="com.nprotect.pluginfree.*"
%> <%@ page import="com.nprotect.pluginfree.modules.PluginFreeRequest"
%> <%

```

● 넥사크로플랫폼 라이브러리 초기화 및 복호화용 변수 선언

```

// 넥사크로플랫폼 라이브러리를 사용하여 DataSet을 처리합니다.
HttpPlatformRequest platformRequest = new
HttpPlatformRequest(request,PlatformType.CONTENT_TYPE_XML,"utf-8");
platformRequest.receiveData();
PlatformData inPD = platformRequest.getData();

// 전달받은 데이터를 Map으로 변환합니다.

```

```
List<String> parameterNames = new ArrayList<String>();
Map<String, String> map = new HashMap<String, String>();
```

- 데이터 Mapping – VariableList 를 사용하는 경우

```
// VariableList를 사용하는 경우
VariableList inVariableList = inPD.getVariableList();
List keyList = inVariableList.keyList();
if(keyList != null && keyList.size() > 0) {
    System.out.println("받은 값(VariableList) ");
    for(int index = 0 ; index < keyList.size() ; index++) {
        String name = (String)keyList.get(index);
        String value = inVariableList.getString(name);
        System.out.println("받은 값 : " + header.getName() + " : "+ inDS.getString(0, index));
        map.put(name, value);
        parameterNames.add(name);
    }
}
```

- 데이터 Mapping – DataSetList 를 사용하는 경우

```
// DataSetList를 사용하는 경우
DataSetList inDataSetList = inPD.getDataSetList();
DataSet inDS = inDataSetList.get("Dataset00");
int columnCount = inDS.getColumnCount();
if(columnCount > 0) {
    System.out.println("받은 값(DataSetList) ");
    for(int index = 0 ; index < columnCount ; index++) {
        ColumnHeader header = inDS.getColumn(index);
        System.out.println("받은 값 : " + header.getName() + " : "+ inDS.getString(0, index));
        map.put(header.getName(), inDS.getString(0, index));
        parameterNames.add(header.getName());
    }
}
```

- PluginFree 복호화 객체 생성 및 데이터 위변조 여부 검사

```
// PluginFree 복호화 객체를 생성합니다.
HttpServletRequest req = new PluginFreeRequest(request, map);

// 키보드보안/마우스입력기의 데이터위변조가 없는지 확인합니다.
try {
    PluginFree.verify(req, new String[]{"cardNo1"});
}
```

```

} catch(Exception e) {
    System.out.println("키보드보안/마우스입력기 복호화 검증 오류가 발생하였습니다.<br />");
}

// 단말정보 수집에 데이터 위변조가 없는지 확인합니다.
try {
    PluginFree.verifyFds(req);
} catch(PluginFreeException e) {
    System.out.println("단말정보수집 복호화 검증 오류가 발생하였습니다.");
    return;
}

```

- 단말수집정보 얻기 (단말정보수집기능을 사용하는 경우에만 적용)

```

// 단말정보 수집은 new PluginFree객체의 get()함수를 사용하여
// PluginFreeDTO객체를 반환받습니다.
PluginFree pf = new PluginFree(req);
PluginFreeDTO dto = pf.get();
if(dto != null){
    System.out.println(dto.toString());
} else {
    System.out.println("값을 파싱할 수 없습니다.");
}

```

- 복호화 데이터 출력

```

// 복호화된 데이터를 출력합니다.
Collections.sort(parameterNames);
for(String p : parameterNames) {
    System.out.println("복호화 : " + p + " : " + req.getParameter(p));
}

```

- 복호화 데이터 로그 확인

```

// 출력되는 로그는 다음과 같습니다.
받은값 : Column0 : aaaaaaa
받은값 : Column2 : 111111
받은값 : i_borun : 20bafa25e3514f6.....
받은값 : i_e2e_id : 001449736354589
받은값 : Column2__E2E__ : 6312ba054.....
받은값 : Column0__E2E__ : 5581a7bdb9002c02821ed1.....
.....
복호화 : Column0 : abcdefg

```

복호화 : Column0_E2E_ : 5581a7bdb9002c02821ed1.....

복호화 : Column2 : 123456

복호화 : Column2_E2E_ : 6312ba054.....

복호화 : i_borun : 01

.....

복호화 : i_version : 2015.07.24.01

----- 단말정보수집 네트워크 정보 -----

NAT IP : 210.217.175.101

NAT IP 획득 에러 코드 : null

.....

단말정보수집모듈 동작여부 : Y

5.6.4. 넥사크로플랫폼 추가 Javascript API

5.6.4.1. 동적 생성 필드제거(destoryField)

- 동적으로 생성된 필드를 제거하기 위해 npPfsCtrl.destroyField() 함수를 사용합니다.

- 특정 필드만 삭제 처리

```
// 특정 동적 필드를 destroy 할 경우
var elementsArray = [];
elementsArray.push(this.Edit00._input_element._input_handle.id);
elementsArray.push(this.Edit01._input_element._input_handle.id);
npPfsCtrl.destroyFields(elementsArray);
```

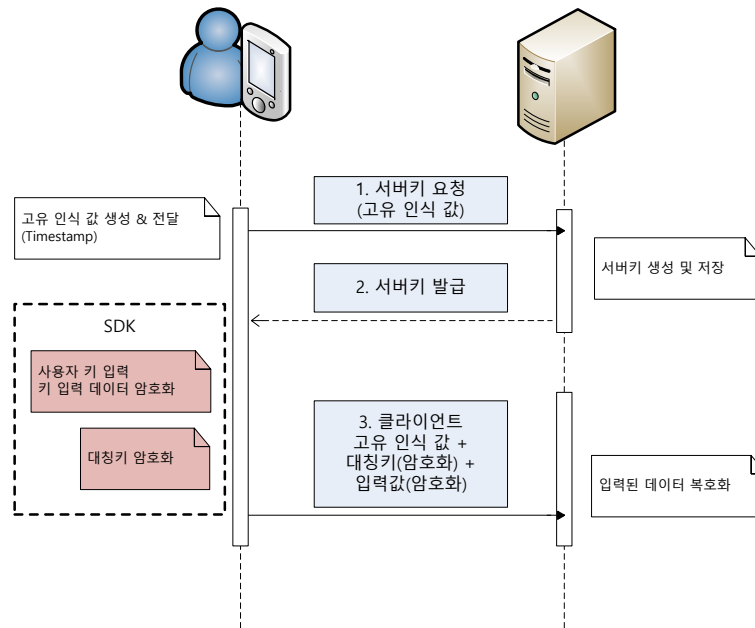
- 모든 필드 삭제처리

```
// 전체 동적 필드를 destroy 할 경우
npPfsCtrl.destroyFields();
```

6. 모바일 적용 가이드

6.1. 모바일 보안 키패드 적용가이드

6.1.1. 개요



1. WAS 기동 시에 RSA 키 쌍을 생성합니다.
2. 클라이언트의 요청 시 임의의 서버 랜덤 값을 생성하고 이 키를 RSA 공개키와 함께 암호화하여 클라이언트에 전송합니다.
3. 클라이언트가 받은 서버 키를 보안 키패드 SDK 에 전달하면 SDK 내부에서 파싱 후 서버에서 생성한 서버 랜덤 값과 SDK 내부에서 생성된 클라이언트 랜덤 값을 혼합하여 [데이터 암호화용 키]를 생성합니다.
4. 생성된 클라이언트 랜덤 값은 RSA 공개키로 암호화합니다.
5. 보안 키패드를 통한 키 입력 데이터는 매번 암호화 되어 저장됩니다.
6. 사용자의 입력 완료 후 [암호화된 키 입력 데이터] 와 [암호화된 클라이언트 랜덤 키] 를 서버로 전달합니다.
7. 서버는 공개키로 암호화된 클라이언트 랜덤 키를 RSA 개인키로 복호화하고 이 것을 서버 랜덤 키를 혼합하여 암호키를 구성하고 이로 사용자의 입력 데이터를 복호화합니다.

6.1.2. Android 적용 가이드

6.1.2.1. 라이브러리 설치 및 설정

1. 라이브러리 및 리소스 파일 복사

1. 제공하는 라이브러리 파일 모두를 해당 프로젝트의 libs 폴더에 복사해 넣습니다.
라이브러리 목록: keycrypt.jar, libIncaCommon.jar, npcrypto.jar, libnpcf_ma_u.so
2. 제공하는 리소스 파일들을 res 폴더의 적절한 위치에 복사해 넣습니다.
 - i. values 폴더 내의 파일들은 res/values 폴더에, values-ko 폴더 내의 파일들은 res/values-ko 폴더에 복사합니다. 해당 폴더가 없는 경우 새로 생성하여 복사합니다.
 - ii. zix_custom_input.xml 파일을 res/layout 폴더에 복사합니다.
3. 제공하는 이미지 파일들을 프로젝트 폴더의 res/drawable 폴더에 복사합니다.

2. AndroidManifest 설정

1. AndroidManifest.xml 파일에 아래의 권한을 추가합니다.

```
<uses-permission android:name="android.permission.INTERNET">
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE">
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE">
<uses-permission android:name="android.permission.READ_PHONE_STATE">
```

2. 보안 키패드 액티비티를 아래와 같이 추가합니다.

```
<activity android:name=" com.nprotect.keycryptm.IxCustomInputActivity"
android:configChanges="keyboardHidden|orientation|screenSize">
```

6.1.2.2. 코드 작성 예시

1. 라이선스 설정

제공받은 "라이선스 키"와 "고객사 ID"를 설정합니다. 가능하면 애플리케이션에서 한번 호출하도록 합니다.

예) Application 클래스나 메인 Activity 의 onCreate() 안에서 호출

```
public class ApplicationClass extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        IxSecureManager.initLicense(this, "LICENSE", "CUSTOMERID"); // 라이선스 설정
    }
}
```

```
}
```

2. Helper 객체 생성 및 각 입력창 설정

1. 보안 키패드 Helper 클래스인 `IxKeypadManageHelper` 클래스 객체를 생성
2. 키패드 콜백 리스너 등록 (상태 변화 콜백)
3. 각 입력창 관련 설정 (키패드 종류, 최대 입력 길이 등)

```
// 키패드 헬퍼 객체 생성
keypadMngHelper = new IxKeypadManageHelper(this, REQUEST_CODE_KEYPAD);

// 키패드 콜백 리스너 등록
keypadMngHelper.setSecureKeypadEventListener(
    new IxKeypadManageHelper.SecureKeypadEventListener() {

        @Override /** 키패드 높이 반환 */
        public void onKeypadChangeHeight(int height) {
        }

        @Override /** 입력 제한 정책 위반 콜백 */
        public void onInputViolationOccured(int errorCode) {
        }

        @Override /** 포커싱 변환 콜백 */
        public void onChangeEditText(EditText editText) {
        }

    });

// 입력창 설정
inputConfigs = new IxConfigureInputItem[2];

edtText1 = (EditText) findViewById(R.id.edt_input1);
IxConfigureInputItem inputConfig = new IxConfigureInputItem(edtText1,
    Defines.KEYPAD_TYPE_QWERTY, Defines.SHUFFLE_TYPE_GAPKEY);
inputConfig.setMaxLength(10); // 최대 입력 가능 길이 설정
inputConfigs[0] = inputConfig;

.....

keypadMngHelper.configureInputBoxes(inputConfigs); // 입력창 설정
```

4. `onActivityResult()` 메소드에 아래와 같이 `IxKeypadManageHelper` 의 `processResults()` 메소드 호출해줍니다.

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if ( keypadMngHelper.processResults(requestCode, resultCode, data) ) {
        return;
    }
}

```

3. 입력 데이터 수집 및 전송

입력 데이터를 모바일 기기 내에서만 사용할 수도 있고 서버와 연동해서 사용할 수도 있습니다. 이에 따라 입력 데이터 처리하는 코드가 다릅니다.

1. 모바일 기기 내에서만 사용하는 경우 (Stand Alone 모드)

입력 완료 후 복호화 된 입력 데이터를 받을 시 아래와 같이 `getRealText()` 메소드를 사용합니다.

```
String result_edtText1 = keypadMngHelper.getRealText(edtText1);
```

2. 서버 연동해서 사용하는 경우 (E2E 모드)

- i. 미리 서버에서 키를 가져와서 `setServerKey()` 메소드로 설정합니다.
- ii. 보안 키패드 입력 완료 후 아래와 같이 `getInputResults()` 메소드를 사용하여 서버기로 암호화 된 입력 데이터 객체(`IXResultItem`)를 전달 받습니다. (`IXResultItem` 클래스는 API 항목 참조)
- iii. 암호화된 입력 데이터와 키를 정해진 메시지 형태로 구성하여 서버로 전달합니다.
주의) 서버로 전송할 때까지 서버에서 키를 가져올 때의 세션을 유지하고 있어야 합니다.

```
IXResultItem[] resultItem = keypadMngHelper.getInputResults();
```

6.1.2.3. API

1. IxSecureManager 클래스

IxSecureManager	SDK 관리 클래스
package	com.nprotect
initLicense	라이선스 설정
Syntax	public static boolean initLicense (Context ctx, String license,

	String customerId)	
Parameter 1	Context	Activity 또는 Application의 Context 객체
Parameter 2	String	라이선스
Parameter 3	String	고객사 식별값
return	boolean	라이선스 유효 여부
getVersion	라이브러리 버전 (정수형)	
Syntax	public static final int getVersion();	
return	int	라이브러리 버전 (정수형)
getVersionString	라이브러리 버전 (문자열)	
Syntax	public static final String getVersionString();	
return	String	라이브러리 버전 (문자열)
checkRooting	루팅 상태 체크	
Syntax	public static boolean checkRooting();	
return	boolean	루팅 여부

2. IxKeypadManageHelper 클래스

IxKeypadManageHelper	보안 키패드 관리 헬퍼 클래스	
package	com.nprotect.keycryptm	
IxKeypadManageHelper	보안 키패드 헬퍼 객체 생성	
Syntax	public IxKeypadManageHelper(Context context, int requestCode);	
Parameter 1	Context	보안 키패드 헬퍼를 사용하는 Activity 객체
Parameter 2	int	보안 키패드 호출한 결과를 받기 위한 응답코드
setSecureKeypadEventListener	보안 키패드 이벤트 리스너 등록 참고) 해당 이벤트 리스너를 사용한다면 IKeypadHelperListener 인터페이스를 사용할 필요 없음	
Syntax	public void setSecureKeypadEventListener(SecureKeypadEventListener listener);	
Parameter 1	SecureKeypadEventListener	보안 키패드 이벤트 리스너 객체
configureInputBoxes	보안 키패드 적용 입력 창 설정	
Syntax	public void configureInputBoxes(IxConfigureInputItem[] configs);	
Parameter 1	IxConfigureInputItem[]	설정 객체 리스트
configureInputBoxes	보안 키패드 적용 입력 창 설정	
Syntax	public void configureInputBoxes(Activity context, int[] rsclids, int[]	

	maxLens);	
Parameter 1	Activity	호출하는 Activity
Parameter 2	Int[]	입력창 리소스 ID 배열
Parameter 3	Int[]	입력 가능 최대 개수 배열
configureInputBoxes	보안 키패드 적용 입력 창 설정	
Syntax	public void configureInputBoxes(Activity context, int[] rsclDs, int[] maxLens, int inputType);	
Parameter 1	Activity	호출 Activity
Parameter 2	Int[]	입력창 리소스 ID 배열
Parameter 3	Int[]	입력 가능 최대 개수 배열
Parameter 4	Int	보안 키패드 타입 (IConfigureInputItem 클래스 내용 참고)
configureInputBoxes	보안 키패드 적용 입력 창 설정	
Syntax	public void configureInputBoxes(Activity context, int[] rsclDs, int[] maxLens, int[] inputTypes);	
Parameter 1	Activity	호출하는 Activity
Parameter 2	Int[]	입력창 리소스 ID 배열
Parameter 3	Int[]	입력 가능 최대 개수 배열
Parameter 4	Int[]	보안 키패드 타입 배열 (IConfigureInputItem 클래스 내용 참조)
setUiVisibility	화면 구성 설정 (기본값: 상단 Indicator 보임)	
Syntax	public void setUiVisibility(int value);	
Parameter 2	int	화면 구성 플래그 Defines.FLAG_UI_NO_INDICATOR: 화면 상단 Indicator 숨김
setAutoHidden	최대 입력 값 도달 시 키패드 자동 숨김 설정 (기본값: 자동 숨김)	
Syntax	public void setAutoHidden(boolean value);	
Parameter 2	boolean	true: 자동 숨김 (기본값이므로 호출할 필요 없음) false: 자동 숨김 OFF
setCryptoPlugin	암호화를 위한 암호화 플러그인 설정 (Standalone 전용)	
Syntax	public void setCryptoPlugin(ICryptoPlugin plugin);	
Parameter 1	ICryptoPlugin	ICryptoPlugin 객체 (해당 클래스 내용 참고)
setCryptoPlugin	암호화를 위한 암호화 플러그인 설정 (Standalone 전용)	

Syntax	public void setCryptoPlugin(int pluginId);	
Parameter 1	int	암호화 플러그인 지시자
setServerKey	서버에서 발급받은 서버 키 설정	
Syntax	public void setServerKey(String key);	
Parameter 1	String	서버에서 발급받은 서버 키
Throws	-	InvalidServerKeyException, CryptoOperationException 서버키 파싱 오류
getInputResults	입력 결과 객체 리스트 반환	
Syntax	public IxResultItem[] getInputResults();	
return	IxResultItem[]	입력 결과 리스트
getInputResult	입력 결과 객체 반환	
Syntax	public IxResultItem getInputResult(int rsclId);	
Parameter 1	int	입력창 리소스 ID
return	IxResultItem	입력 결과
getRealText	암호화된 입력 데이터를 복호화하여 반환	
Syntax	public static String getRealText(EditText edittext);	
Parameter 1	EditText	해당 입력 창
return	String	복호화 된 입력 문자열 주의) E2E 모드에서는 null 반환
getRealText	암호화된 입력 데이터를 복호화하여 반환	
Syntax	public String getRealText(Activity context, int rsclId);	
Parameter 1	Activity	호출 Activity
Parameter 2	int	입력창 리소스 ID
return	String	복호화 된 입력 문자열 주의) E2E 모드에서는 null 반환
getFocusedInputBox	현재 포커싱되고 있는 입력 창 반환	
Syntax	public EditText getFocusedInputBox();	
return	EditText	포커싱 중인 입력 창
onFocusEditText	입력 창 포커싱 변화 시 현재 포커싱 된 입력 창 전달	
Syntax	public void onFocusEditText (EditText edittext);	
Parameter 1	EditText	포커싱 되는 입력 창
setCurrentInputContext	보안 키패드 Context 객체 설정	
Syntax	public void setCurrentInputContext(Context ctx);	
Parameter 1	Context	보안 키패드 Context
getCurrentInputContext	보안 키패드 Context 객체 전달	

Syntax	public Context getCurrentInputContext();	
return	Context	보안 키패드 Context
hideSecureKeypad	보안 키패드 숨기기	
Syntax	public void hideSecureKeypad();	
processResults	보안 키패드 종료 후 결과 정리 주의) IxKeypadManageHelper 를 사용하고 있다면 보안 키패드 적용한 액티비티의 onActivityResult() 내에서 필히 호출	
Syntax	public boolean processResults(int requestCode, int resultCode, Intent data);	
Parameter 1	int	보안 키패드로부터 반환된 응답코드
Parameter 2	int	보안 키패드로부터 반환된 결과코드
Parameter 3	Intent	보안 키패드로부터 반환된 intent 데이터
return	boolean	true: 결과 정리 성공
postProclnputs	입력 데이터 후처리	
Syntax	public boolean postProclnputs();	
return	boolean	입력 데이터 존재 여부
isExistAllInputs	모든 입력 데이터 존재 여부 확인	
Syntax	public boolean isExistAllInputs();	
return	boolean	모든 입력 데이터 존재 여부
validateInputs	입력 값 내에 위반 문자열 존재 여부 확인 주의) 이 메소드는 Stand Alone 모드에서 사용 가능 E2E 로 사용할 시 repackResultsWithServerKey() 메소드를 이용하여 서버키로 재 암호화 필요	
Syntax	public int validateInputs (EditText editText, ArrayList<String> patterns)	
Parameter 1	EditText	확인하려는 입력 창
Parameter 2	ArrayList	위반 문자열 리스트
return	int	위반 문자열 검출 시 검출된 문자열 인덱스 -1: 위반 문자열 미 발견 시
compare	두 입력 창의 입력 값 비교 주의) 이 메소드는 Stand Alone 모드에서 사용 가능 E2E 로 사용할 시 repackResultsWithServerKey() 메소드를 이용하여 서버키로 재 암호화 필요	
Syntax	public int compare(EditText editText1, EditText editText2)	
Parameter 1	EditText	첫 번째 입력 창
Parameter 2	EditText	두 번째 입력 창
return	int	0: 같음, 그 외: 다름
repackResultWithServerKey	입력 결과를 서버 키로 재암호화	

	Stand Alone 모드에서 암호화 된 입력 데이터를 서버 키로 재암호화	
Syntax	public IActionResult repackResultWithServerKey(EditText editText, String serverKey)	
Parameter 1	EditText	서버키로 패키징된 입력 결과를 받으려는 입력 창
Parameter 2	String	E2E 서버에서 발급받은 서버 키
return	IActionResult	서버 키로 패키징된 IActionResult 객체
repackResultsWithServerKey	모든 입력 결과를 서버 키로 재암호화 Stand Alone 모드에서 암호화 된 입력 데이터를 서버 키로 재암호화	
Syntax	public IActionResult[] repackResultsWithServerKey(String serverKey)	
Parameter 1	String	E2E 서버에서 발급받은 서버 키
return	IActionResult[]	서버 키로 패키징된 IActionResult 객체 리스트
moveToPrevInputBox	지정된 이전 입력 창으로 이동	
Syntax	public EditText moveToPrevInputBox()	
return	EditText	이동한 뒤의 입력 창
moveToNextInputBox	지정된 다음 입력 창으로 이동	
Syntax	public EditText moveToNextInputBox()	
return	EditText	이동한 뒤의 입력 창
setExtraToolbar	추가 커스텀 툴바 뷰 설정	
Syntax	public void setExtraToolbar(View extraToolbar)	
Parameter 1	View	추가 커스텀 툴바 뷰 객체

3. IxKeypadManageHelper.SecureKeypadEventListener 인터페이스

SecureKeypadEventListener	보안 키패드 이벤트 리스너 인터페이스 (IxKeypadManageHelper 클래스의 Inner 인터페이스)	
package	com.nprotect.keycryptm	
onKeypadChangeHeight	보안 키패드 높이 변화 시 호출	
Syntax	void onKeypadChangeHeight(int height);	
Parameter 1	int	보안 키패드의 높이
onChangeEditText	입력창 포커싱 변화 시 호출	
Syntax	void onChangeEditText(EditText edittext);	
Parameter 1	EditText	현재 포커싱 된 입력 창
onInputViolationOccured	키 입력 값의 제한 규칙 위반 시 호출	
Syntax	void onInputViolationOccured(int errorCode);	
Parameter 1	int	제한 규칙 위반 코드 (위반한 제한 규칙 그대로 반환) setInputCheckPolicy() 메소드 내용 참조

4. IxConfigureInputItem 클래스

IxConfigureInputItem	입력 창 설정 객체 클래스	
package	com.nprotect.keycryptm	
IxConfigureInputItem	입력 창 설정 객체 생성	
Syntax	public IxConfigureInputItem(EditText edittext, int inputType)	
Parameter 1	EditText	입력 창 객체
Parameter 2	int	보안 키패드 타입 0: 쿼티 키패드 1: 숫자 키패드
IxConfigureInputItem	입력 창 설정 객체 생성	
Syntax	public IxConfigureInputItem(EditText edittext, int inputType, int shuffleType)	
Parameter 1	EditText	입력 창 객체
Parameter 2	int	보안 키패드 타입 KEYPAD_TYPE_QWERTY: 쿼티 키패드 KEYPAD_TYPE_NUMBER: 숫자 키패드
Parameter 3	int	Qwerty 키패드의 셔플 타입 0 : 키셀 확장형 키패드(default) 1 : 마스크 이미지 삽입형 키패드 2 : 키 갭 삽입형 키패드
getInputBox	입력 창 설정 객체에서 입력 창 객체 반환	
Syntax	public EditText getInputBox()	
return	EditText	입력 창 객체
setMinLength	입력문자 최소 개수 설정	
Syntax	public void setMinLength(int minLength)	
Parameter 1	int	입력문자 최소 개수
setMaxLength	입력문자 최대 개수 설정	
Syntax	public void setMaxLength(int maxLength)	
Parameter 1	int	입력문자 최대 개수
setInputPolicy	입력 값 제한 정책 설정	
Syntax	public void setInputPolicy(int inputPolicy)	
Parameter 1	int	적용할 제한 정책 CHECKRULE_NONE: 정책 설정하지 않음 CHECKRULE_ALLOW_NUMBER: 숫자 입력 가능

		CHECKRULE_ALLOW_ALPHABET 알파벳 입력 가능 CHECKRULE_ALLOW_SPECIAL 특수문자 입력 가능 CHECKRULE_ALLOW_ALL 모든 문자 입력 가능 CHECK_POLICY_DUPLICATED_2CHAR: 연속 2개 같은 글자 입력 불가 예) "aa", "CC" CHECK_POLICY_DUPLICATED_3CHAR: 연속 3개 같은 글자 입력 불가 예) "aaa", "CCC" CHECK_POLICY_CONTINUOUS_3CHARS: 3개 이상의 연속된 글자 입력 불가 예) "abc", "zyx", "123" CHECK_POLICY_CONTINUOUS_4CHARS: 4개 이상의 연속된 글자 입력 불가 예) "1234", "abcd", "zyxw"
--	--	---

5. IxResultItem 클래스

IxResultItem	입력 데이터 결과 클래스	
package	com.nprotect.keycryptm	
getKey	입력 데이터 암호화 키 반환	
Syntax	public String getKey();	
return	String	입력 데이터 암호화 키
getEncData	암호화 된 입력 데이터 반환	
Syntax	public String getEncData();	
return	String	암호화 된 입력 데이터
getEncDataLen	입력 데이터 길이 반환	
Syntax	public int getEncDataLen();	
return	int	입력 데이터 길이
getMasterKey	마스터 키 반환 (Crypto Plugin 사용 시에만 사용)	
Syntax	public byte[] getMasterKey();	
return	int	입력 데이터 길이

6. IxCryptoPlugin 클래스

IxCryptoPlugin	암호화 플러그인 클래스 (E2E 사용시에는 필요치 않으며, 구현 시 별도의 샘플 제공)
package	com.nprotect.keycryptm.plugins

7. IKeypadHelperListener 인터페이스

IKeypadHelperListener	보안 키패드 콜백 리스너 인터페이스	
package	com.nprotect.keycryptm	
onTouchFromKeypadView	보안 키패드의 터치 이벤트 전달 (콜백) 주의) 뷰 모드 적용 시 해당 메소드 내에서 dispatchTouchEvent() 를 필히 호출	
Syntax	public void onTouchFromKeypadView(MotionEvent e);	
Parameter 1	MotionEvent	터치 모션 이벤트
setCurrentInputContext	보안 키패드의 Context 객체 전달 (콜백) 주의) IKeypadManageHelper 를 사용하고 있다면 IKeypadManageHelper 의 setCurrentInputContext() 를 필히 호출	
Syntax	public void setCurrentInputContext(Context ctx);	
Parameter 1	Context	보안 키패드 Context 객체
syncViewChangeHeight	보안 키패드 높이 변화 시 호출 (콜백)	
Syntax	public void syncViewChangeHeight(int height)	
Parameter 1	int	보안 키패드의 높이
onFocusEditText	입력창 포커싱 변화 시 현재 포커싱 된 입력 창 전달 (콜백) 주의) IKeypadManageHelper 를 사용하고 있다면 IKeypadManageHelper 의 onFocusEditText() 를 필히 호출	
Syntax	public void onFocusEditText(EditText edittext);	
Parameter 1	EditText	현재 포커싱 된 입력 창
inputViolationOccured	키 입력 값의 제한 규칙 위반 시 호출 (콜백)	
Syntax	public void inputViolationOccured (int errorCode);	
Parameter 1	int	제한 규칙 위반 코드 (위반한 제한 규칙 그대로 반환) setInputCheckPolicy() 메소드 내용 참조

6.1.2.4. 참고사항 및 유의사항

1. 암호화된 데이터 전송 시 유의사항

E2E 모드인 경우 "암호화 된 입력 데이터"와 더불어 "암호화 된 키" 가 최종 생성됩니다.
이 2 개의 데이터를 쌍으로 동시에 전송해야 합니다. (샘플 소스 코드 참조)

2. 난독화 (proguard) 적용 시 유의사항

안드로이드 SDK 에서 제공하는 난독화 도구인 proguard 적용 시 라이브러리에서 오류가 발생하는 경우 아래와 같은 조치를 취합니다. 프로젝트 폴더의 proguard-project.txt 파일에 다음 내용을 추가합니다.

```
-keep class com.nprotect.** {*,}
```

6.1.3. iOS 적용 가이드

6.1.3.1. 라이브러리 설치 및 설정

1. 라이브러리 및 리소스 파일 복사

1. 제공하는 라이브러리 파일을 해당 프로젝트에 추가합니다.
라이브러리 목록: keycryptm.a
2. 제공하는 zix_resources.bundle (이미지 번들) 파일을 해당 프로젝트에 추가합니다.
3. 제공하는 헤더 파일들을 해당 프로젝트에 추가합니다.
헤더 파일 목록: ixcSecureManager.h, UIKeyViewController.h 등

2. 프레임워크 추가

[Build Phases -> Link Binary With Libraries] 에서 아래의 프레임워크를 추가합니다.
"AudioToolbox.framework", "Security.framework", "libstdc++.dylib"

6.1.3.2. 코드 작성 예시

1. 라이선스 설정

부여 받은 "라이선스 키"와 "고객사 ID"를 설정합니다. 가능하면 애플리케이션에서 한번 호출하도록 합니다.

예) application: didFinishLaunchingWithOptions: 안에서 호출

```
[ixcSecureManager initLicense:license andCustomid:customer_id]; // 라이선스 및 고객사 ID 설정
```

2. 보안 키패드 설정

1. 보안 키패드를 적용할 UIViewController 의 Super 클래스를 "UIKeyViewController" 로 변경합니다.

```
#import " UIKeyViewController.h"

@interface ViewController : UIKeyViewController

...
```

2. XIB 혹은 Storyboard 에서 보안 키패드를 적용할 TextField 의 속성을 변경합니다.
"Secure Text Entry"를 체크 ON 하고 숫자전용 키패드를 사용할 TextField 인 경우는 "Keyboard Type"을 "Number Pad"로 지정합니다. QWERTY 키패드는 "Default" 그대로 나둡니다.
3. 보안 키패드를 적용할 TextField 각각을 아래처럼 지정해줍니다.

```
- (void)viewDidLoad {
    [super viewDidLoad];
    [self registerViews:self.view forced:NO];
    [self addSecureField:userPWtextfield]; // 보안 키패드 사용으로 설정
    [self addSecureField:userCBtextfield]; // 보안 캐패드 사용으로 설정
}
```

3. 입력 데이터 수집 및 전송

입력 데이터를 모바일 기기 내에서만 사용할 수도 있고 서버와 연동해서 사용할 수도 있습니다. 이에 따라 입력 데이터 처리하는 코드가 다릅니다.

1. 모바일 기기 내에서만 사용하는 경우 (Stand Alone 모드)

입력 완료 후 복호화 된 입력 데이터를 받을 시는 아래와 같이 `getRealText()` 메소드를 사용합니다.

```
NSString *result01 = [self getRealText:inputBox01];
```

2. 서버 연동해서 사용하는 경우 (E2E 모드)

- i. 미리 서버에서 키를 가져와서 `setServerKey()` 메소드로 설정합니다.
- ii. 보안 키패드 입력 완료 후 아래와 같이 `getServerData()` 메소드를 사용하여 서버 키로 암호화 된 입력 데이터와 암호화 된 키를 전달 받습니다.
- iii. 암호화 된 입력 데이터와 키를 정해진 메시지 형태로 구성하여 서버로 전달합니다.

주의) 서버로 전송할 때까지 서버에서 키를 가져올 때의 세션을 유지하고 있어야 합니다.

```
[self getServerData:inputBox01 :&cipherText Key:&serverKey];
```

6.1.3.3. API

1. `ixcSecureManager` 클래스

<code>ixcSecureManager</code>	SDK 관리 클래스
<code>initLicense</code>	라이선스 설정
Syntax	+ (BOOL)initLicense:(NSString *)license andCustomID:(NSString *)customID

Parameter 1	NSString *	라이선스
Parameter 2	NSString *	고객사 ID
return	BOOL	라이선스 유효
getVersion	라이브러리 버전 (정수형)	
Syntax	- (int)getVersion	
return	int	라이브러리 버전 (정수형)
getStringVersion	라이브러리 버전 (문자열)	
Syntax	- (NSString *)getStringVersion	
return	NSString *	라이브러리 버전 (문자열)
checkJailbreak	루팅 상태 체크	
Syntax	+ (BOOL)checkJailbreak	
return	BOOL	탈옥 여부

2. UIKeyViewController 클래스

UIKeyViewController	보안 키패드 UIViewController 클래스	
addSecureField	특정 TextField 를 보안 키패드 적용	
Syntax	- (void)addSecureField:(UITextField *)textField	
Parameter 1	UITextField *	적용할 TextField
setMinLength: Of:	입력문자 최소 개수 지정 (생략 가능)	
Syntax	- (void)setMinLength:(unsigned long)length Of:(UITextField *)textField;	
Parameter 1	UITextField *	적용할 TextField
Parameter 2	Integer	입력문자 최소 개수
setMaxLength: Of:	입력문자 최대 개수 지정 (생략 가능)	
Syntax	-(void)setMaxLength:(unsigned long)length Of:(UITextField *)textField;	
Parameter 1	UITextField *	적용할 TextField
Parameter 2	Integer	입력문자 최대 개수
setInputCheckPolicy: To:	입력 값 제한 정책 설정	
Syntax	- (void)setInputCheckPolicy:(unsigned long)policy To:(UITextField *)textField;	
Parameter 1	unsigned long	적용할 제한 정책 kCHECK_POLICY_ONLY_ALPHABET: 알파벳 글자만 입력 가능 kCHECK_POLICY_ONLY_ALPHANUMERIC:

		알파벳, 숫자만 입력 가능 (특수문자 입력 불가) kCHECK_POLICY_ONLY_PASSWORD_SYMBOL: 패스워드 가능 글자만 입력 가능 (원화, 달러, 엔 같은 특수문자 입력 불가) kCHECK_POLICY_DUPLICATED_2CHAR: 연속 2개 같은 글자 입력 불가 예) "aa", "CC" kCHECK_POLICY_DUPLICATED_3CHAR: 연속 3개 같은 글자 입력 불가 예) "aaa", "CCC" kCHECK_POLICY_CONTINUOUS_3CHARS: 3개 이상의 연속된 글자 입력 불가 예) "abc", "zyx", "123" kCHECK_POLICY_CONTINUOUS_4CHARS: 4개 이상의 연속된 글자 입력 불가 예) "1234", "abcd", "zyxw"
Parameter 2	UITextField *	적용할 TextField
setCryptoPlugin	암호화를 위한 암호화 플러그인 설정 (Standalone 전용)	
Syntax	- (void)setCryptoPlugin:(t_CRYPTO_PLUGIN)pluginId;	
Parameter 1	int	암호화 플러그인 ID
validateInputs	입력 값 내에 위반 문자열 존재 여부 확인	
Syntax	- (int)validateInputs:(NSArray *)patterns To:(UITextField *)textField FoundIndex:(uint *)pFoundIndex;	
Parameter 1	NSArray *	위반 문자열 배열
Parameter 2	UITextField *	대상 TextField
Parameter 3	uint *	위반 문자열 검출 시 검출된 문자열 인덱스
return	int	위반 문자열 검출 여부 0: 미발견, 이외: 발견
setAutoMoveMode	자동 TextField 이동 설정	
Syntax	- (void)setAutoMoveMode:(BOOL)mode offField:(UITextField *)textField nextField:(UITextField *)nextField;	
Parameter 1	BOOL	모드 ON / OFF
Parameter 2	UITextField *	적용할 TextField

Parameter 3	UITextField *	다음 이동 TextField
setAutoMoveMode	자동 TextField 이동 설정 (다음 TextField는 Tag 값에서 1을 더한 Tag를 갖고 있는 TextField)	
Syntax	- (void)setAutoMoveMode:(BOOL)mode ofField:(UITextField *)textField;	
Parameter 1	BOOL	모드 ON / OFF
Parameter 2	UITextField *	적용할 TextField
setAutoMoveModeOverAll	자동 TextField 이동 설정 (다음 TextField는 Tag 값에서 1을 더한 Tag를 갖고 있는 TextField)	
Syntax	- (void)setAutoMoveModeOverAll:(BOOL)mode;	
Parameter 1	BOOL	모드 ON / OFF
setAutoMoveWithChains	자동 TextField 이동 설정 (자동 이동할 TextField 를 배열로 지정)	
Syntax	- (void)setAutoMoveWithChains:(NSArray *)inputFields;	
Parameter 1	NSArray *	자동 이동할 TextField 배열
moveToPrevInputBox	지정된 이전 TextField 로 이동	
Syntax	- (UITextField *)moveToPrevInputBox;	
return	UITextField *	이동한 뒤의 TextField
moveToNextInputBox	지정된 다음 TextField 로 이동	
Syntax	- (UITextField *)moveToNextInputBox;	
return	UITextField *	이동한 뒤의 TextField
registerViews	TextField 를 찾아서 등록. (해당 View와 그 Child View 까지 확인)	
Syntax	- (void)registerViews:(UIView *)view forced:(BOOL)forced;	
Parameter 1	UIView *	적용할 View. (TextField 바로 지정 가능)
Parameter 2	BOOL	강제 설정 (일반적으로 NO 설정)
setServerKey	E2E 서버에서 발급받은 서버 키 설정 (서버 키 설정 시 E2E 모드로 동작)	
Syntax	- (void)setServerKey:(NSString *)key To:(UITextField *)textField;	
Parameter 1	NSString *	E2E 서버에서 발급받은 서버키
Parameter 2	UITextField *	해당 TextField
setShuffleType	Qwerty 키보드의 셔플 타입 설정	
Syntax	- (void)setShuffleType:(int)nType;	
Parameter 1	int	Qwerty 키패드의 셔플 타입
getRealText	입력 데이터 복호화하여 반환	

	(StandAlone 경우만 복호화되며 해당 TextField에는 가짜데이터 입력)	
Syntax	- (NSString *)getRealText:(UITextField *)textField;	
Parameter 1	UITextField *	해당 TextField
return	NSString *	입력 결과 값 리턴 (Stand Alone 모드일 경우만 반환)
getServerData	E2E 서버에서 발급받은 서버키 등록 사용하여 서버키 설정 시 E2E 모드로 설정	
Syntax	(BOOL)getServerData:(UITextField *)textField :(NSString **)cipherData Key:(NSString **)symKey;	
Parameter 1	UITextField *	해당 TextField
Parameter 2	NSString **	암호화 된 입력 데이터. 주의) 더블 포인터
Parameter 3	NSString **	암호화 된 키. 주의) 더블 포인터
return	BOOL	데이터 유효 여부
getInputResultOf	입력 데이터 반환	
Syntax	- (IxcInputCipherResult *)getInputResultOf:(UITextField *)textField;	
Parameter 1	UITextField *	해당 TextField
return	IxcInputCipherResult	입력 데이터 결과 (IxcInputCipherResult 클래스 참고)
compare	두 개의 TextField의 입력 값을 비교	
Syntax	- (NSComparisonResult)compare:(UITextField *)textField1 With:(UITextField *)textField2;	
Parameter 1	UITextField *	대상 TextField #1
Parameter 2	UITextField *	대상 TextField #2
return	NSComparisonResult	비교 결과 반환 NSOrderedSame: 같음, 그 외 : 다름
disableHideButton	키패드 숨김 버튼 비활성화	
Syntax	- (void)disableHideButton;	
setExtraToolbar	추가 커스텀 툴바 뷰 설정	
Syntax	- (void)setExtraToolbar:(UIView *)toolBar;	
Parameter 1	UIView *	추가 커스텀 툴바 뷰 객체
showExtraToolbar	추가 커스텀 툴바 보이기 / 숨기기	
Syntax	- (void)showExtraToolbar:(BOOL)isShow;	
Parameter 1	BOOL	추가 커스텀 툴바 뷰 보이기 or 숨기기
syncViewChangeHeight	보안 키패드 높이 변화 시 호출 (콜백)	
Syntax	- (void)syncViewChangeHeight:(CGFloat)padHeight;	

Parameter 1	CGFloat	보안 키패드 높이
inputViolationOccured	키 입력 값의 제한 규칙 위반 시 호출 (콜백)	
Syntax	- (void)inputViolationOccured:(unsigned long)errorCode;	
Parameter 1	unsigned long	제한 규칙 위반 코드 (위반한 제한 규칙 그대로 반환) setInputCheckPolicy() 메소드 내용 참조
textFieldWillMoving	textField 이동 시작 시 호출 (콜백)	
Syntax	- (void)textFieldWillMoving:(UITextField *)textField	
Parameter 1	UITextField *	대상 TextField
textFieldDidMoving	textField 이동 완료 시 호출 (콜백)	
Syntax	- (void)textFieldDidMoving:(UITextField *)textField	
Parameter 1	UITextField *	대상 TextField

3. IxcInputCipherResult 클래스

IxcInputCipherResult	입력 데이터 결과 클래스
key	입력 데이터 암호화 키 반환
Type	NSString *
encData	암호화 된 입력 데이터 반환
Type	NSString *
encDataLength	입력 데이터 길이 반환
Type	int
masterKey	마스터 키 반환 (Crypto Plugin 사용 시에만 사용)
Type	NSData *

6.1.3.4. 참고사항 및 유의사항

1. 주의 사항

- willAnimateRotationToInterfaceOrientation 사용 시
willAnimateRotationToInterfaceOrientation() 사용 시 함수 마지막에
[super willAnimateRotationToInterfaceOrientation] 을 반드시 호출합니다.
- textFieldDidBeginEditing 사용 시
textFieldDidBeginEditing 사용 시 함수 마지막에
[super textFieldDidBeginEditing] 을 반드시 호출합니다.
- viewWillAppear 사용 시

viewWillAppear 사용 시 함수 마지막에
[super viewWillAppear:animated] 을 반드시 호출합니다.

4. XIB 나 Storyboard 이외에 코드 상으로 TextField 추가하여 적용하는 경우
 - a) viewDidLoad 에서 TextField 들을 모두 addSubview 하고 나서 마지막에
[self registerViews:self.view forced:NO] 을 호출합니다.
 - b) TextField 하나씩 추가하려면 아래와 같이
[self registerViews:textfield forced:NO] 을 호출합니다.

6.1.4. 서버 적용 가이드⁵⁶

6.1.4.1. 라이브러리 설치

모든 jar 파일을 웹사이트 라이브러리 폴더(예, /WEB-INF/lib)에 복사합니다.
(WAS 시스템에 따라 라이브러리 경로가 다를 수 있으니 해당 라이브러리 경로에 복사합니다.)

6.1.4.2. 키 발급 페이지 적용방법⁵⁷

클라이언트에서 전달받은 uuid 값을 PluginFreeMobile.generate(session, uuid) 함수에 전달하여 암호화 키 값을 받습니다.

```
String uuid = request.getParameter("u");
String key = PluginFreeMobile.generate(session, uuid);
```

6.1.4.3. 데이터 복호화 페이지 적용방법

1. 하나의 페이지에 암호/복호화 키를 1 개 생성한 경우

PluginFreeMobile.decoder 함수를 통해 암호화 객체를 1 개 생성하고 재사용합니다.
PluginFreeMobile.keypad(decoder, input, value); 함수를 사용하여 값을 복호화합니다.

```
try {
    String uuid = n2b(request.getParameter("uid"));
    String key = n2b(request.getParameter("key"));
    String[] inputs = request.getParameterValues("i");
    if(inputs != null && inputs.length > 0) {
        PluginFreeDecrypt decoder = PluginFreeMobile.decoder(session, uuid, key);
        for(String inputname : inputs){
            String input = n2b(request.getParameter(inputname));
            String value = n2b(request.getParameter(inputname + "_e2e"));
            String plain = PluginFreeMobile.keypad(decoder, input, value);
            if(plain != null) {
                out.print(plain + "<br />");
            }
        }
    }
}
```

⁵⁶ Java, HttpServletRequest, HttpSession 기반 WAS 서버환경에서의 적용가이드입니다. 사전에 적용이 가능한지 / 고객맞춤형으로 변경제작이 가능한지 먼저 문의하십시오.

⁵⁷ 예제코드에서의 HttpServletRequest.getParameter()는 nProtect Online Security 모바일 API의 함수가 아니며 전문 통신 또는 다른 네트워크 통신을 통하여 데이터가 전달되고 있다는 전제하에 작성된 코드입니다. 즉, 전문통신 또는 네트워크통신은 nProtect Online Security 모바일에서 제공하는 기능이 아닙니다.

```

    }
} catch (PluginFreeException e){
    System.err.println(String.format("nProtect Plugin Free Service (Mobile, Keypad), %s : %s",
    e.getCode(), e.getMessage()));
}

```

2. 각각의 입력양식에 각각 암호/복호화 키를 생성한 경우

PluginFreeMobile.decoder 함수를 통해 암호화 객체를 각각 생성하여 처리합니다
 PluginFreeMobile.keypad(decoder, input, value); 함수를 사용하여 값을 복호화합니다.

```

try {
    String[] inputs = request.getParameterValues("i");
    if(inputs != null && inputs.length > 0) {
        for(String inputname : inputs){
            String input = n2b(request.getParameter(inputname));
            String uuid = n2b(request.getParameter(inputname + "_uid"));
            String key = n2b(request.getParameter(inputname + "_key"));
            String value = n2b(request.getParameter(inputname + "_e2e"));
            PluginFreeDecrypt decoder = PluginFreeMobile.decoder(session, uuid, key);
            String plain = PluginFreeMobile.keypad(decoder, input, value);
            if(plain != null) {
                out.print(plain + "<br />");
            }
        }
    }
} catch (PluginFreeException e){
    System.err.println(String.format("nProtect Plugin Free Service (Mobile, Keypad), %s : %s",
    e.getCode(), e.getMessage()));
}

```

6.1.4.4. 서버 API

1. PluginFreeMobile 클래스

클라이언트에서 전달받은 uuid값을 PluginFreeMobile.generate(session, uuid) 함수에 전달하여 암호화 키 값을 받습니다.

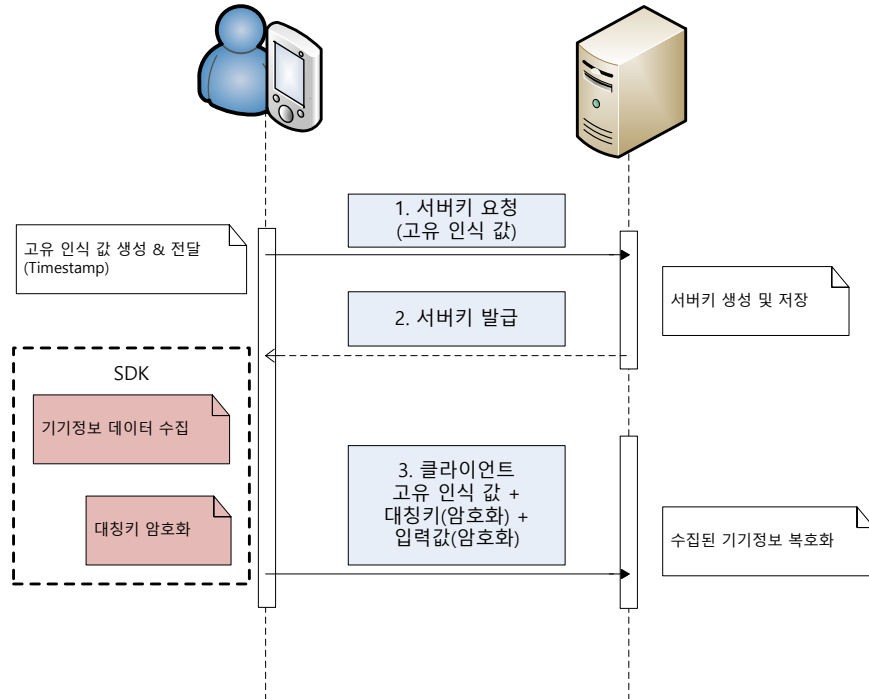
generate	[서버 암호화 랜덤키] 발급	
Syntax	public static String generate(HttpSession, String)	
Parameter 1	HttpSession	서버 세션 객체 인스턴스
Parameter 2	String	클라이언트에서 전달한 UUID(타입스탬프)

decoder	암복호화 처리 클래스 인스턴스	
Syntax	public static PluginFreeDecrypt decoder(HttpSession, String, String);	
Parameter 1	HttpSession	서버 세션 객체 인스턴스
Parameter 2	String	클라이언트에서 전달한 UUID(타입스탬프)
Parameter 3	String	클라이언트에서 생성한 [클라이언트 랜덤키] 값
keypad	데이터를 복호화합니다.	
Syntax	public void keypad(PluginFreeDecrypt, String, String);	
Parameter 1	PluginFreeDecrypt	PluginFreeDecrypt 객체 인스턴스
Parameter 2	String	입력양식의 데이터 값
Parameter 3	String	암호화된 데이터 값

6.2. 모바일 기기정보 수집 적용가이드

6.2.1. 모바일 기기정보 수집 방법

6.2.1.1. 기기정보 수집 방식 도식도



1. WAS 기동 시에 RSA 키 쌍을 생성합니다.
2. 클라이언트의 요청 시 임의의 서버 랜덤 값을 생성하고 이 키를 RSA 공개키와 함께 암호화하여 클라이언트에 전송합니다.
3. 클라이언트가 받은 서버 키를 보안 키패드 SDK 에 전달하면 SDK 내부에서 파싱 후 서버에서 생성한 서버 랜덤 값과 SDK 내부에서 생성된 클라이언트 랜덤 값을 혼합하여 [데이터 암호화용 키]를 생성합니다.
4. 생성된 클라이언트 랜덤 값은 RSA 공개키로 암호화합니다.
5. 수집된 기기 데이터들을 암호화 하여 저장합니다.
6. 사용자의 입력 완료 후 [암호화된 수집 기기정보] 와 [암호화된 클라이언트 랜덤 키] 를 서버로 전달합니다.
7. 서버는 공개키로 암호화된 클라이언트 랜덤 키를 RSA 개인키로 복호화하고 이 것을 서버 랜덤 키를 혼합하여 암호키를 구성하고 이로 수집한 사용자 기기정보를 복호화합니다.

6.2.2. Android 적용 가이드

6.2.2.1. 라이브러리 설치 및 설정

1 라이브러리 파일 복사

제공하는 라이브러리 파일 모두를 해당 프로젝트의 libs 폴더에 복사해 넣습니다.

라이브러리 목록: libSLM.jar, libE2E.jar, npcrypto.jar, libnpcf_ma_u.so

2 AndroidManifest 설정

AndroidManifest.xml 파일에 아래의 권한을 추가합니다.

```
<uses-permission android:name="android.permission.INTERNET">
<uses-permission android:name="android.permission.GET_TASKS">
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE">
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE">
<uses-permission android:name="android.permission.READ_PHONE_STATE">
```

6.2.2.2. 코드 작성 예시

1. 라이선스를 설정

부여 받은 라이선스 키와 고객사 ID 를 설정합니다. 애플리케이션에서 한번 호출
예) Application 클래스나 메인 Activity onCreate() 안에서 호출

```
ixSecureManager.initLicense(context, license, customer_id); // 라이선스 및 고객사 ID 설정
```

2. 기기정보 수집 클래스 초기화

```
ixLogMessage ixLogMessage = new ixLogMessage(context); // 인스턴스 생성
ixLogMessage.setSeverKey(serverkey); // 서버에서 가져온 서버키 설정
ixLogMessage.setCheckApp(pkgName); // 체크할 앱의 패키지명 설정 (생략 가능)
```

3. 암호화된 수집 정보 획득

```
String encryptedLogData = ixLogMessage.getEveryLog(); // 암호화된 수집 정보
```

4. 암호화된 클라이언트 키 획득

```
String encryptedKey = ixLogMessage.getBuiltKey(); // 암호화된 클라이언트 키
```

6.2.2.3. API

1. ixSecureManager 클래스

lxSecureManager	SDK 관리 클래스	
package	com.nprotect	
initLicense	라이선스 설정	
Syntax	public static boolean initLicense(String license, String customerId)	
Parameter 1	String	라이선스
Parameter 2	String	고객사 ID
return	boolean	라이선스 유효
getVersion	라이브러리 버전 (정수형)	
Syntax	public static int getVersion()	
return	String	라이브러리 버전 (정수형)
getVersionString	라이브러리 버전 (문자열)	
Syntax	public static String getVersionString()	
return	String	SDK 버전 문자열
checkRooting	루팅 상태 체크	
Syntax	public static boolean checkRooting()	
return	boolean	루팅 여부

2. lxLogMessage 클래스

lxLogMessage	기기정보 수집 클래스	
package	com.nprotect.seculog	
lxLogMessage	기기정보 수집 클래스의 생성자	
Syntax	public lxLogMessage(Context ctx)	
Parameter 1	Context	Activity 또는 Application의 Context 객체
setSeverKey	서버키 설정	
Syntax	public boolean setSeverKey(String serverKey)	
Parameter 1	String	서버키
return	boolean	서버키 유효
setCheckApp	설치&동작 여부 확인할 앱의 패키지명 설정	
Syntax	public void setCheckApp(String pkgName)	
Parameter 1	String	확인할 앱의 패키지명
getEveryLog	암호화된 수집 정보 전달	
Syntax	public String getEveryLog()	
return	String	암호화된 수집 정보
getBuiltKey	암호화된 클라이언트 키 전달	

Syntax	public String getBuiltKey()	
return	String	암호화된 클라이언트 키

3. IxLogManager 클래스

IxLogManager	기기정보 수집 클래스	
Package	com.nprotect.seculog	
IxLogManager	기기정보 수집 클래스의 생성자	
Syntax	public IxLogManager(Context ctx)	
Parameter 1	Context	Activity 또는 Application의 Context 객체
getEveryLogByString	암호화되지 않은 수집 정보 전달	
Syntax	public String getEveryLogByString()	
return	String	구분자로 분리된 수집 정보 문자열 예) KEY1=DATA1&&KEY2=DATA2 ...
getEveryLogByMap	암호화되지 않은 수집 정보 전달	
Syntax	public HashMap<String, String> getEveryLogByString()	
return	HashMap	수집 정보 맵 객체

6.2.2.4. 참고사항 및 유의사항

1. 난독화(Proguard) 적용 시 유의사항

안드로이드 SDK 에서 제공하는 난독화 도구인 proguard 적용 시 라이브러리에서 오류가 발생하는 경우 아래와 같은 조치를 취합니다. 프로젝트 폴더의 proguard-project.txt 파일에 다음 내용을 추가합니다.

```
-keep class com.nprotect.** {*,}
```

6.2.3. iOS 적용 가이드

6.2.3.1. 라이브러리 설치 및 설정

1. 라이브러리 파일 복사

제공하는 라이브러리 파일을 해당 프로젝트에 추가합니다.

라이브러리: libSecuLogMaster.a

헤더 파일: ixcSecureManager.h, ixcSecuLogMaster.h

2. 필수 프레임워크 추가

[Build Phases -> Link Binary With Libraries] 에서 아래의 프레임워크를 추가합니다.

"CoreTelephony.framework", "SystemConfiguration.framework", libstdc++.dylib

6.2.3.2. 코드 작성 예시

1. SDK 라이선스를 설정합니다.

부여 받은 라이선스 키와 고객사 ID 를 설정합니다. 애플리케이션에서 한번 호출
예) application: didFinishLaunchingWithOptions: 안에서 호출

```
[ixcSecureManager initWithLicense:license andCustomid:customer_id]; // 라이선스 및 고객사 ID 설정
```

2. 기기정보 수집 클래스를 초기화 합니다.

```
ixcSecuLogMaster *logMaster = [[ixcSecuLogMaster alloc] init]; // 인스턴스 생성
[logMaster.setSeverKey:serverkey]; // 서버에서 가져온 서버키 설정
[logMaster.setCheckApp:SchemeURL]; // 체크할 앱의 Scheme URL 설정 (생략 가능)
```

3. 암호화된 수집 정보를 가져옵니다.

```
NSString *encryptedLogData = [logMaster getEveryLog]; // 암호화된 수집 정보
```

4. 암호화된 클라이언트 키를 가져옵니다.

```
NSString *encryptedKey = [logMaster getBuiltKey]; // 암호화된 클라이언트 키
```

6.2.3.3. API

1. ixcSecureManager 클래스

ixcSecureManager	SDK 관리 클래스
initWithLicense	라이선스 설정
Syntax	+ (BOOL)initWithLicense:(NSString *)license

	andCustomID:(NSString *)customID	
Parameter 1	NSString *	라이선스
Parameter 2	NSString *	고객사 ID
return	BOOL	라이선스 유효
getVersion	라이브러리 버전 (정수형)	
Syntax	- (int)getVersion	
return	int	라이브러리 버전 (정수형)
getVersionString	라이브러리 버전 (문자열)	
Syntax	- (NSString *)getVersionString	
return	NSString *	라이브러리 버전 (문자열)
checkJailbreak	루팅 상태 체크	
Syntax	+ (BOOL)checkJailbreak	
return	BOOL	탈옥 여부

2. ixcSecuLogMaster 클래스

ixcSecuLogMaster	기기정보 수집 클래스	
setSeverKey	서버키 설정	
Syntax	- (BOOL)setServerKey:(NSString *)key	
Parameter 1	NSString *	서버키
return	BOOL	서버키 유효
setCheckApp	설치&동작 여부 확인할 앱의 패키지명 설정	
Syntax	- (void)setCheckApp:(NSString *)appPackageName	
Parameter 1	NSString *	확인할 앱의 Schme URL
getEveryLog	암호화된 수집 정보 전달	
Syntax	- (NSString *)getEveryLog	
return	NSString *	암호화된 수집 정보 문자열
getBuiltKey	암호화된 클라이언트 키 전달	
Syntax	- (NSString *)getBuiledKey	
return	NSString *	암호화된 클라이언트 키
getEveryLogByString	암호화되지 않은 수집 정보 전달	
Syntax	- (NSString *)getEveryLogByString	
return	NSString *	구분자로 분리된 수집 정보 문자열 예) KEY1=DATA1&&KEY2=DATA2 ...
getEveryLogByDictionary	암호화되지 않은 수집 정보 전달	

Syntax	- (NSDictionary *)getEveryLogByDictionary	
return	NSDictionary *	수집 정보 Dictionary 객체

6.2.4. 서버 적용 가이드⁵⁸

6.2.4.1. 라이브러리 설치

모든 jar 파일을 웹사이트 라이브러리 폴더(예, /WEB-INF/lib)에 복사합니다.
(WAS 시스템에 따라 라이브러리 경로가 다를 수 있으니 해당 라이브러리 경로에 복사합니다.)

6.2.4.2. 키 발급 페이지 적용방법⁵⁹

클라이언트에서 전달받은 uuid 값을 PluginFreeMobile.generate(session, uuid) 함수에 전달하여 암호화 키 값을 받습니다.

```
String uuid = request.getParameter("u");
String key = PluginFreeMobile.generate(session, uuid);
```

6.2.4.3. 모바일 단말수집정보 복호화 페이지 적용방법

PluginFreeMobile.decoder 함수를 통해 암호복호화 객체를 1 개 생성하고 재사용합니다.
PluginFreeMobile.mobile(decoder, value); 함수를 사용하여 값을 복호화합니다.

자세한 수집 내용은 [별첨 1.3 모바일단말정보] 수집정보를 참고하십시오.

```
try {
    String uuid = n2b(request.getParameter("u"));
    String key = n2b(request.getParameter("k"));
    String value = n2b(request.getParameter("d"));

    PluginFreeDecrypt decoder = PluginFreeMobile.decoder(session, uuid, key);

    PluginFreeMobileDTO dto = PluginFreeMobile.mobile(decoder, value);
}
```

⁵⁸ Java, HttpServletRequest, HttpSession 기반 WAS 서버환경에서의 적용가이드입니다. 사전에 적용이 가능한지 / 고객맞춤형으로 변경제작이 가능한지 먼저 문의하십시오.

⁵⁹ 예제코드에서의 HttpServletRequest.getParameter()는 nProtect Online Security 모바일 API의 함수가 아니며 전문 통신 또는 다른 네트워크 통신을 통하여 데이터가 전달되고 있다는 전제하에 작성된 코드입니다. 즉, 전문통신 또는 네트워크통신은 nProtect Online Security 모바일에서 제공하는 기능이 아닙니다.

```

        if(dto != null){
            out.print(dto.toString().replaceAll("\n", "<br />\n"));
        } else {
            System.out.println("nProtect Plugin Free Service (Mobile, FDS), Empty.");
        }
    } catch (PluginFreeException e){
        System.err.println(String.format("nProtect Plugin Free Service (Mobile, FDS), %s : %s",
            e.getCode(), e.getMessage()));
    }
}

```

6.2.4.4. 서버 API

1. PluginFreeMobile 클래스

generate	[서버 암호화 랜덤키] 발급	
Syntax	public static String generate(HttpSession, String)	
Parameter 1	HttpSession	서버 세션 객체 인스턴스
Parameter 2	String	클라이언트에서 전달한 UUID(타입스탬프)

decoder	암복호화 처리 클래스 인스턴스	
Syntax	public static PluginFreeDecrypt decoder(HttpSession, String, String);	
Parameter 1	HttpSession	서버 세션 객체 인스턴스
Parameter 2	String	클라이언트에서 전달한 UUID(타입스탬프)
Parameter 3	String	클라이언트에서 생성한 [클라이언트 랜덤키] 값

keypad	데이터를 복호화합니다.	
Syntax	public PluginFreeMobileDTO mobile(PluginFreeDecrypt, String);	
Parameter 1	PluginFreeDecrypt	PluginFreeDecrypt 객체 인스턴스
Parameter 2	String	암호화된 데이터 값

7. 키보드보안 Flash SDK 적용 가이드

7.1. Flash SDK⁶⁰ 파일 구성 및 API

7.1.1. Flash SDK 파일의 구성

파일명	설명
com.nprotect.pluginfree.flash.sdk.swc	- Flash 용 KeyCrypt SDK Library 파일
com.nprotect.pluginfree.flash.sdk.examples.zip	- Flash Builder용 기본 샘플 프로젝트 코드
com.nprotect.pluginfree.zip	- WAS 및 웹 서버 모듈

[표 6-1] Flash SDK 모듈

7.1.2. PluginFreeFlashSDK 클래스

PluginFreeFlashSDK	Flash SDK 관리 클래스	
PluginFreeFlashSDK	Flash SDK 설정	
Syntax	+ new com.nprotect.pluginfree.flash.PluginFreeFlashSDK(Array)	
Parameter 1	Array	보호하려는 TextInput 필드 배열
return	instance	Flash SDK 인스턴스
registField	동적으로 추가되는 TextInput에 대한 추가등록 함수	
Syntax	+ new com.nprotect.pluginfree.flash.PluginFreeFlashSDK(Array)	
Parameter 1	TextInput	보호하려는 TextInput 필드
return	Boolean	등록 성공 여부(true : 성공, false : 실패)

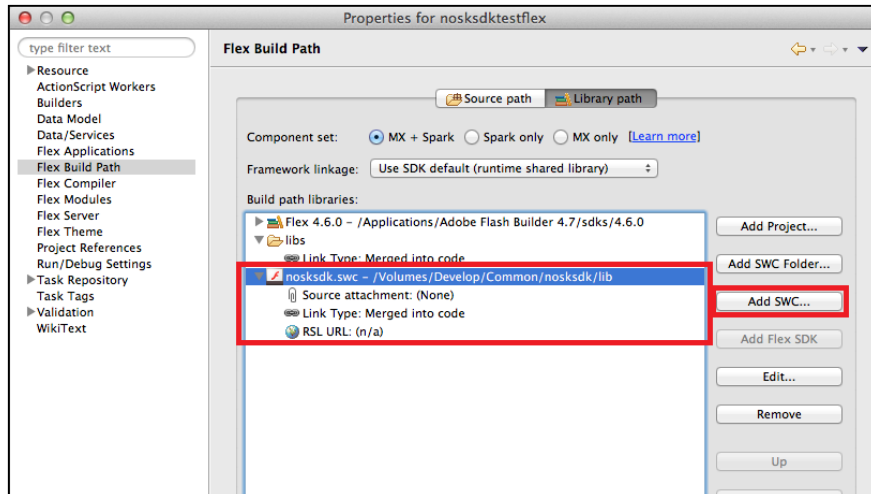
[표 6-2] PluginFreeFlashSDK 클래스

7.2. Flash SDK 프로젝트 설정 및 적용방법

7.2.1. Flash SDK 사용을 위한 ActionScript 추가 및 수정

- 1) 프로젝트 속성 > Flex Builder Path에서 com.nprotect.pluginfree.flash.sdk.swc 라이브러리를 추가합니다. (프로젝트의 libs 경로에 Drag&Drop하여 라이브러리의 등록이 가능합니다.)

⁶⁰ 2015년 9월 기준, nProtect Online Security의 FlashSDK는 E2E를 지원하지 않습니다.



2) `ActionScript(some.action.script.as)` 코드에 Flash SDK 사용을 위하여 객체 import 및 변수를 선언 합니다.

```
import com.nprotect.pluginfree.flash.PluginFreeFlashSDK;
```

```
private var nossdk : PluginFreeFlashSDK;
```

3) 변수를 선언 후 생성을 합니다.

- `initFlashSDK` 함수는 MXML 파일에서 호출 하기 위한 함수

```
public function initFlashSDK(txtInputs:Array):void
{
    if( null == nossdk )
        nossdk = new PluginFreeFlashSDK(txtInputs);
    else {
        for(var i:uint = 0 ; i < txtInputs.length; i++) {
            nossdk.registField(txtInputs[i]);
        }
    }
}
```

4) 예제 코드 [UserActionScript.as]

- Flash SDK 관련 코드는 다음과 같습니다.

```
import spark.components.TextInput;
import com.nprotect.pluginfree.flash.PluginFreeFlashSDK;

private var nossdk : PluginFreeFlashSDK;

public function initFlashSDK(txtInputs:Array):void
{
```

```

        nossdk = new PluginFreeFlashSDK(txtInputs);
    }

    public function registField(txtInput:TextInput):void
    {
        var result:Boolean = nossdk.registField(txtInput);
    }

```

7.2.2. FlashSDK 사용을 위한 MXML 수정

- 1) 화면 MXML파일을 다음과 같이 수정합니다. Flash 로딩 완료 후 ActionScript 파일에서 등록된 FlashSDK 함수를 호출 하기 위해 creationComplete 함수를 지정합니다.

기존 내용

```

<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
               xmlns:s="library://ns.adobe.com/flex/spark"
               xmlns:mx="library://ns.adobe.com/flex/mx">

```

변경 내용

```

<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
               xmlns:s="library://ns.adobe.com/flex/spark"
               xmlns:mx="library://ns.adobe.com/flex/mx"
               creationComplete="init()">

```

- 2) 수정한 Action Script 파일을 등록합니다.

```

<fx:Script source="UserActionScript.as"></fx:Script>

```

- 3) KeyCrypt FlashSDK 함수 호출을 위한 함수 추가합니다.

- Init 함수는 **creationComplete** 에서 호출 하는 함수
- 키보드 입력을 보호하고자 하는 TextInput 객체를 배열형태로 인자값으로 지정합니다.

```

<fx:Script>
    <![CDATA[
        public function init():void
        {
            initFlashSDK(new Array(testInput1, testInput2));
            registField(testInput3);
        }
    ]]>

```

```
</fx:Script>
```

4) Password TextInput Field 일 경우 반드시 `displayAsPassword="true"` 속성을 등록합니다.

```
<s:TextInput x="0" y="0" id="testInput1" enabled="true" text="" />
<s:TextInput x="0" y="0" id="testInput2" enabled="true"
    displayAsPassword="true" text="" />
```

5) 예제

- Flash SDK 관련 코드는 다음과 같습니다.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark"
    xmlns:mx="library://ns.adobe.com/flex/mx"
    creationComplete="init()">

    <fx:Script source="UserActionScript.as"></fx:Script>
    <fx:Script>
        <![CDATA[
            public function init():void
            {
                initFlashSDK(new Array(testInput1, testInput2));
            }
        ]]>
    </fx:Script>
    <s:VGroup>
        <s:HGroup>
            <s:TextInput x="0" y="0" id="testInput1" enabled="true" />
        </s:HGroup>

        <s:HGroup>
            <s:TextInput x="0" y="0" id="testInput2" enabled="true"
                displayAsPassword="true" />
        </s:HGroup>

        <s:HGroup>
            <s:TextInput x="0" y="0" id="testInput3" enabled="true"
                displayAsPassword="true" />
        </s:HGroup>
```

```
</s:VGroup>
</s:Application>
```

7.2.3. FlashSDK Javascript 적용

- 1) HTML 페이지의 <head>부분에 아래와 같이 코드를 삽입합니다.

```
<script type="text/javascript" src="/pluginfree/js/jquery-1.11.0.min.js"></script>
<script type="text/javascript" src="/pluginfree/jsp/nppfs.script.jsp"></script>
<script type="text/javascript" src="/pluginfree/js/nppfs-1.x.x.js"></script>
```

- 2) 페이지 onLoad시에 npPfsStartup함수가 실행되도록 합니다. 이때 지정하는 기능선택은 페이지에서 다른 기능을 사용하지 않는다면 모두 false로 지정합니다.

```
<script type="text/javascript">
jQuery(document).ready(function(){
    npPfsStartup(document.form1, false, false, false, false, "npkencrypt", "on");
});
</script>
```

- 3) Flash 객체의 속성에서 Flash와 Javascript가 상호 호출이 가능하도록 allowScriptAccess 옵션을 always로 하여 권한을 추가합니다.

```
<object
    type="application/x-shockwave-flash"
    id="UserActionScript"
    name="UserActionScript"
    data="/flash/UserActionScript.swf"
    width="600px" height="120px">
    <param name="menu" value="false">
    <param name="scale" value="noScale">
    <param name="allowFullscreen" value="true">
    <param name="allowScriptAccess" value="always">
    <param name="bgcolor" value="">
    <param name="wmode" value="direct">
</object>
```

- 4) 예제. /pluginfree/examples/flash.jsp 페이지 참고

```
<%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8"
%><!DOCTYPE html>
<html>
```

```

<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<title>nProtect Online Security v1.0.0</title>
<script type="text/javascript" src="/pluginfree/js/jquery-1.11.0.min.js"></script>
<script type="text/javascript" src="/pluginfree/jsp/nppfs.script.jsp"></script>
<script type="text/javascript" src="/pluginfree/js/nppfs-1.5.0.js"></script>
<script type="text/javascript">
jQuery(document).ready(function(){
    npPfsStartup(document.form1, false, false, false, false, "npkencrypt", "on");
});
</script>
</head>
<body>
<div style="margin-bottom:20px; padding:10px; border:1px solid #000;">
    <object
        type="application/x-shockwave-flash"
        id="UserActionScript"
        name="UserActionScript"
        data="/flash/UserActionScript.swf"
        width="400px" height="120px">
        <param name="menu" value="false">
        <param name="scale" value="noScale">
        <param name="allowFullscreen" value="true">
        <param name="allowScriptAccess" value="always">
        <param name="bgcolor" value="">
        <param name="wmode" value="direct">
    </object>
</div>
</body>
</html>

```

8. 제품 설치 및 삭제

8.1. 설치하기

nProtect Online Security V1.0 이 설치되지 않은 상태에서 서비스 페이지에 접속하면 설치 안내 페이지로 이동하게 됩니다. 설치 안내 페이지에서 설치 파일을 다운로드 합니다.

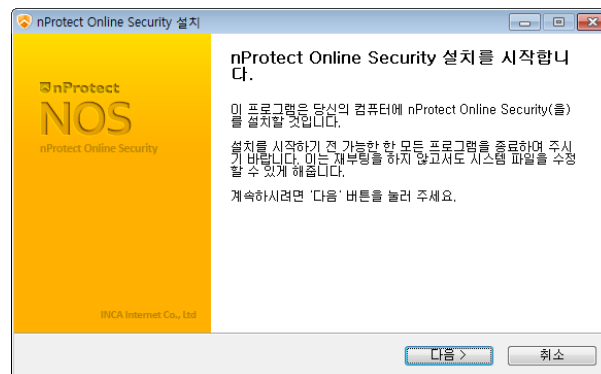
8.1.1. Windows

1. 설치 사이트에서 다운로드 하면 아래 [그림 8-1]과 같은 메시지가 나타납니다. nProtect Online Security 를 설치 하기 위해 "실행(R)"을 클릭합니다.



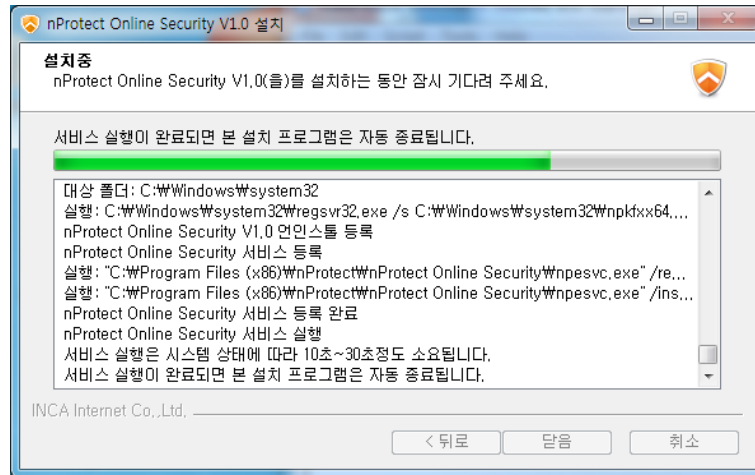
[그림 8-1] 설치 파일 다운로드

2. 설치 파일 다운로드가 완료되면 아래 [그림 8-2]와 같이 설치 화면이 나타납니다. "다음"을 클릭하여 nProtect Online Security 를 설치해 주십시오.



[그림 8-2] 설치 프로그램 실행 화면

3. 그러면 [그림 8-3]과 같이 설치를 진행합니다.



[그림 8-3] 설치 진행 화면

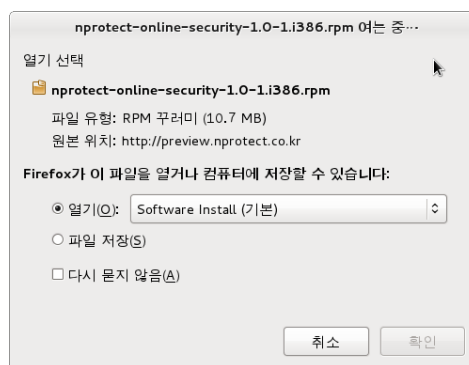
4. 설치 및 업데이트가 완료되어 실행되면 모니터 오른쪽 하단에 [그림 8-4]와 같이 nProtect Online Security의 트레이 아이콘이 표시됩니다.



[그림 5-4] 설치 후 표시되는 트레이 아이콘 이미지

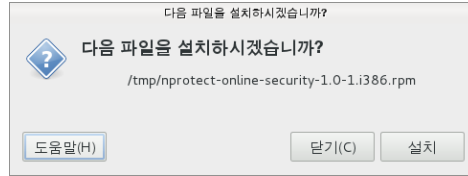
8.1.2. Linux

1. 설치 사이트에서 다운로드 하면 아래 [그림 8-4]과 같은 메시지가 나타납니다. nProtect Online Security를 설치 하기 위해 "열기(O)"을 클릭합니다.



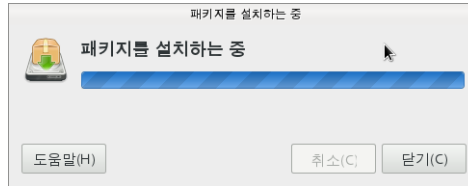
[그림 8-4] 설치 파일 다운로드

2. 설치 파일 다운로드가 완료되면 아래 [그림 8-5]와 같이 설치 화면이 나타납니다. "설치"을 클릭하여 nProtect Online Security를 설치해 주십시오.



[그림 8-5] 설치 동의

3. 그러면 [그림 8-6]과 같이 설치를 진행합니다.



[그림 8-6] 설치 진행 화면

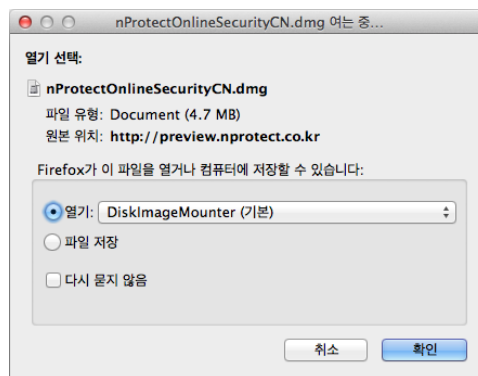
4. 설치 및 업데이트가 완료 후 nProtect Online Security V1.0 서비스 페이지에 접속하면 모니터 오른쪽 하단에 [그림 8-7]와 같이 nProtect Online Security 의 트레이 아이콘이 표시됩니다.



[그림 8-7] 프로그램 실행 화면

8.1.3. Macintosh

1. 설치 사이트에서 다운로드 하면 아래 [그림 8-8]과 같은 메시지가 나타납니다. nProtect Online Security 를 설치 하기 위해 "열기"을 클릭합니다.



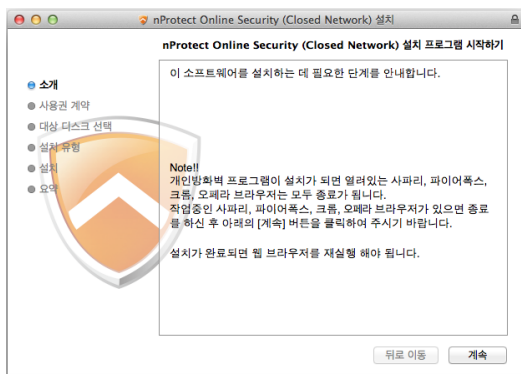
[그림 8-8] 설치 파일 다운로드

- 설치 파일 다운로드가 완료되면 아래 [그림 8-9]와 같이 패키지 파일이 화면에 나타납니다. 패키지 파일을 더블 클릭하여 nProtect Online Security 를 설치해 주십시오.

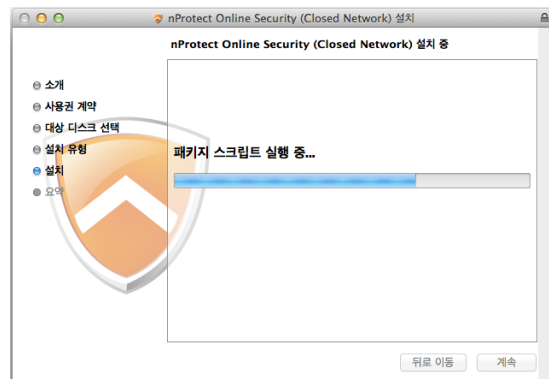


[그림 8-9] 설치 프로그램

- 아래 [그림 8-10]와 같이 설치 화면이 나타나면 “계속”을 클릭하여 nProtect Online Security 를 설치해 주십시오. [그림 8-10] 설치 프로그램 실행 화면 [그림 8-11]과 같이 설치가 진행됩니다.

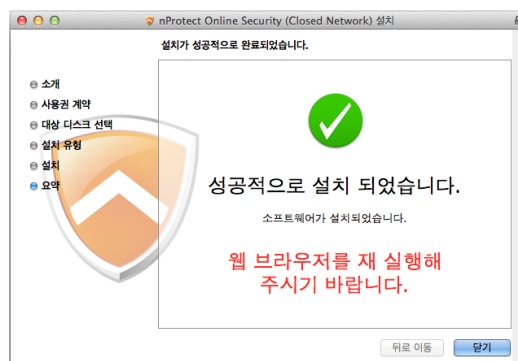


[그림 8-10] 설치 프로그램 실행 화면



[그림 8-11] 설치 진행 화면

- 설치가 완료 되면 [그림 8-12]과 같이 완료 화면이 출력됩니다.



[그림 8-12] 설치 완료

8.2. 삭제하기

nProtect Online Security V1.0 이 설치된 상태에서 제품을 삭제하기 위한 순서입니다.

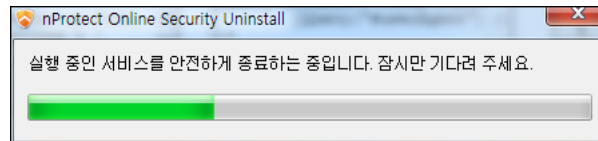
8.2.1. Windows

1. [그림 8-13]와 같이 제어판의 “프로그램 및 기능”에서 nProtect Online Security V1.0(PFS)를 찾아 “제거”를 선택합니다.



[그림 8-13] 프로그램 제거 선택 화면

2. nProtect Online Security V1.0(PFS) 삭제를 실행하면 다음과 같이 삭제를 진행하는 화면이 표시됩니다.

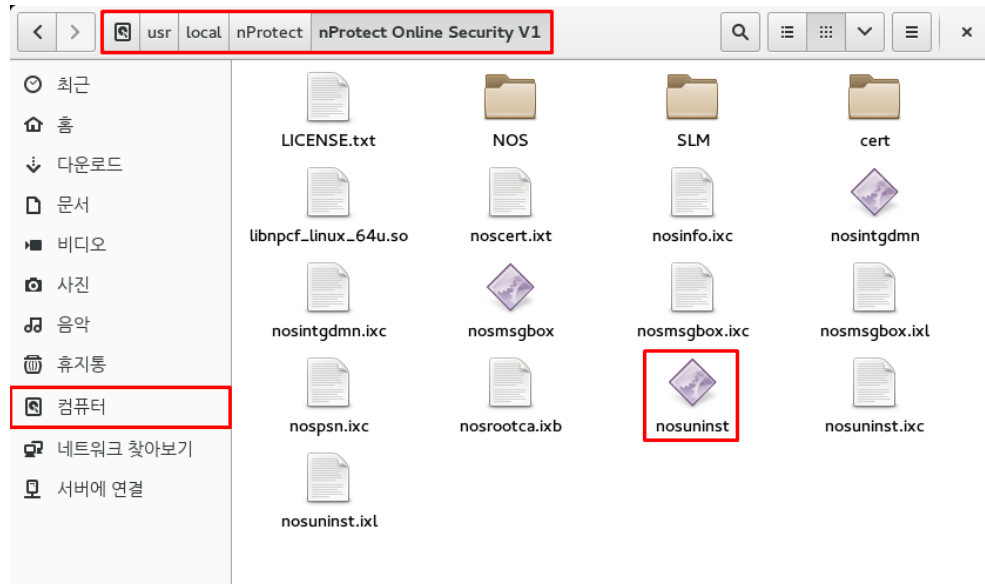


[그림 8-14] 프로그램 삭제 진행 화면

3. 삭제가 완료되면 프로그램 삭제 진행 화면이 자동으로 종료됩니다.

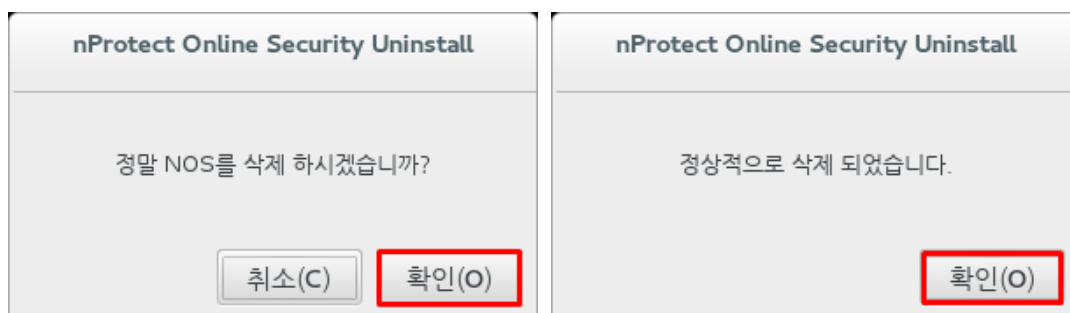
8.2.2. Linux

- 1 아래 [그림 8-15]와 같이 usr > local > nProtect > nProtect Online Security V1 > nosuninst 파일을 실행합니다.



[그림 8-15] Uninstall 프로그램 선택

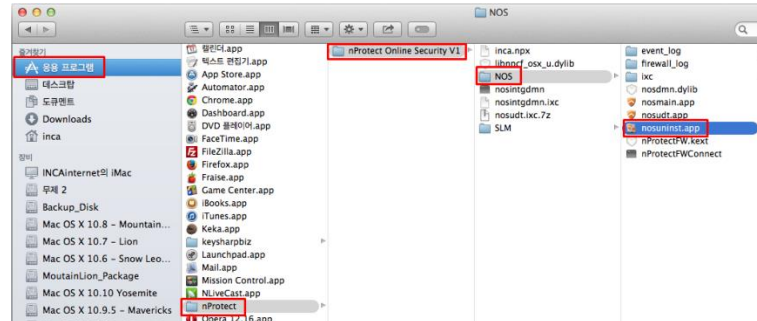
- 2 삭제를 진행하기 위해 사용자계정 암호를 입력하고 “확인” 버튼을 클릭하여 삭제를 진행합니다.
- 3 [그림 8-16] 과 같은 삭제 알림 창에서 “확인”을 누르면 삭제가 진행되고, 삭제완료 시 알림 창이 출력과 함께 삭제가 완료됩니다.



[그림 8-16] 삭제 알림 창

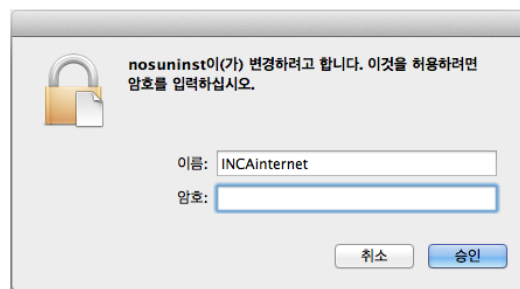
8.2.3. Macintosh

1. 아래 그림 [그림 8-17]과 같이 Finder > 응용프로그램 > nProtect > nProtect Online Security V1 > NOS > nosuninst.app 파일을 실행 합니다.



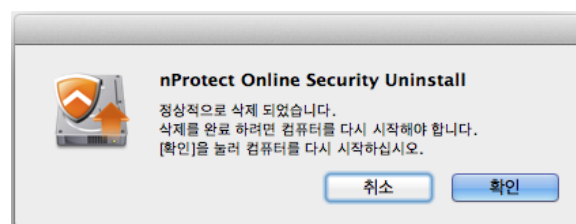
[그림 8-17] Uninstall 프로그램 선택

2. 삭제를 진행하기 위해 [그림 8-18] 창에서 사용자계정 암호를 입력하고 “승인” 버튼을 클릭하여 삭제를 진행합니다.



[그림 8-18] 사용자계정 암호 입력 창

3. [그림 8-19]과 같은 PC 재부팅 알림 창에서 “확인”을 누르면 PC 재 부팅과 동시에 정상적으로 삭제 완료됩니다



[그림 8-19] 재 부팅 알림 창

9. FAQ

9.1. 설치 및 삭제

1. nProtect Online Security V1.0(이하 NOS)을 어디에서 다운로드 받을 수 있나요?

- NOS 는 제휴된 웹 사이트(금융기관 또는 공공기관 등) 이용 시 안내에 따라 설치페이지로 이동한 후 다운로드하여 설치가 가능합니다.

2. NOS Setup 파일이 다운로드되지 않는 경우 어떻게 해야 하나요?

- NOS Setup 파일의 다운로드 서버로의 접근을 차단하는 프로그램이나 네트워크 환경, 또는 파일 다운로드를 관리하는 프로그램에서 다운로드를 차단하는 경우가 있습니다.
- PC 내에 설치된 프로그램 및 네트워크 환경(방화벽, 프록시 설정, DNS 설정 등)을 확인하여 다운로드를 차단하지 않도록 설정해 주시기 바랍니다.

3. 설치 시 PC의 관리자 권한이 필요한가요?

- Windows Vista 이상의 OS 에서 NOS 프로그램을 설치하기 위해서는 관리자 권한이 필요합니다.
- 설치 후에는 일반 유저 권한으로도 이용이 가능합니다.

4. 설치 안내 메시지가 지속적으로 출력되는 경우

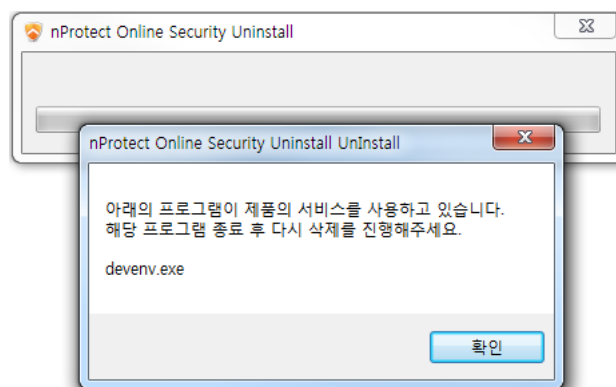
- 설치 안내 메시지는 NOS 가 설치되지 않았거나 정상적으로 실행되지 않은 경우에 표시되는 메시지입니다.
- [3. 운영 환경]을 참고하여 사용하고 계신 OS 와 브라우저가 지원 범위에 포함되는지 확인하여 주십시오. 지원 범위에 포함된 환경에서 해당 페이지에 접속해 주시기 바랍니다.
- NOS 를 설치하지 않았다면 [6.1.1]과 [5.1. 설치하기]를 참고하여 NOS 를 설치한 후 해당 페이지에 다시 접속해 주시기 바랍니다.

5. NOS 설치/실행이 정상적으로 되지 않습니다.

- NOS 의 설치/실행이 정상적으로 진행되지 않는 경우에는 PC 환경 및 네트워크의 환경적인 부분을 확인할 필요가 있습니다.

- [3. 운영 환경]을 참고하여 사용하고 계신 OS 와 브라우저가 지원 범위에 포함되는지 확인하여 주십시오. 지원 범위에 포함된 환경에서 해당 페이지에 접속해 주시기 바랍니다.
- NOS 설치 시 문제가 발생할 경우 현재 구동 중인 상태와 관련된 부분일 수 있으니 우선 PC 를 재부팅하여 재시도 바랍니다.
- 재부팅 후 재시도 시에도 동일한 증상이 발생한다면 [5.2. 삭제하기]를 참고하여 NOS 를 삭제하신 후 해당 페이지에 접속하여 NOS 를 재설치 해주시기 바랍니다. ([5.1. 설치하기] 참고)
- 또한 PC 에 설치된 보안프로그램 등에서 NOS 의 설치/실행을 차단하는지 확인하시기 바랍니다. 사내에서 사용하는 PC 인 경우 사내 보안을 강화하기 위해 설치된 다양한 보안 프로그램이 있다면 해당 보안 프로그램에 대해 다음과 같은 확인이 필요합니다.
 - 사내 다른 PC 에서도 동일한 증상이 발생하는지 확인
 - 해당 보안프로그램 off 혹은 삭제 후 다시 확인
 - 사내 보안팀(전산팀)에 NOS 제품에 대한 자체보호, 방화벽, 접근/실행 차단 설정되어 있는지 확인하여 해당 기능 해제 후 다시 확인

6. 삭제 시 “~ 해당 프로그램 종료 후 다시 삭제를 진행해주세요.” 메시지가 지속적으로 출력되는 경우



[NOS 삭제 중단 메시지]

- 해당 메시지는 삭제해야 할 필수 파일을 다른 프로그램이 사용하고 있는 경우에 표시되는 메시지입니다.
- 메시지에 표시된 프로그램을 종료 후에 삭제를 다시 진행하여 주시기 바랍니다.

9.2. 제품 일반 문의

1. nProtect Online Security(이하 NOS)는 어떤 제품인가요?

- NOS 는 한 번의 설치만으로 웹 및 PC 이용 시 보안 기능이 동작하도록 설계된 보안 프로그램입니다.
- 웹 및 PC 이용 시 발생할 수 있는 악성코드로 인한 감염, 해킹, 개인정보 유출 등을 방어할 수 있는 다양한 기능을 제공하고 있습니다. 지원 기능은 [1.1.2. 주요 기능]을 참고하시기 바랍니다.

2. NOS 가 왜 필요한가요?

- NOS 는 웹 및 PC 이용 시 발생할 수 있는 악성코드로 인한 감염, 해킹, 개인정보 유출 등을 방어할 수 있는 다양한 기능을 제공하므로 안전한 PC 이용을 위해 반드시 필요한 제품이라고 할 수 있습니다.

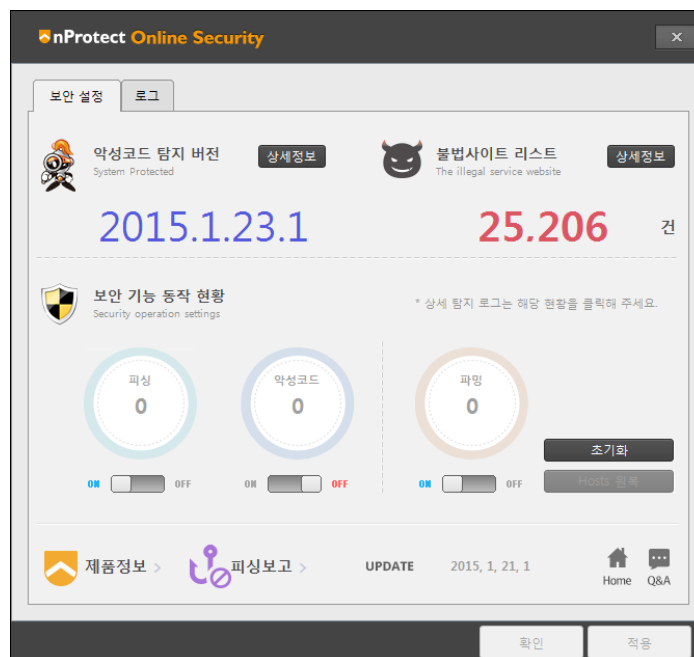
3. NOS UI 는 무엇인가요?

- NOS 가 실행되면 트레이에 주황색 방패 모양의 아이콘이 표시됩니다.



[NOS 트레이 아이콘 이미지]

- 주황색 방패 모양의 아이콘을 더블클릭하면 메인 UI 가 표시됩니다.
- 메인 UI 에는 피싱 방지, 악성코드 실시간 감시, 파밍 방지 기능의 현재 설정값을 보여줍니다.



[NOS 메인 UI]

4. 라이선스가 있나요?

- NOS 는 제휴된 웹 사이트 이용 시 설치 확인을 거쳐 향상된 보안 기능이 실행됩니다. 해당 웹 사이트를 이용하시는 고객님들께는 무료로 제공되고 있습니다.

5. 타 백신프로그램을 사용 중인데 여전히 NOS 가 필요한가요?

- NOS 에는 실시간 감시 등 타 백신프로그램과 유사한 기능도 있지만, 피싱 방지, 키보드 보안, 메모리 보호 등 일반적인 백신프로그램에서 제공하지 않는 기능도 제공하고 있습니다.
- 또한 중복되는 기능이라고 해도 업체별 DB 및 기술력에 차이가 있으므로 세부적인 기능의 차이가 존재합니다. 때문에 동일한 기능을 2 개 사용한다기보다는 더욱 안전한 2 중 보안의 효과를 누릴 수 있습니다.

6. NOS 를 설치한 후에도, 타 백신프로그램이 여전히 필요한가요?

- NOS 는 기본적으로 피싱 방지, 파밍 방지 기능을 제공하다며 제휴된 웹 사이트 이용 시 키보드 보안, 방화벽, 실시간 감시 등 향상된 보안 기능을 제공하고 있습니다.
- 때문에 해당 웹 사이트를 이용하지 않는 상황에서의 보안유지를 위해 신뢰성이 있는 제품을 설치하여 이용하시길 권장해 드립니다.

7. 사용자 개인정보를 저장 또는 서버로 전송하나요?

- NOS 를 이용하기 위해 사용자 개인정보 등의 자료가 필요하지 않으며 키보드 보안의 경우에도 입력한 키보드 값을 암호화, 복호화하는 기능만을 제공하므로 개인정보의 저장, 서버로의 전송 등의 작업은 하지 않습니다.

8. NOS 를 설치하지 않은 PC 에서 제휴된 웹 사이트의 서비스를 이용할 수 있나요?

- 이것은 제휴된 웹 사이트에서 정한 정책에 따라 다릅니다.
- NOS 를 설치하고 서비스를 이용할 것인지, 설치하지 않고 이용할 것인지에 대해 사용자 선택을 제공하는 웹 사이트도 있습니다.

- NOS 는 제휴된 웹 사이트에서 요구하는 보안 기능(특히 키보드 보안 기능)을 제공하고 있으므로 NOS 를 설치하여 보안성도 높이고 원활한 서비스도 제공 받으시길 권장해 드립니다.

9. 제휴 사이트에서 NOS 를 설치하라는 권유를 받고 있는데 어떻게 해야 하나요?

- NOS 는 제휴된 웹 사이트에서 요구하는 보안 기능(특히 키보드 보안 기능)이 동작하고 있지 않으면 보안상 취약하다고 판단하여 서비스를 이용하실 수 없는 페이지도 있습니다.
- 따라서, NOS 를 설치하여 보안성도 높이고 원활한 서비스도 제공 받으시길 권장해 드립니다.

10. 어떤 공격들을 차단할 수 있나요?

- NOS 가 실행 중인 상태에서는 기본적으로 피싱 방지, 파밍 방지 기능을 통해 의도치 않는 변조된 사이트로의 진입을 차단할 수 있습니다.
- 그리고 제휴된 웹 사이트 이용 시 실시간 감시를 통해 해킹툴이 실행되는 부분을 막을 수 있고, 방화벽 기능을 통해 해킹과 관련된 패킷의 in/out 을 차단할 수 있으며 만일 해킹툴이 동작하는 상황이라고 해도 메모리 보호, 키보드 보안 기능으로 중요한 정보의 유출을 차단할 수 있도록 하는, 다중보안구조로 보호가 수행됩니다.

11. 기술적 배경지식이 많은 파워유저 입니다. 그래도 NOS 를 설치해야 하나요?

- 기술적인 지식이 많고 PC 에 여러 보안 솔루션이 있는 상태에서도 NOS 는 제휴된 웹 사이트에서 요구하는 보안 기능(특히 키보드 보안 기능)이 동작하고 있지 않으면 보안상 취약하다고 판단하여 서비스를 이용하실 수 없는 페이지도 있습니다.
- 따라서, NOS 를 설치하여 보안성도 높이고 원활한 서비스도 제공 받으시길 권장해 드립니다.

12. 기술적 배경지식이 전혀 없는 초보사용자 입니다. 그래도 NOS 를 사용할 수 있을까요?

- NOS 는 설치 또는 실행이 간편하고 기본 설정으로도 안전한 보안이 가능하기 때문에 기술적 지식이 없어도 얼마든지 사용이 가능합니다.
- 환경설정 등이 필요할 경우에도 도움말을 참고하시면 필요한 설정을 무리 없이 조절 가능합니다.

13.NOS 가 진단 및 치료할 수 있는 바이러스의 종류에는 어떤 것들이 있나요?

- Trojan, Hack, Worm, Backdoor, Malware, Adware, Spyware, Grayware 등 다양한 해킹툴 및 악성코드에 대한 진단 및 치료가 가능합니다.

14.NOS 는 PC 내에서 항상 실행되고 있는 부분이 있나요?

- 해당 웹사이트 보호를 위해 피싱 방지와 파밍 방지 기능이 동작하고 있습니다.
- 피싱, 파밍이 무엇인지에 대해서는 [6.2.16]을 참고하시기 바랍니다.

15.NOS 는 어떤 기능들을 가지고 있나요?

- NOS 는 다음과 같은 기능을 제공합니다.
 - 피싱 방지
 - 파밍 방지
 - 실시간 프로그램 감시 및 진단/치료
 - 악성 프로그램 검사 및 치료
 - 통신프로그램 인증
 - 공유폴더 접근 제어
 - 키보드입력 보호
 - 부정거래방지

16.피싱, 파밍이란 무엇인가요?

- 피싱이란, 주로 금융기관, 정부 등의 이름으로 이메일을 발송, 변조된 웹사이트로 접속을 유도하여 개인정보를 탈취하는 사기수법 입니다. NOS 에서는 변조된 것으로 확인된 사이트 정보를 기준으로 해당 사이트 접속 시 경고 메시지를 발생시키는 방법으로 피싱을 방지합니다.
- 파밍이란, 넓은 의미에서는 피싱의 한 종류로, 정상적인 사이트 URL 로 접속하여도 변조된 사이트로 강제로 접속하게 만들어 개인정보를 탈취하는 사기수법 입니다. NOS 에서는 Hosts 파일 변조를 탐지하여 변조가 있을 시 메시지를 발생시키는 방법으로 파밍을 방지합니다.

9.3. 제품 운영 관련 문의

1. NOS 패턴 업데이트는 어떻게 하나요?

- NOS 는 악성코드에 대한 Cloud 엔진을 사용하기 때문에 PC 에 패턴을 업데이트하지 않아도 항상 최신의 패턴을 제공받습니다.

2. NOS 버전 업데이트는 어떻게 하나요?

- NOS 는 자동업데이트 기능을 가지고 있어, 제품 실행 시 자동으로 최신 버전으로 업데이트를 진행하고 있습니다.
- 또한, 컴퓨터를 켜 놓은 상태에서는 12 시간마다 업데이트를 진행하고 있습니다.

3. NOS 가 동작 중이라는 것을 어떻게 확인할 수 있나요?

- NOS 를 설치한 후에 모니터 화면 우측 하단에 표시되는 주황색 방패 모양의 트레이 아이콘을 확인하시면 됩니다.



[NOS 트레이 아이콘 이미지]

4. NOS 를 시작할 때마다 서버와 통신을 하는 이유는 뭔가요?

- NOS 제품이 실행될 때 서버에 최신 버전이 존재하는지 확인하고 존재한다면 자동으로 업데이트하기 위해서 입니다.

5. PC 에 다수의 사용자계정이 존재하는 경우, 모든 사용자를 보호해 주나요?

- NOS 제품은 기본적으로 모든 사용자 용으로 설치되기 때문에 설치 후 다른 사용자 계정으로 이용 시에도 보호가 가능합니다.

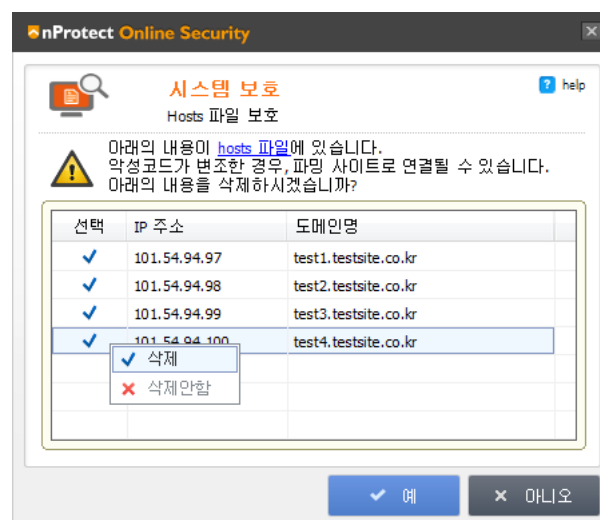
6. NOS 가 왜 사용자의 로그인 정보를 감시하나요?

- NOS 는 사용자가 웹 상에 로그인 정보를 입력할 경우 이 내용을 암호화하는 키보드 보안 기능을 가지고 있으므로 감시한다고 볼 수 있으나 해당 정보를 따로 저장하거나 외부로 전송하는 동작은 없으니 안심하고 이용해 주셔도 됩니다.

7. 공공장소에서 Wi-Fi 를 이용하여 제휴 사이트에 접속 시, 해킹위험으로부터 충분히 안전한가요?

- 위험성이 높은 오픈된 인터넷 환경에서 PC 를 사용하더라도 NOS 가 실행된 상태라면 피싱 방지, 파밍 방지, 방화벽, 실시간 감시, 키보드 보안 기능이 있으므로 해킹으로부터 안전한 이용이 가능합니다.

8. 설치 후 Hosts 파일이 변조되었다는 메시지가 발생합니다.



[Hosts 파일 보호 메시지]

- 해당 메시지는 파밍에 악용될 수 있는 Hosts 파일이 변조된 것을 탐지했을 때 발생합니다.
- 메시지에서 보여주는 변조된 내용을 확인 후 사용환경에 따라 직접 추가한 경우에는 선택을 "삭제안함"으로 선택하고 사용자 모르게 추가된 내용으로 의심된다면 "삭제"를 선택하여 기본값으로 복원 후 "예"를 클릭하시면 됩니다.
- 파밍 방지 기능은 NOS 의 설정에서 on/off 가 가능합니다.

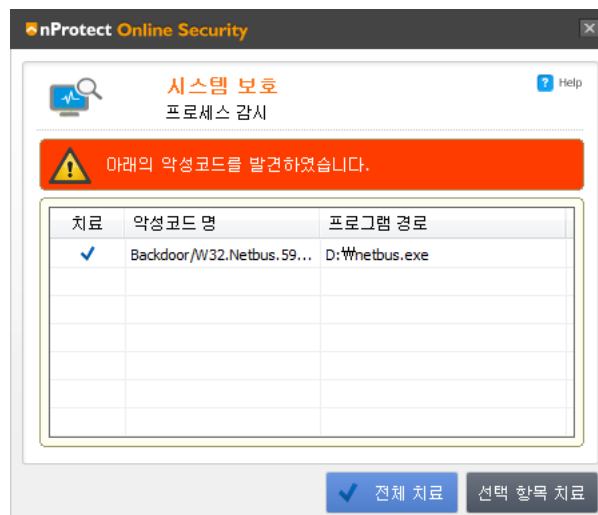
9. 설치 후 피싱 경고 메시지가 발생합니다.



[피싱사이트 접속 경고 메시지]

- 해당 메시지는 피싱 사이트로 확인된 URL 에 접속을 시도할 시 발생합니다.
- 현재 접속하는 사이트가 정상적인 사이트가 맞는지 다시 한번 면밀히 확인해주시고 확인되기 전까지는 어떠한 개인정보도 입력하지 않도록 권장합니다.
- 피싱 방지 기능은 NOS 의 설정에서 on/off 가 가능합니다.

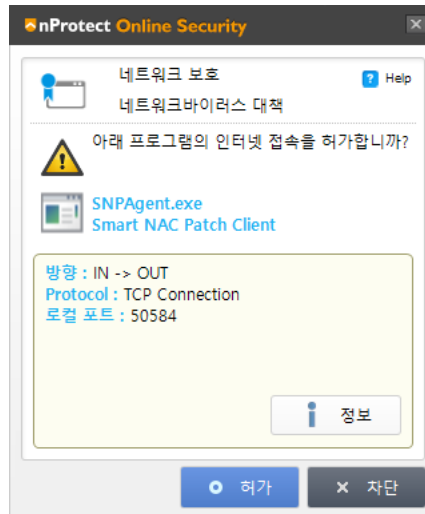
10. 악성코드를 발견하였다는 메시지가 발생합니다.



[악성코드 발견 메시지]

- 해당 메시지는 실시간 감시가 켜져 있는 상태에서 실행중인 악성코드가 있을 시 발생합니다.
- 프로그램 경로 등을 확인 후 [전체 치료] 또는 [선택 항목 치료]로 악성코드를 치료하시면 됩니다.

11. 네트워크 보호 메시지가 발생합니다.



[네트워크 보호 메시지]

- 해당 메시지는 방화벽 기능이 켜져 있는 상태에서 네트워크에 접속하려는 프로그램이 있을 시 발생합니다.
- 악성코드처럼 위험한 항목만 탐지하는 것이 아니므로 접속 시도 프로그램을 확인 후 정상적인 프로그램일 경우 [허가]를 클릭해주시면 됩니다.[차단]을 클릭하시는 경우 해당 프로그램은 네트워크에 접근할 수 없습니다.
- 허용, 차단 항목은 NOS의 설정에서 추가/변경이 가능합니다.

9.4. 장애 관련 문의

1. NOS를 설치하는 중에 발생할 수 있는 문제에는 어떤 것들이 있나요?

- NOS의 설치가 정상적으로 진행되지 않는 경우에는 PC 및 네트워크의 환경적인 부분을 확인할 필요가 있습니다.
- PC에 설치된 보안프로그램 등에서 NOS의 설치 및 실행을 차단하거나, 네트워크 보안에서 NOS가 실행되면서 접속하는 네트워크(특히 업데이트 서버)를 차단하거나, 그리고 사내에서 실시되는 다양한 보안 기능들(예를 들면 Active Directory 정책)과의 충돌이 발생할 수 있습니다.

2. 몇몇 특정 웹사이트에 한해 인터넷 연결이 되지 않는 경우, 그 원인에는 어떤 것들이 있나요?

- NOS 가 동작할 때 특정 웹사이트로의 연결에 문제가 있는 경우, 방화벽, 실시간 감시, 피싱 방지 등의 기능과 연관이 있을 수 있습니다.
- 실제 보안상에 문제가 없는 웹사이트로 확인될 경우 NOS 의 기능을 일시 중지하고 테스트를 진행할 필요가 있습니다.

3. NOS 아이콘을 클릭하더라도 아무런 동작을 하지 않는 경우, 그 원인에는 어떤 것들이 있나요?

- NOS 의 트레이 아이콘을 더블클릭, 오른쪽 마우스 클릭해도 아무런 동작이 없는 경우, NOS 가 정상적으로 설치/실행되지 않은 경우 입니다. [5.2 삭제하기]와 [5.1 설치하기]를 참고하여 NOS 를 삭제 후 재설치하여 다시 확인해보시기 바랍니다.

4. NOS 사용 중 PC 에 블루스크린이 발생한 경우 어떻게 해야 하나요?

- NOS 사용 중 블루스크린이 발생한 경우, PC 환경 또는 PC 에 구동 중인 다른 프로그램과의 충돌 문제인지 확인해야 합니다. 다음의 방법으로 확인해 보시기 바랍니다.
 - [03. 운영 환경]을 참고하여 현재 사용자의 PC 환경이 미지원 환경에 해당하지는 않는지 확인 바랍니다.
 - 또한 USB 키보드의 경우 일시적으로 이탈 증상이 발생하는 경우도 있으니 USB 키보드의 연결해제 후 재연결(다른 USB 포트 추천)하여 확인 바랍니다.
 - 현재 사용 중인 타 보안 프로그램을 모두 중지(혹은 삭제)하고 NOS 를 재설치하신 후 확인 바랍니다.
- 만일 문제가 계속될 경우 저희 지원센터로 문의 바랍니다.

5. 키보드 입력이 정상적으로 되지 않는 경우, 어떻게 해야 하나요?

- NOS 가 실행 중인 상태에서 키보드 입력이 정상적으로 되지 않는 경우, 미지원 환경 또는 NOS 외에 다른 키보드 보안 프로그램이 있거나 키후커, 키로거 등의 키 입력 값을 가로채는 악성코드가 실행 중인지 확인해야 합니다. 다음의 방법으로 확인해 보시기 바랍니다.
 - [03. 운영 환경]을 참고하여 현재 사용자의 PC 환경이 미지원 환경에 해당하지는 않는지 확인 바랍니다.
 - 또한 USB 키보드의 경우 일시적으로 이탈 증상이 발생하는 경우도 있으니 USB 키보드의 연결해제 후 재연결(다른 USB 포트 추천)하여 확인 바랍니다.

- 현재 사용 중인 타 보안 프로그램을 모두 중지(혹은 삭제)하고 NOS 를 재설치하신 후 확인 바랍니다.
- 백신 프로그램으로 전체 검사를 수행하여 진단되는 악성코드 치료 후 재시도 바랍니다.
- 만일 문제가 계속될 경우 저희 지원센터로 문의 바랍니다.

6. NOS 설치 후 PC 가 현저히 느려졌다면 어떻게 해야 하나요?

- NOS 설치 후 PC 가 현저히 느려지는 증상이 발생한다면 현재 PC 내에 실행 중인 프로그램과의 문제 또는 보안상의 문제가 있는지 확인해야 합니다. 다음의 방법으로 확인해 보시기 바랍니다.
 - [5.2 삭제하기]와 [5.1 설치하기]를 참고하여 NOS 를 삭제 후 재설치하여 다시 확인해보시기 바랍니다.
 - 현재 사용 중인 타 보안 프로그램을 모두 중지(혹은 삭제)하고 NOS 를 재설치하신 후 확인 바랍니다.
 - 백신 프로그램으로 전체 검사를 수행하여 진단되는 악성코드 치료 후 재시도 바랍니다.
- 만일 문제가 계속될 경우 저희 지원센터로 문의 바랍니다.

7. 타 백신프로그램이 NOS 에 대해 보안위험경고를 한 경우, 어떻게 해야 하나요?

- 백신프로그램의 경우 어떤 프로그램이든 오진단의 가능성을 가지고 있습니다. 만일 타 백신프로그램에서 NOS 를 악성코드 등으로 진단하는 경우 해당 백신프로그램 개발사로 오진여부 확인이 필요합니다.

8. 제휴 웹사이트를 나온 이후에도 계속해서 NOS 가 주황색 아이콘으로 바뀌지 않고 녹색 아이콘인 경우가 있습니다. 어떻게 해야 하나요?

- 제휴 웹사이트를 나온 이후에도 NOS 가 녹색 아이콘으로 동작하고 있는 경우, 우선 열려있는 브라우저를 모두 종료하고 확인바랍니다. 만일 그래도 계속 동작이 유지될 경우, 보이지 않는 브라우저가 실행되고 있을 수 있으니 작업 관리자를 실행하여 브라우저가 프로세스에 실행 중인지 확인하여 종료해 주시면 됩니다.

9. 기존에 알려진 타 보안프로그램과의 충돌이슈에는 어떤 것들이 있나요?

- NOS 에서 제공하는 기능 중 방화벽과 키보드 보안 기능의 경우, 타 프로그램과의 충돌이라고 할 수 있는 문제가 발생할 가능성이 있습니다. 한 PC 에서 중복된 방화벽,

키보드 보안이 동작하는 경우 블루스크린이나 키보드 입력 문제 등이 발생할 수 있으니
가급적이면 다른 부분을 중지하고 이용을 권장합니다.

- 현재까지 특정 타 보안프로그램과의 충돌 이슈는 없습니다.

9.5. FDS 수집항목

1. PC 단말정보수집 항목

- 각 항목의 값들은 고객 PC 에 따라 수집되지 않을 수 있습니다.
- 리눅스의 경우에는 관리자권한이 없을 경우 PC 식별정보 HDD 시리얼, 메인보드 시리얼, CPU 시리얼을 얻어오지 못합니다. 부트 UUID 를 사용하여 PC 를 식별하도록 하십시오. (단, 타 운영체제에서는 해당값이 없음)

대분류	중분류	설명	윈도우		리눅스		맥	
			관리자	사용자	root	일반	root	일반
하드웨어	CPU	CPU ID	O	O	O	X	X	X
		CPU 식별번호	O	O	O	O	X	X
		CPU 프로세스명	O	O	O	O	O	O
	HDD	HDD 모델	O	O	O	O	O	O
		HDD 시리얼번호	O	O	O	X	O	O
		HDD 볼륨명	O	O	O	O	X	X
	Mother Board	Main Board 제조사	O	O	O	X	X	X
		Main Board 제품명	O	O	O	X	X	X
		Main Board 시리얼번호	O	O	O	X	X	X
네트워크	Network	게이트웨이 IP	O	O	O	O	O	O
		로컬 PC에 설정된 IP	O	O	O	O	O	O
		MAC 주소	O	O	O	O	O	O
		NAT IP	O	O	O	X	O	O
		Proxy IP	O	O	O	O	O	O
		Proxy Server IP	O	O	O	O	O	O
		VPN IP	O	O	O	O	X	X
		VPN로컬설정IP(사용자PC)	O	O	O	O	X	X
		원격연결시 접속 IP	O	O	O	O	X	X
		원격연결 사용여부	O	O	O	O	X	X
		MAC 주소 변조 여부	O	O	O	O	X	X
소프트웨어	OS	키보드 타입	O	O	O	O	O	O
		USB 시리얼	O	O	O	O	X	X
		OS 코드	O	O	X	X	O	O
		OS 버전	O	O	X	X	O	O
		OS 구분(신규)	O	O	O	O	O	O
		OS 구분(이전)	O	O	O	O	O	O
		OS 주버전	O	O	O	O	O	O

		OS 하위버전	O	O	O	O	O	O
		OS 상세버전	O	O	O	O	O	O
		윈도우 보안패치 적용여부	X	X	X	X	X	X
		윈도우 서비스 팩	O	O	X	X	X	X
		OS 언어팩	O	O	O	O	O	O
보안	Security Function	Proxy 사용여부	O	O	O	O	O	O
		로컬 PC 방화벽설정여부	O	O	O	O	O	O
		원격접속 허용여부	O	O	O	O	X	X
		잉카인터넷 방화벽 상태	O	O	O	O	X	X
		잉카인터넷 키보드보안 상태	O	O	X	X	X	X
		VPN 사용여부	O	O	O	O	X	X
		USB에 공인인증서 폴더 존재여부	O	O	X	X	X	X

[표 6-1] PC 단말정보수집 항목

2. 모바일 단말정보수집 항목

- 각 항목의 값들은 단말기 및 운영체제에 따라 수집되지 않을 수 있습니다.

설명	Android		iOS	상세 설명
	Cell	WiFi		
OS 종류	O	O	O	모바일 OS 종류
UUID	O	O	O	기기별 고유 식별번호
추가 UUID	O	O	O	추가 식별번호
OS 버전	O	O	O	Android, iOS 의 숫자 형식으로 된 OS 버전 예) Android : 4.2.2, iOS : 6.1.3
OS 코드네임	O	O	X	Android 에서만 수집되는 항목이며 Android 의 숫자형식의 OS 버전과 매칭되는 OS 코드명 예) JELLY_BEAN
제조사	O	O	O	제조사 이름 예) samsung, lge, apple, google
모델명	O	O	O	기기의 고유 모델명. iOS 인 경우 실제 판매 제품명이 아닌 OS 에서 제공하는 기기 명칭(아이폰 5S 인 경우 "iPhone 6,2")
음성통화상태	O	X	X	통화상태
데이터통신상태	O	X	X	데이터통신 상태
기기 고유 번호	O	X	X	기기 고유 번호 (IMEI)
단말기 전화번호	O	X	X	전화번호

네트워크 국가코드	O	X	X	네트워크를 제공하는 국가의 국가코드
SIM 카드 국가코드	O	X	X	SIM 카드를 제공한 국가의 국가코드
네트워크 망 사업자 코드	O	X	X	망 사업자코드
네트워크 망 사업자명	O	X	X	망 사업자명
SIM 카드 망사업자 코드	O	X	X	SIM 카드를 제공한 사업자코드
SIM 카드 망사업자명	O	X	X	SIM 카드를 제공한 사업자명
네트워크 타입	O	X	X	기기가 제공하는 네트워크 망 타입
단말기 종류	O	X	X	기기가 제공하는 음성 통신 망 타입
SIM 카드 일련번호	O	X	X	SIM 카드의 일련번호
SIM Card 상태	O	X	X	SIM 카드의 상태
가입자 ID	O	X	X	고유한 가입자 아이디
연결된 네트워크 정보	O	O	O	연결된 네트워크 정보
Mac Address	O	O	△	Mac Address 수집(xx:xx:xx:xx:xx:xx)
루팅/탈옥 여부	O	O	O	android: Rooting 여부, iOS: Jailbreak 여부
앱 설치 및 실행 여부	O	O	O	특정 앱의 설치 및 실행 여부
정보 수집 시작 시간	O	O	O	정보 수집 시작 시간(yyyy-MM-dd HH:mm:ss)
정보 수집 소요 시간	O	O	O	정보 수집 소요 시간 - 밀리초
수집 SDK 빌드 번호 (버전)	O	O	O	수집 SDK 빌드 버전 (x.yy.zzzz)
앱 패키지 이름	O	O	O	앱 패키지 이름
앱 버전	O	O	O	앱 버전 - 문자열
표본 데이터 해시값	O	O	O	해쉬값

[표 6-2] 모바일 단말정보수집 항목

9.6. 미지원 환경

1. FDS 미지원 환경

- 일반적인 환경에서 FDS 서비스가 동작하지만, 아래와 같은 특수한 환경에서는 FDS 서비스의 동작을 보장하지 않습니다.

■ 가상화 (Virtualization) 환경

가상화 환경이란, 물리적인 환경을 추상화하여 실제로는 존재하지 않지만 가상으로 존재하는 것처럼 구현한 환경을 말합니다. 가상의 환경이다 보니 수집되는 시스템의 정보가 부정확하고, 정보도 신뢰할 수 없게 되므로, FDS 서비스는 가상화 환경을 지원하지 않습니다. 가상화 환경에는 가상 컴퓨터, 가상 하드 디스크, 클라우드 PC 등이 포함됩니다. 아래는 현재까지 보고된 가상화 환경들의 종류입니다.

✓ VMWare (브이엠웨어)

VMWare 는 가상머신(Virtual Machine), 즉 가상의 컴퓨터를 사용할 수 있게 해주는 소프트웨어입니다.

✓ VirtualBox (버추얼박스)

VMWare 와 마찬가지로 가상머신을 사용할 수 있게 해주는 소프트웨어 입니다. Oracle 社에서 무료로 배포합니다.

✓ VirtualPC (버추얼피씨)

Microsoft 社에서 배포하는 가상머신을 사용할 수 있게 해주는 소프트웨어입니다.

✓ BootCamp (부트캠프)

Apple 社에서 제작한 소프트웨어로 가상머신을 사용할 수 있게 해주는 소프트웨어 입니다. 일반적으로 Mac 에서 윈도우를 사용하려고 할 때 사용합니다.

■ 클라우드 PC (Cloud PC)

클라우드(Cloud)란, 가상화 환경과 비슷한 개념으로 인터넷 기반의 서비스를 말합니다. 예를 들어 컴퓨터의 하드 디스크 같은 기억공간을 내 PC 가 아닌 인터넷 상의 서버로 대체해서 사용하는 방식입니다.

클라우드 PC 는 이런 클라우드 개념을 이용하여 웹을 기반으로 한 소프트웨어 서비스 입니다. 클라우드 PC 도 가상의 환경이기 때문에 수집되는 시스템의 정보가 부정확하고 신뢰성도 낮아 FDS 서비스에서 지원하지 않는 환경입니다.

■ 레이드 (RAID, Redundant Array of Inexpensive Disk)

하드 디스크와 같은 보조 기억장치로, 여러 개의 디스크를 하나처럼 사용할 수 있게 하여 성능과 신뢰성을 향상시킨 저장장치입니다.

레이드 저장장치는 동일한 데이터가 복수 개의 디스크에 저장되고, 복수 개의 디스크를 사용하기 때문에 수집하는 정보가 일정하지 않아 수집 정보의 신뢰도가 낮아 지원하지 않습니다.

■ SSD (Solid State Drive)

하드 디스크와 같은 보조 기억장치로 하드디스크에 비해 훨씬 빠른 속도와 적은 소음이 장점인 저장장치입니다. 램(RAM) 방식과 플래시(FLASH) 방식이 있으나, 두 방식 모두 FDS 서비스에서는 지원을 하지 않습니다.

■ POS 단말기

POS 단말기는 POS 기기라고도 하며, 금전등록관리기와 컴퓨터를 결합한 시스템을 말합니다. POS 단말기는 주로 식당이나 매장 등에서 사용하는데, 인터넷 연결도 되므로 간혹 컴퓨터로도 사용하는 사례가 있습니다. 하지만 일반적인 컴퓨터 장비가 아니기 때문에 시스템 정보가 누락되기도 하여 FDS 서비스에서 지원을 하지 않습니다.

■ IE 인식불가 (다회선 이용을 위한 브라우저 설정 값 변경)

KT 의 다회선 제한을 회피하기 위해 일부 사용자들이 행하는 방법으로, 인터넷 브라우저를 다른 브라우저로 인식하게끔 설정(레지스트리) 값을 변경하고 있습니다. 이 방식을 이용하면 FDS 서비스를 설치할 수 있지만 시스템 정보가 달라져 다른 PC 로 오픈할 수 있습니다.

2. 키보드보안 미지원 환경

- 일반적인 환경에서 키보드보안 서비스가 동작하지만, 아래와 같은 특수한 환경에서는 키보드보안 서비스의 동작을 보장하지 않습니다.

■ 가상화 (Virtualization) 환경

가상화 환경이란, 물리적인 환경을 추상화하여 실제로는 존재하지 않지만 가상으로 존재하는 것처럼 구현한 환경을 말합니다. 가상환경으로 키 입력이 발생될 때 물리적인 환경과는 다르게 소프트웨어적으로 키 입력이 전달됨으로 키보드보안은 가상화 환경을 지원하지 않습니다.

가상화 환경에는 가상 컴퓨터, 가상 하드 디스크, 클라우드 PC 등이 포함됩니다.

■ 한글 미 보호 사유

알파벳은 1 글자당 1 개의 CODE 가 있는 반면 한글을 조합에 따라 그 CODE 가 변경 됩니다. 해당 CODE 를 전부 핸들링 할 수 없으며 이미 입력된 키와 입력 되는 키와의 조합형태는 IME 프로그램에서 설정 하므로 키보드보안 모듈에서는 이를 지원 할 수 없습니다. 따라서 "영어, 숫자, 특수문자"에 한해 키보드 입력을 보호합니다.

10. 문의 및 고객센터

10.1. 프로그램 공급자

공급자명 : (주)잉카인터넷

주소 : 152-740) 서울시 구로구 디지털로 26 길 5, 12 층

대표전화 : 02) 6220-8000

FAX : 02) 6220-8080

10.2. 고객센터 센터

nProtect Online Security 와 관련된 문의사항은 잉카인터넷으로 연락하시기 바랍니다.

고객지원센터 : 1566-0808

업무시간 : 평일 09:00~18:00, 토요일 : 09:00~13:00, 공휴일 : 휴무

1. 별첨 – 단말정보수집 수집항목

1.1. 단말정보수집 수집항목

- PluginFreeDTO 객체의 변수목록

구분	변수타입	변수명 ⁶¹	기본값	설명
네트워크	String	ipNat	null	NAT IP
	String	natErrCd	null	NAT IP 획득 에러 코드
	String	natCntryCd	null	NAT IP 국가코드
	String	ipEth0	null	ETH0 IP
	String	ipEth1	null	ETH1 IP
	String	ipEth2	null	ETH2 IP
	String	gipEth0	null	ETH0 GATEWAY IP
	String	gipEth1	null	ETH1 GATEWAY IP
	String	gipEth2	null	ETH2 GATEWAY IP
	String	macEth0	null	ETH0 MAC
	String	macEth1	null	ETH1 MAC
	String	macEth2	null	ETH2 MAC
	String	macLogicalEth0	null	ETH0 Logical MAC
	String	macLogicalEth1	null	ETH1 Logical MAC
	String	macLogicalEth2	null	ETH2 Logical MAC
	boolean	forgeryMac	false	ETH MAC 변조여부
	boolean	forgeryMacEth0	false	ETH0 MAC 변조여부
	boolean	forgeryMacEth1	false	ETH1 MAC 변조여부
	boolean	forgeryMacEth2	false	ETH2 MAC 변조여부
	String	prxyYn	N	Proxy 사용 여부
	String	ipPrxyLcal	null	Proxy 로컬설정 IP
	String	cntryPrxyLcal	null	Proxy 로컬설정 IP 국가코드
	String	ipPrxy	null	Proxy IP
	String	cntryPrxy	null	Proxy IP 국가코드
	String	vpnYn	N	VPN 사용 여부
	String	ipVpn	null	VPN IP
	String	cntryVpn	null	VPN IP 국가코드
	String	ipVpnLcal	null	VPN 로컬설정
	String	cntryVpnLcal	null	VPN 로컬설정 국가코드

⁶¹ 본 변수들의 값을 얻는 방법은 getter 메소드를 호출하여 얻어옵니다. 예를 들어 숫자(int)/문자열(String) 변수일 때 변수명이 ipNat 라면 getIpNat() 함수를 호출하여 해당값을 얻어올 수 있습니다. 또한 참/거짓(boolean) 변수일 때 변수명이 forgeryMac 이면 isForgeryMac() 함수를 호출하여 해당값을 얻을 수 있습니다.

	String	ipRmt	null	원격지 IP
	String	cntryRmt	null	원격지 IP 국가코드
	String	ipEtc	null	예비 IP
소프트웨어	String	osCd	null	운영체제 코드
	String	osLocale	null	운영체제 언어팩 코드
	String	osGuid	null	운영체제 식별정보
	String	osRayn	null	운영체제 원격접속허용여부
	String	osSpvsn	null	운영체제 서비스팩 코드
	String	osVsnCd	null	운영체제 버전
	String	osFirewallCd	null	운영체제 방화벽 설정 코드
	String	osTypeCd	null	운영체제 구분 코드
	String	osPatchsts	null	운영체제 패치 상태
	String	bootUuid	null	운영체제 부트 UUID
	String	bwVsnCd	null	브라우저 버전
	String	ieVsn	null	브라우저 버전코드
	String	ieDetvsn	null	브라우저 상세버전
	String	iePrdctid	null	브라우저 PRODUCT ID
하드웨어	String	hddMdel	null	HDD 모델
	String	hddSrial	null	HDD 시리얼
	String[]	hddSrialAll	null	HDD 시리얼(전체)
	String	usbSrial1	null	USB 시리얼 1
	String	usbNpkiYn1	N	USB NPKI 폴더 여부
	String	usbSrial2	null	USB 시리얼 2
	String	usbNpkiYn2	N	USB NPKI 폴더 여부
	String	usbSrial3	null	USB 시리얼 3
	String	usbNpkiYn3	N	USB NPKI 폴더 여부
	String	mbSrial	null	마더보드 시리얼
	String	mbMnft	null	마더보드 제조업체
	String	mbPrdct	null	마더보드 PRODUCT
	String	cpuPrcesnm	null	CPU 프로세스명
	String	cpuldnr	null	CPU 식별자
	String	cpuld	null	CPU ID
	String	keyboardType	null	키보드 타입 코드
보안	String	stsDhack	null	해킹차단기 상태 코드
	String	stsKey	null	키보드보안 상태 코드
	boolean	remoteDesktop	false	원격데스크톱 접속여부
	String	remoteDesktopCode	00	원격데스크톱 구분코드
	boolean	virtualOs	false	가상운영체제 접속여부
	String	virtualOsCode	00	가상운영체제 구분코드
	String[]	certInfo	null	저장된 인증서 목록
기타	int	elapsedTime	-1	로그수집소요시간

	String	totalHash	null	Total 해쉬(디버그모드일 때만 수집)
	String	totalHashYn	N	Total 해쉬 일치 여부
	String	efdsVersion	null	단말정보수집모듈 버전
	String	logYn	N	동작 여부(자바스크립트)
	String	executeYn	N	정상수집 여부(서버키발급비교)
	String	httpReferer	null	이전페이지 정보

[표 1] PluginFreeDTO 객체 구조

1.2. 무설치 단말정보수집 수집항목

- PluginFreeDeviceDTO 객체의 변수목록

타입	상세속성	변수명 ⁶²	설명
Date		serverStamp	수집시각
String		uuid ⁶³	단말기 고유식별정보
String		natIp	IP 주소
String		clientIp ⁶⁴	IP 주소(자바스크립트기준)
String		remoteIp ⁶⁵	IP 주소(WAS 서버기준)
String		remoteHost	접속지 호스트명
Integer		remotePort	접속지 연결포트
String		proxyIp ⁶⁶	프록시 서버 IP
String		proxyHost	프록시 서버 호스트
String		requestHost	요청 호스트
String		sessionId	세션 ID
boolean		sessionIsNew	신규세션여부

⁶² 본 변수들의 값을 얻는 방법은 getter 메소드를 호출하여 얻어옵니다. 예를 들어 숫자(int)/문자열(String) 변수일 때 변수명이 ipNat 라면 getIpNat() 함수를 호출하여 해당값을 얻어올 수 있습니다. 또한 참/거짓(boolean) 변수일 때 변수명이 forgeryMac 이면 isForgeryMac() 함수를 호출하여 해당값을 얻을 수 있습니다.

⁶³ 식별정보는 수집된 정보를 바탕으로 WAS 서버에서 생성되며 웹 브라우저의 저장소에 식별정보를 저장합니다. 웹 브라우저를 신규로 설치했을 경우에는 이 식별정보는 변경됩니다.

⁶⁴ Microsoft Internet Explorer 8 이하의 브라우저 환경 또는 방화벽 차단 등으로 외부 네트워크가 되지 않는 환경에서는 값이 수집되지 않습니다.

⁶⁵ WAS 서버에서 바라본 마지막 접속지정보입니다. 이 정보는 실제 사용자의 IP와 다를 수 있습니다.

⁶⁶ Http Header 정보를 읽어서 프록시를 통해 우회하였는지 확인합니다. 이 값이 세팅되지 않았다고 하여 프록시를 통해 우회하지 않았다고 판단해서는 안됩니다. 악의적인 프록시의 경의 헤더정보를 넣지 않을 수 있습니다.

Date		sessionCreateStamp	세션 생성 시각
Date		sessionLastAccessStamp	세션 마지막 접속 시각
boolean		exchange ⁶⁷	데이터 교환 여부
boolean		indirect ⁶⁸	우회여부
AgentData		agentData⁶⁹	UserAgent 파싱정보
	String	browserType	브라우저 종류
	String	browserName	브라우저 명
	String	browserRenderingEngine	브라우저 렌더링 엔진
	String	browserVersion	브라우저 버전
	String	browserMajorVersion	브라우저 주버전
	String	browserMinorVersion	브라우저 상세버전
	String	browserManufacturer	브라우저 제조사
	String	osName	운영체제명
	String	osManufacturer	운영체제 제조사
	String	osGroup	운영체제 그룹
	String	deviceType	단말기 종류
NavigatorData		navigatorData⁷⁰	웹 브라우저 정보
	String	product	웹 브라우저 엔진 명
	String	productSub	웹 브라우저 빌드 버전
	String	appName	웹 브라우저 코드명
	String	appVersion	웹 브라우저 명
	String	appMinorVersion	웹 브라우저 버전
	String	platform	웹 브라우저 컴파일 플랫폼
	String	userAgent	웹 브라우저 정보
	String	vendor	웹 브라우저 제조사
	String	vendorSub	웹 브라우저 제조사 버전번호
	String	language	웹 브라우저 사용언어
	String	userLanguage	사용자 언어
	String	browserLanguage	브라우저 언어
	String	systemLanguage	시스템 언어
	String	cpuClass	브라우저 CPU bit 수
	String	oscpu	운영체제 플랫폼 정보

⁶⁷ 단말정보가 수집되고 수집된 값을 검증하는 과정을 거쳤는지 확인합니다. 검증에 실패하거나 기존에 저장된 식별값이 없을 경우에는 신규로 식별값을 생성합니다.

⁶⁸ clientIp 주소와 remoteIp 주소가 일치하지 않을 경우 우회되었다고 판단합니다. clientIp 가 수집되지 않으면 이 속성은 true 를 반환합니다.

⁶⁹ window.navigator.userAgent 정보를 파싱하여 단말정보를 수집합니다.

⁷⁰ window.navigator 객체의 속성값에서 단말정보를 수집합니다.

	String	buildID	웹 브라우저 빌드 ID
	String	doNotTrack	제 3 자 쿠키제공 설정
	Boolean	msDoNotTrack	제 3 자 쿠키제공 설정여부(MS IE)
	Boolean	cookieEnabled	쿠키사용가능 여부
	Boolean	cookiesTest	쿠키 사용 테스트 결과
	Boolean	javaEnabled	자바애플릿 사용가능여부
	Boolean	flashEnabled	플래시 사용가능여부
	Boolean	silverlightEnabled	실버라이트 사용가능여부
	Boolean	geolocationCapable	위치정보 사용가능 여부
	Boolean	onLine	온라인 여부
	Integer	maxTouchPoints	터치입력 최대수
	List<String>	languages	등록된 브라우저 언어목록
	Boolean	webLocalStorageCapable	로컬 스토리지 사용여부
	Boolean	webLocalStorageTest	로컬 스토리지 테스트 여부
DocumentData		documentData⁷¹	웹 문서 정보
	String	URL	문서 URL
	String	documentURI	문서 URL
	String	origin	문서의 근원 주소
	String	compatMode	표준 부합모드
	String	characterSet	문서 문자셋
	String	charset	문서 문자셋
	String	inputEncoding	문서를 파싱할 때 쓴 인코딩
	String	contentType	MIME 헤더 Content-Type
	String	domain	문서 도메인
	String	referrer	이 문서로 링크된 페이지 URL
	String	title	문서 제목
	String	lastModified	문서 변경일
	String	cookie	쿠키 문자열
	String	readyState	로딩상태
	String	designMode	디자인 모드 여부
	String	visibilityState	BODY 의 보여짐 여부
	Boolean	spellcheck	맞춤법 검사 지원여부
	String	namespaceURI	문서 파싱 정보
	Integer	childElementCount	자식 항목 개수
	String	baseURI	현재 문서 경로
	Boolean	isConnected	온라인 여부
DisplayData		displayData⁷²	화면정보

⁷¹ window.document 객체의 속성값에서 단말정보를 수집합니다.

⁷² window.screen 객체의 속성값에서 단말정보를 수집합니다.

	Integer	width	화면 너비
	Integer	height	화면 높이
	Integer	availWidth	실제화면에서 사용가능한 너비
	Integer	availHeight	실제화면에서 사용가능한 높이
	Integer	colorDepth	사용가능한 색상수
	Integer	pixelDepth	한 픽셀당 비트수
	Float	pixelRatio	CSS 픽셀에 대한 현재 디스플레이 장치 상의 하나의 물리적인 픽셀의 비율
	Integer	innerWidth	브라우저 내용 너비
	Integer	innerHeight	브라우저 내용 높이
	Integer	outerWidth	브라우저 전체 너비
	Integer	outerHeight	브라우저 전체 높이
	Integer	offsetX	부모 항목과의 상대 X 좌표
	Integer	offsetY	부모 항목과의 상대 Y 좌표
	Integer	offsetWidth	보여지는 BODY의 너비(테두리포함)
	Integer	offsetHeight	보여지는 BODY의 높이(테두리포함)
	Integer	scrollX	스크롤된 X 좌표값
	Integer	scrollY	스크롤된 Y 좌표값
	Integer	scrollWidth	스크롤되는 전체 문서의 너비
	Integer	scrollHeight	스크롤되는 전체 문서의 높이
	Integer	clientX	테두리에서 본문까지의 X 좌표값
	Integer	clientY	테두리에서 본문까지의 Y 좌표값
	Integer	clientWidth	보여지는 BODY의 너비(테두리제외)
	Integer	clientHeight	보여지는 BODY의 높이(테두리제외)
	Integer	screenX	창의 X 좌표
	Integer	screenY	창의 Y 좌표
	Boolean	fontSmoothingEnabled	폰트 알리아싱 여부
	Integer	bufferDepth	단일 픽셀의 화소 비트수
	Integer	deviceXDPI	단말기 X DPI
	Integer	deviceYDPI	단말기 Y DPI
	Integer	logicalXDPI	논리 X DPI
	Integer	logicalYDPI	논리 Y DPI
	Integer	systemXDPI	시스템 X DPI
	Integer	systemYDPI	시스템 Y DPI
	Integer	updateInterval	화명 갱신 주기
	Integer	inchWidth	1 인치 너비 픽셀수
	Integer	inchHeight	1 인치 높이 픽셀수
	Float	screenWidth	화면 너비 인치수
	Float	screenHeight	화면 높이 인치수
	Float	screenDianol	화면 대각선 인치수

BrowserLocation		browserLocation ⁷³	웹 브라우저 위치정보
	Date	stamp	위치정보 시각
	Double	latitude	위도
	Double	longitude	경도
	Integer	accuracy	위치 정확도
	Double	altitude	고도
	Integer	altitudeAccuracy	고도 정확도
	Double	heading	방향
	Double	speed	속도
	String	errorMessage	미수집 원인
List<RequestHeader>		requestHeaders ⁷⁴	HTTP 헤더
	String	name	헤더 키
	String	value	헤더 값
List<PluginData>		plugins ⁷⁵	웹브라우저 플러그인 목록
	String	name	플러그인 명
	String	version	플러그인 버전
	String	description	플러그인 설명
	String	filename	플러그인 파일명
	List<MimeTypeData> >	mimetypes	해당 플러그인 지원 Mime Type 목록
List<MimeTypeData>		mimeTypes ⁷⁶	지원 Mime Type 목록
	String	type	Mime-Type
	String	suffixes	Mime-Type 확장자
	String	description	Mime-Type 설명
	Boolean	pluginEnabled	Mime-Type 활성화 여부
	PluginData	plugins	Mime-Type 지원 플러그인 목록
LocationData		location ⁷⁷	IP 기반 위치정보
	String	ipAddress	IP 주소
	Date	stamp	조회 시각
	Double	latitude	위도
	Double	longitude	경도

⁷³ window.navigator.geolocation 객체의 속성값에서 단말정보를 수집합니다. 이 값을 수집할 때 브라우저에서 위치정보를 수집한다는 경고창을 출력합니다. 또한 사용자 접속환경(유선랜, http 접속) 및 웹브라우저에 따라 수집되지 않을 수 있습니다.

⁷⁴ HTTP Header 에서 단말정보를 수집합니다.

⁷⁵ 웹 브라우저에 설치된 Plugin 정보를 수집합니다.

⁷⁶ 웹 브라우저에서 지원하는 Mime-Type 정보를 수집합니다.

⁷⁷ 수집된 Nat IP 를 기준으로 위치정보를 습득합니다. WAS 가 외부네트워크와 차단된 상태라면 수집할 수 없습니다.

	String	countryCode	국가코드
	String	countryName	국가명
	String	regionCode	지역코드
	String	regionName	지역명
	String	city	도시명
	String	metroCode	전철 코드
	String	areaCode	지역 코드
	String	timeZone	Time Zone
	String	continent	대륙
	String	postalCode	우편번호
	String	isp	ISP 정보
	String	org	ISP 조직
	String	domain	ISP 도메인
	String	asnum	ISP AS 번호
	String	netSpeed	인터넷 속도
	String	userType	사용자 구분
	Integer	accuracyRadius	정확도 범위
	String	countryConf	ISP 국가
	String	cityConf	ISP 도시
	String	regionConf	ISP 지역
	String	postalConf	ISP 우편번호
	String	error	오류정보
FlashData		flashData⁷⁸	플래시 정보
	String	language	실행되는 시스템 언어
	String	cpuArchitecture	CPU 구조
	String	os	운영체제 명
	Boolean	supports32BitProcesses	32 비트 프로세서 여부
	Boolean	supports64BitProcesses	64 비트 프로세서 여부
	String	touchscreenType	터치 스크린 종류
	Boolean	avHardwareDisable	오디오/비디오 하드웨어 비활성여부
	Boolean	hasAccessibility	접근성 지원여부
	Boolean	hasAudio	오디오 지원여부
	Boolean	hasAudioEncoder	오디오 인코더 지원여부
	Boolean	hasEmbeddedVideo	비디오 지원여부
	Boolean	hasIME	IME 지원여부
	Boolean	hasMP3	MP3 지원여부
	Boolean	hasPrinting	프린팅 지원여부
	Boolean	hasScreenBroadcast	화면 중계 지원여부
	Boolean	hasScreenPlayback	화면 플레이백 지원여부

⁷⁸ Adobe Flash 에서 단말정보를 수집합니다. 모바일 환경에서는 Flash 가 동작하지 않습니다.

	Boolean	hasStreamingAudio	오디오 스트리밍 지원여부
	Boolean	hasStreamingVideo	비디오 스트리밍 지원여부
	Boolean	hasTLS	TLS 지원여부
	Boolean	hasVideoEncoder	비디오 인코더 지원여부
	Boolean	isDebugger	디버그 모듈 여부
	Boolean	IsoStorageTest	로컬 저장소 지원여부
	Boolean	localFileReadDisable	로컬 파일 읽기 비활성여부
	String	manufacturer	플래시 제조사
	String	maxLevelIDC	클라이언트 하드웨어가 지원하는 최상의 H.264 IDC 항목
	Integer	pixelAspectRatio	화면 종횡비
	String	playerType	플레이어 종류
	String	screenColor	화면 색상수
	Integer	screenDPI	화면 DPI
	Integer	screenResolutionX	X 해상도
	Integer	screenResolutionY	Y 해상도
	String	version	플래시 버전
SilverlightData		silverlightData⁷⁹	실버라이트 정보
	String	osVersion	운영체제 버전
	Integer	processorCount	CPU 프로세서 코어수
	String	clrVersion	공용 언어 런타임 버전
	String	assemblyClrVersion	사용하는 .NET 버전
	Integer	sysUptimeMs	시스템 시각
	Boolean	isolatedStorageEnabled	임시저장소 활성화여부
	Boolean	isolatedStorageTest	임시저장소 사용가능 여부
AppletData		appletData⁸⁰	자바애플릿 정보
	String	javaVendor	자바 제조사
	String	javaVendorUrl	자바 제조사 URL
	String	javaVersion	자바 버전
	String	osArch	운영체제 구조
	String	osName	운영체제 명
	String	osVersion	운영체제 버전
	String	lanIpAddress	랜카드 할당 IP 주소

[표 2] PluginFreeDTO 객체 구조

⁷⁹ Microsoft Silverlight 에서 단말정보를 수집합니다. Microsoft 계열의 Internet Explorer 와 Edge 브라우저에서만 수집이 가능합니다.

⁸⁰ Java Applet 에서 단말정보를 수집합니다. 보안상 대부분의 웹 브라우저에서는 Java Applet 의 동작이 되지 않습니다. 실행할 수 있는 웹 브라우저의 경우에도 실행에 따른 경고창을 출력합니다.

1.3. 모바일 단말정보수집 수집항목

- PluginFreeMobileDTO 객체의 변수목록

구분	변수타입	변수명 ⁸¹	기본값	설명
모바일 단말정보	String	osMobile	null	모바일 OS 종류
	String	uuid	null	기기별 고유 식별번호
	String	osVersion	null	OS 버전
	String	osCodeName	null	OS 코드네임
	String	manufacturer	null	제조사
	String	model	null	모델명
	String	callState	null	음성통화상태
	String	dataState	null	데이터통신상태
	String	deviceId	null	기기 고유 번호 (IMEI)
	String	phoneNumber	null	단말기 전화번호
	String	netCountryCode	null	네트워크 국가코드
	String	simCountryCode	null	SIM 카드 국가코드
	String	netOpCode	null	네트워크 망 사업자 코드
	String	netOpCodeName	null	네트워크 망 사업자명
	String	simOpCode	null	SIM 카드 망사업자 코드
	String	simOpCodeName	null	SIM 카드 망사업자명
	String	networkType	null	네트워크 타입
	String	phoneType	null	단말기 종류
	String	simSerialNumber	null	SIM 카드 일련번호
	String	simState	null	SIM Card 상태
	String	subscriberId	null	가입자 ID
	String	connNetwork	null	연결된 네트워크 정보
	String	macAddress	null	MAC 주소
	String	isRooting	null	루팅/탈옥 여부
	String	ipNat	null	NAT IP
	String	ipEth0	null	ETH0 IP
	String	ipEth1	null	ETH1 IP
	String	gipEth0	null	ETH0 Gateway IP
	String	gipEth1	null	ETH1 Gateway IP
	String	osLocale	null	운영체제 언어팩 코드
기타	String	appVersion	null	앱 버전
	String	packageName	null	앱 패키지 이름
	String	logStamp	null	정보 수집 시작 시간
	String	sdkVersion	null	수집 SDK 빌드 번호 (버전)

⁸¹ 본 변수들의 값을 얻는 방법은 getter 메소드를 호출하여 얻어옵니다. 예를 들어 숫자(int)/문자열(String) 변수일 때 변수명이 model 라면 getModel() 함수를 호출하여 해당값을 얻어올 수 있습니다.

	String	flagSearchApp	null	특정 앱의 설치 및 실행 여부
	String	hash	null	해쉬
	String	hashYn	null	해쉬 일치여부
	int	elapsedTime	-1	수집소요시간

[표 3] PluginFreeMobileDTO 객체 구조

2. 별첨 – 단말기 수집정보 테이블 가이드

2.1. PC 단말기 수집정보 테이블⁸²

구분	컬럼명	컬럼타입	길이	설명
네트워크	ipNat	VARCHAR	15	NAT IP
	natErrCd	CHAR	1	NAT IP 획득 에러 코드
	natCntryCd	CHAR	2	NAT IP 국가코드
	ipEth0	VARCHAR	15	ETH0 IP
	ipEth1	VARCHAR	15	ETH1 IP
	ipEth2	VARCHAR	15	ETH2 IP
	gipEth0	VARCHAR	15	ETH0 GATEWAY IP
	gipEth1	VARCHAR	15	ETH1 GATEWAY IP
	gipEth2	VARCHAR	15	ETH2 GATEWAY IP
	macEth0	VARCHAR	17	ETH0 MAC
	macEth1	VARCHAR	17	ETH1 MAC
	macEth2	VARCHAR	17	ETH2 MAC
	macLogicalEth0	VARCHAR	17	ETH0 Logical MAC
	macLogicalEth1	VARCHAR	17	ETH1 Logical MAC
	macLogicalEth2	VARCHAR	17	ETH2 Logical MAC
	forgeryMac	CHAR	1	ETH MAC 변조여부
	forgeryMacEth0	CHAR	1	ETH0 MAC 변조여부
	forgeryMacEth1	CHAR	1	ETH1 MAC 변조여부
	forgeryMacEth2	CHAR	1	ETH2 MAC 변조여부
	prxyYn	CHAR	1	Proxy 사용 여부
	ipPrxyLcal	VARCHAR	15	Proxy 로컬설정 IP
	cntryPrxyLcal	CHAR	2	Proxy 로컬설정 IP 국가코드
	ipPrxy	CHAR	1	Proxy IP
	cntryPrxy	CHAR	2	Proxy IP 국가코드
	vpnYn	CHAR	1	VPN 사용 여부
	ipVpn	VARCHAR	15	VPN IP
	cntryVpn	CHAR	2	VPN IP 국가코드
	ipVpnLcal	VARCHAR	15	VPN 로컬설정
	cntryVpnLcal	CHAR	2	VPN 로컬설정 국가코드
	ipRmt	VARCHAR	15	원격지 IP
	cntryRmt	CHAR	2	원격지 IP 국가코드
	ipEtc	VARCHAR	15	예비 IP
소프트웨어	osCd	CHAR	2	운영체제 코드

⁸² 컬럼타입과 길이에 – 로 표시된 항목은 멀티 브라우저/멀티운영체제를 지원하지 않는 버전의 호환을 위한 컬럼으로 수집 및 저장을 추천하지 않습니다.

	osLocale	VARCHAR	14	운영체제 언어팩 코드
	osGuid	VARCHAR	40	운영체제 식별정보
	osRayn	CHAR	1	운영체제 원격접속허용여부
	osSpvsn	-	-	운영체제 서비스팩 코드
	osVsnCd	CHAR	13	운영체제버전
	osFirewallCd	CHAR	2	운영체제 방화벽 설정 코드
	osTypeCd	-	-	운영체제 구분 코드
	osPatchsts	-	-	운영체제 패치 상태
	bootUuid	VARCHAR	50	운영체제 부트 UUID
	bwVsnCd	CHAR	2	브라우저 버전
	ieVsn	-	-	브라우저 버전코드
	ieDetvsn	-	-	브라우저 상세버전
	iePrdctid	-	-	브라우저 PRODUCT ID
하드웨어	hddMdel	VARCHAR	50	HDD 모델
	hddSrial	VARCHAR	50	HDD 시리얼
	hddSrialAll	VARCHAR[]	50	HDD 시리얼(전체)
	usbSrial1	VARCHAR	50	USB 시리얼 1
	usbNpkiYn1	CHAR	1	USB NPki 폴더 여부
	usbSrial2	VARCHAR	50	USB 시리얼 2
	usbNpkiYn2	CHAR	1	USB NPki 폴더 여부
	usbSrial3	VARCHAR	50	USB 시리얼 3
	usbNpkiYn3	CHAR	1	USB NPki 폴더 여부
	mbSrial	VARCHAR	50	마더보드 시리얼
	mbMnft	VARCHAR	50	마더보드 제조업체
	mbPrdct	VARCHAR	50	마더보드 PRODUCT
	cpuPrcesnm	VARCHAR	50	CPU 프로세스명
	cpuldnr	VARCHAR	50	CPU 식별자
	cpuld	VARCHAR	50	CPU ID
	keyboardType	CHAR	2	키보드 타입 코드
보안	stsDhack	CHAR	2	해킹차단기 상태 코드
	stsKey	CHAR	2	키보드보안 상태 코드
	remoteDesktop	CHAR	1	원격데스크톱 접속여부
	remoteDesktopCode	CHAR	2	원격데스크톱 구분코드
	virtualOs	CHAR	1	가상운영체제 접속여부
	virtualOsCode	CHAR	2	가상운영체제 구분코드
	certInfo	VARCHAR[]	50	저장된 인증서 목록
기타	elapsedTime	NUMBER	3	로그수집소요시간
	totalHash	-	-	Total 해쉬(디버그모드일 때만 수집)
	totalHashYn	CHAR	1	Total 해쉬 일치 여부
	efdsVersion	VARCHAR	30	단말정보수집모듈 버전
	logYn	CHAR	1	동작 여부(자바스크립트)
	executeYn	CHAR	1	정상수집 여부(서버키발급비교)

	httpReferer	VARCHAR	50	이전페이지 정보
--	-------------	---------	----	----------

2.2. 모바일 단말기 수집정보 테이블

구분	컬럼명	컬럼타입	길이	설명
모바일 단말정보	osMobile	CHAR	2	모바일 OS 종류
	uuid	VARCHAR	40	기기별 고유 식별번호
	osVersion	VARCHAR	10	OS 버전
	osCodeName	VARCHAR	20	OS 코드네임
	manufacturer	VARCHAR	20	제조사
	model	VARCHAR	20	모델명
	callState	CHAR	2	음성통화상태
	dataState	CHAR	2	데이터통신상태
	deviceId	VARCHAR	40	기기 고유 번호 (IMEI)
	phoneNumber	VARCHAR	20	단말기 전화번호
	netCountryCode	VARCHAR	20	네트워크 국가코드
	simCountryCode	VARCHAR	20	SIM 카드 국가코드
	netOpCode	VARCHAR	20	네트워크 망 사업자 코드
	netOpCodeName	VARCHAR	20	네트워크 망 사업자명
	simOpCode	VARCHAR	20	SIM 카드 망사업자 코드
	simOpCodeName	VARCHAR	20	SIM 카드 망사업자명
	networkType	CHAR	2	네트워크 타입
	phoneType	CHAR	2	단말기 종류
	simSerialNumber	VARCHAR	40	SIM 카드 일련번호
	simState	CHAR	2	SIM Card 상태
	subscriberId	VARCHAR	15	가입자 ID
	connNetwork	CHAR	2	연결된 네트워크 정보
	macAddress	CHAR	17	MAC 주소
	isRooting	CHAR	1	루팅/탈옥 여부
	String	VARCHAR	15	NAT IP
	String	VARCHAR	15	ETH0 IP
	String	VARCHAR	15	ETH1 IP
	String	VARCHAR	15	ETH0 Gateway IP
	String	VARCHAR	15	ETH1 Gateway IP
	String	VARCHAR	14	운영체제 언어팩 코드
기타	appVersion	CHAR	6	앱 버전
	packageName	VARCHAR	20	앱 패키지 이름
	logStamp	CHAR	19	정보 수집 시작 시간
	sdkVersion	CHAR	9	수집 SDK 빌드 번호 (버전)
	flagSearchApp	CHAR	2	특정 앱의 설치 및 실행 여부
	hash	CHAR	64	해쉬

	hashYn	CHAR	1	해쉬 일치여부
	elapsedTime	NUMBER	3	수집소요시간

3. 별첨 – PC 지정서비스 절차 가이드

3.1. PC 지정서비스 절차

3.1.1. 보안강화 내용고지



3.1.2. PC 지정서비스 안내

서비스안내	서비스신청	신청내역조회	서비스해지
• 이용PC지정서비스는 이용 등록된 PC에서만 이체가 가능한 보안서비스로, 이용PC가 지정되면 이용PC가 지정되지 않은 PC에서는 조회서비스만 가능합니다. ● 서비스개요 - 대상고객 - 개인인터넷뱅킹 가입고객(본 서비스 가입은 고객 선택 사항) - 신청방법 - 인터넷신청 : 공인인증서 로그인 후 서비스신청 - 서비스내용 - 등록하고자 하는 PC에서 인터넷뱅킹 로그인 후 전화(ARS비용 신청자부담)인증을 거쳐 서비스신청 - 등록대상 PC는 최대 10개까지 가능 - 등록된 PC에서만 등록PC 신청내역 조회 가능 - 서비스 등록PC : 조회 및 이체서비스 - 서비스 미등록PC : 조회서비스 - 서비스 해지 - 등록된 PC에서만 등록된 PC지정 해지가 가능 - 일괄PC 이용해지 및 최종 1대 PC의 이용해지는 전화(ARS)인증 후 해지 - 서비스 이용료 : 무료			

3.1.3. PC 지정 등록

● 등록 1 단계 - PC 별명설정 및 PC 정보 수집

서비스안내	서비스신청	신청내역조회	서비스해지
-------	-------	--------	-------

1 정보입력
2
3

* 이용PC지정서비스는 이용 등록된 PC에서만 이체가 가능한 보안서비스로, 이용PC가 지정되면 이용PC가 지정되지 않은 PC에서는 조회서비스만 가능합니다.
 * PC별명 설정은 15자 이내에서 영문/한글로 지정할 수 있습니다.
 * 다른 PC를 추가등록하시려면, 해당 PC에서 이용PC지정등록을 하셔야 하며, 최대 10개까지 등록 가능합니다.

PC별명 설정하기

PC별명	회사PC	중복확인
------	------	------

▶ PC별명 예제 : 집PC, 회사PC, 여의도사무실, My_PC 등

● 등록 1 단계 - PC 별명 중복확인(옵션)

서비스안내	서비스신청	신청내역조회	서비스해지
-------	-------	--------	-------

1 정보입력
2
3

* 이용PC지정서비스는 이용 등록된 PC에서만 이체가 가능한 보안서비스로, 이용PC가 지정되면 이용PC가 지정되지 않은 PC에서는 조회서비스만 가능합니다.
 * PC별명 설정은 15자 이내에서 영문/한글로 지정할 수 있습니다.
 * 다른 PC를 추가등록하시려면, 해당 PC에서 이용PC지정등록을 하셔야 하며, 최대 10개까지 등록 가능합니다.

PC별명 설정하기

PC별명	회사PC	중복확인	* 사용 가능한 PC별명 입니다.
------	------	------	--------------------

▶ PC별명 예제 : 집PC, 회사PC, 여의도사무실, My_PC 등

● 등록 2 단계 - 보안카드/OTP 인증

● 전화(ARS)인증 휴대폰번호 확인

승인전화번호	[010-3270-2500] * 은행에 등록된 휴대폰 번호를 통해 최종 승인완료후 서비스가 실행됩니다. * 은행에 등록된 휴대폰 번호가 변경된 경우 영업점 및 국민은행 홈페이지(www.kbstar.com)에서 변경 후 재신청하시기 바랍니다.
--------	--

[고객정보수정 바로가기](#)

보안매체 비밀번호 입력

☐ 마우스로 입력

●● [32] 앞의 두자리 ●● [08] 뒤의 두자리	KB 국민은행 Number. 0123456789 <table border="1"> <tr><td>1</td><td>●●●●</td><td>2</td><td>●●●●</td><td>3</td><td>●●●●</td><td>4</td><td>●●●●</td><td>5</td><td>●●●●</td></tr> <tr><td>6</td><td>●●●●</td><td>7</td><td>●●●●</td><td>8</td><td>●●</td><td>9</td><td>●●●●</td><td>10</td><td>●●●●</td></tr> <tr><td>11</td><td>●●●●</td><td>12</td><td>●●●●</td><td>13</td><td>●●●●</td><td>14</td><td>●●●●</td><td>15</td><td>●●●●</td></tr> <tr><td>16</td><td>●●●●</td><td>17</td><td>●●●●</td><td>18</td><td>●●●●</td><td>19</td><td>●●●●</td><td>20</td><td>●●●●</td></tr> <tr><td>21</td><td>●●●●</td><td>22</td><td>●●●●</td><td>23</td><td>●●●●</td><td>24</td><td>●●●●</td><td>25</td><td>●●●●</td></tr> <tr><td>26</td><td>●●●●</td><td>27</td><td>●●●●</td><td>28</td><td>●●●●</td><td>29</td><td>●●●●</td><td>30</td><td>●●●●</td></tr> <tr><td>31</td><td>●●●●</td><td>32</td><td>●●</td><td>33</td><td>●●●●</td><td>34</td><td>●●●●</td><td>35</td><td>●●●●</td></tr> </table>	1	●●●●	2	●●●●	3	●●●●	4	●●●●	5	●●●●	6	●●●●	7	●●●●	8	●●	9	●●●●	10	●●●●	11	●●●●	12	●●●●	13	●●●●	14	●●●●	15	●●●●	16	●●●●	17	●●●●	18	●●●●	19	●●●●	20	●●●●	21	●●●●	22	●●●●	23	●●●●	24	●●●●	25	●●●●	26	●●●●	27	●●●●	28	●●●●	29	●●●●	30	●●●●	31	●●●●	32	●●	33	●●●●	34	●●●●	35	●●●●
1	●●●●	2	●●●●	3	●●●●	4	●●●●	5	●●●●																																																														
6	●●●●	7	●●●●	8	●●	9	●●●●	10	●●●●																																																														
11	●●●●	12	●●●●	13	●●●●	14	●●●●	15	●●●●																																																														
16	●●●●	17	●●●●	18	●●●●	19	●●●●	20	●●●●																																																														
21	●●●●	22	●●●●	23	●●●●	24	●●●●	25	●●●●																																																														
26	●●●●	27	●●●●	28	●●●●	29	●●●●	30	●●●●																																																														
31	●●●●	32	●●	33	●●●●	34	●●●●	35	●●●●																																																														

> 주의안내 : 고객님의 소지하신 보안카드는 발급된 지 3년 이상된 보안카드 입니다.
 > 보다 안전한 거래를 위해 영업점에서 교체 후 이용하시기 바랍니다.
 > 교체 전 보다 안전한 거래를 위해 '이용PC지정서비스' 또는 '인터넷전화승인서비스' 가입 후 이용하시기 바랍니다.

이 카드는 타인에게 알도 대정 할 수 없으며 본인의 관리소홀로 발생한 손해는 은행이 책임지지 않습니다.
 도한 이카드를 분실 또는 타인에게 노출시킨 경우 즉시 신고해 주시기 바랍니다.
 전화 1588-9999

● 등록 3 단계 - 2 채널인증

서비스안내	서비스신청	신청내역조회	서비스해지
-------	-------	--------	-------

1 2 **3 등록완료**

• 등록된 휴대폰번호 [010-3270-2500]로 전화(ARS)가 발신되었습니다. 전화안내에 따라 이용 PC등록여부를 최종 결정하여 주시기 바랍니다.
 • 전화인증 완료 후 재로그인해야 PC지정이 완료됩니다.
 • 다른 PC를 추가등록하시려면, 해당 PC에서 이용PC지정등록을 하셔야 하며, 최대 10대까지 등록 가능합니다.



등록된 전화번호 [010-3270-2500]로 전화(ARS)가 발신되었습니다.
 전화(ARS) 인증후, 신청내역조회 메뉴를 통해 결과를 확인하시기 바랍니다.

3.1.4. PC 지정 삭제

● PC 지정 삭제 1 단계 - 삭제할 PC 선택

서비스안내	서비스신청	신청내역조회	서비스해지
-------	-------	--------	-------

• 서비스해지는 선택적 또는 일괄로 해지 가능하며, 일괄 해지 및 최종 1대 PC의 이용해지는 전화(ARS)인증 후 가능합니다.
 • 이용PC 전체를 해지할 경우 이용PC를 제한없이 사용가능하므로, 가급적 이용PC를 지정하여 사용하시기 바랍니다.

서비스신청내역

☐	등록일시	PC별명	상태
☐	2012/05/16 11:31:29	회사PC	현재PC

확인

● PC 지정 삭제 1 단계 - 삭제할 PC 삭제확인

서비스안내	서비스신청	신청내역조회	서비스해지
-------	-------	--------	-------

• 서비스해지는 선택적 또는 일괄로 해지 가능하며, 일괄 해지 및 최종 1대 PC의 이용해지는 전화(ARS)인증 후 가능합니다.
 • 이용PC 전체를 해지할 경우 이용PC를 제한없이 사용가능하므로, 가급적 이용PC를 지정하여 사용하시기 바랍니다.

서비스신청내역

☒	등록일시	PC별명	상태
☒	2012/05/16 11:31:29	회사PC	현재PC

확인

웹 페이지의 메시지


 선택항목을 서비스해지 하겠습니까?

● PC 지정 삭제 2 단계 – OTP/보안카드 인증

서비스안내	서비스신청	신청내역조회	서비스해지
-------	-------	--------	-------

● 전화(ARS)인증 휴대폰번호 확인

승인전화번호 [010-3270-2500]

* 승인전화번호를 통해 최종 승인완료 후 서비스가 실행됩니다.(통화로 무료)

* 은행에 등록된 휴대폰 번호가 변경된 경우 콜센터(1588-9999)를 통해 전화번호를 변경하시기 바랍니다.

보안매체 비밀번호 입력

☐ 마우스로 입력

●● [30] 앞의 두자리

●● [06] 뒤의 두자리

> 주의안내 : 고객님의 소지하신 보안카드는 발급된 지 3년 이상된 보안카드입니다.

> 보다 안전한 거래를 위해 영업점에서 교체 후 이용하십시오.

KB 국민은행 Number. 0123456789									
1	●●●●	2	●●●●	3	●●●●	4	●●●●	5	●●●●
6	●●	7	●●●●	8	●●●●	9	●●●●	10	●●●●
11	●●●●	12	●●●●	13	●●●●	14	●●●●	15	●●●●
16	●●●●	17	●●●●	18	●●●●	19	●●●●	20	●●●●
21	●●●●	22	●●●●	23	●●●●	24	●●●●	25	●●●●
26	●●●●	27	●●●●	28	●●●●	29	●●●●	30	●●
31	●●●●	32	●●●●	33	●●●●	34	●●●●	35	●●●●

이 카드는 타인에게 알도 대영 할 수 있으며 본인 외의 관리소홀로 발생한 손해는 은행이 책임지지 않습니다.

● PC 지정 삭제 3 단계 – 2 채널인증

1 2 3 등록완료

● 등록된 휴대폰번호 [010-3270-2500]로 전화(ARS)가 발신되었습니다. 전화안내에 따라 이용 PC등록여부를 최종 결정하여 주시기 바랍니다.

● 전화인증 완료 후 재로그인해야 PC지정이 완료됩니다.

● 다른 PC를 추가등록하시려면, 해당 PC에서 이용PC지정등록을 하셔야 하며, 최대 10대까지 등록 가능합니다.

등록된 전화번호 [010-3270-2500]로 전화(ARS)가 발신되었습니다.
전화(ARS) 인증후, 신청내역조회 메뉴를 통해 결과를 확인하시기 바랍니다.

3.1.5. PC 지정 등록 PC 목록

서비스안내	서비스신청	신청내역조회	서비스해지
-------	-------	--------	-------

● 고객님의 신청하신 이용PC등록 내역입니다.

● 다른 PC를 추가등록하시려면, 해당PC에서 이용PC지정등록을 하셔야 하며, 최대 10개까지 등록가능합니다.

● 이용PC 해지는 서비스 해지 메뉴에서 가능하며, 최종 1대 PC의 이용해지는 전화(ARS)인증 절차가 추가됩니다.

PC등록내역

번호	등록일시	해지일시	PC별명	전화(ARS)인증 휴대폰번호	상태
1	2012/05/16 11:31:29		회사PC	010-3270-****	현재PC

확인