본인확인서비스 이용기관 취약점 자체점검 체크리스트

기 관 명					
담 당 자		연 락 처		이 메 일	
작성일자	2024. 00. 00.				

순번	본인확인서비스 이용기관(웹사이트) 취약점 자체점검 항목	검토 여부
1	**(불필요한 중요정보 평문 노출)** 본인확인 이후 회원가입 단계에서 이용자에게 불필요한 개인정보(CI/DI)가 평문으로 드러나지 않도록 조치하였는가?	Y/N □□
2	**(파라미터 변조)** 본인확인 이후 회원가입 과정에서 이용자가 입력한 데이터를 서버 또는 Web to Web으로 전송할 때, 본인확인 결과정보(이름, 생년월일 등)를 다른 정보로 변조할 수 없도록 조치하였는가?	Y/N □□
3	**(입력정보 일치여부)** 본인확인기관(또는 대행사)으로부터 수신한 결과정보를 복호화한 값과 이용자가 입력한 값(회원가입 등) 간 일치 여부를 검증하였는가?	Y/N □□
4	**(복합인증 교차검증)** 신원확인 단계(실명확인, 본인확인, 계좌점유, 신분증 진위여부 등)에서 각각의 입력·응답 값의 일치 여부를 검증하였는가?	Y/N □□
5	**(인증결과 우회)** 본인확인 결과를 처리하는 과정에서 인증 실패시, 실패코드를 변조 또는 우회할 수 없도록 조치하였는가?	Y/N □□
6	**(데이터 재사용)** 동일 웹사이트에서 과거에 수집된 인증정보(암호화 데이터, 거래번호, 토큰, 세션 등)를 재사용하지 못하도록 조치하였는가?	Y/N □□
7	**(암호키 노출)** 본인확인서비스 테스트를 위한 샘플페이지 내 인증모듈 복호화 키와 실제 키가 동일하지 않도록 설정하고 해당 암호키가 노출되지 않도록 조치하였는가?	Y/N □□
8	**(데이터 위·변조)** 본인확인 결과정보가 웹사이트가 수신하는 과정에서 데이터를 위·변조 할 수 없도록 조치하였는가?	Y/N □□
9	**(관리자 페이지 노출)** 유추하기 쉬운 URL로 관리자 페이지가 노출되지 않도록 조치하였는가?	Y/N □□
10	**(전송구간 암호화)** 서버와 클라이언트 간 통신 시 전송구간을 암호화하였는가?	Y/N □□
11	**(프로세스 검증 누락)** 인증이 필요한 웹 사이트의 중요 페이지(관리자 페이지, 회원변경 페이지 등)에 대한 접근통제를 수행하고 있는가?	Y/N □□

참 고 체크리스트 항목별 취약점 사례 및 조치방안

① **(불필요한 중요정보 평문 노출)** 본인확인 이후 회원가입 단계에서 이용자에게 불필요한 개인정보(CI/DI)가 평문으로 드러나지 않도록 조치하였는가?

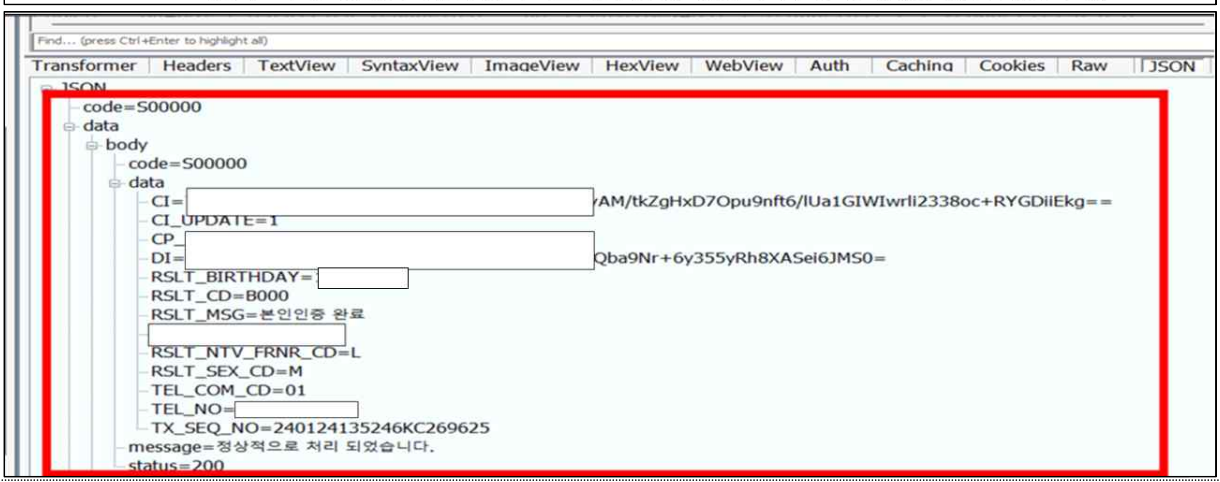
취약점 사례

○ 이용기관은 본인확인결과정보를 활용하기 위해 Hidden tag를 이용하여 필요한 데이터를 A페이지에서 B페이지로 전달하는 방법을 사용하고 있으나,

○ 회원가입에 필요한 정보뿐 만 아니라 이용자에게 노출될 필요가 없는 중요정보(CI/DI, CP코드 등)가 평문으로 노출되고 있는 상황.

○ 해당 데이터를 활용하여 부정가입 및 명의 도용 등 피해가 발생할 수 있음

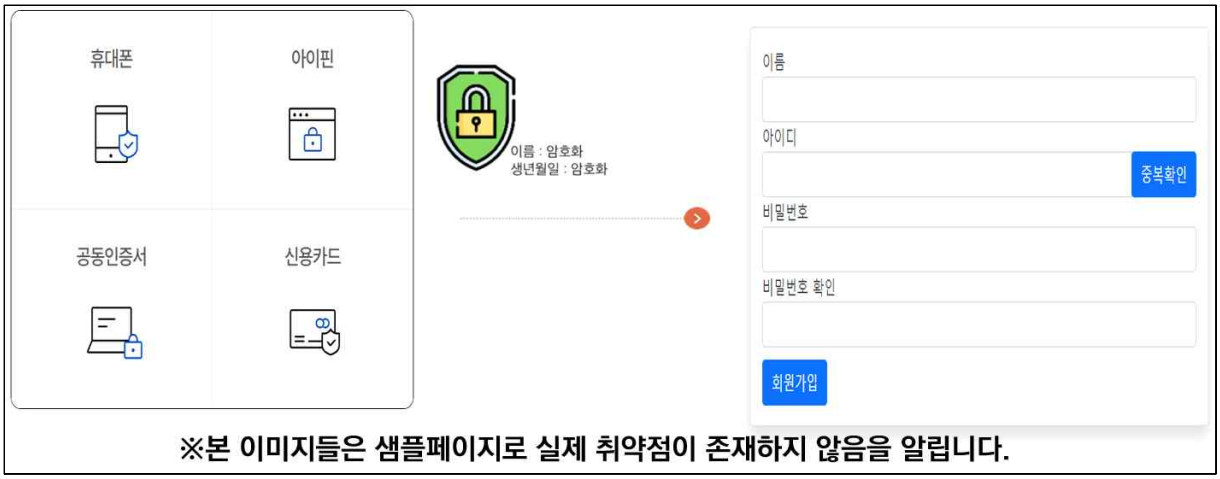
※본 이미지들은 샘플페이지로 실제 취약점이 존재하지 않음을 알립니다.



조치방안

○ 회원가입 단계에서 필요한 이용자의 개인정보는 암호 알고리즘을 사용하여 데이터 암호화 후 Hidden tag로 이용자 정보를 전달하여 정보노출 최소화

○ 본인확인 결과를 페이지에서 넘기지 않고 세션과 같은 이용자를 식별 가능한 임시 데이터를 활용하여 복호화된 본인확인 결과 데이터 처리



※본 이미지들은 샘플페이지로 실제 취약점이 존재하지 않음을 알립니다.

Body

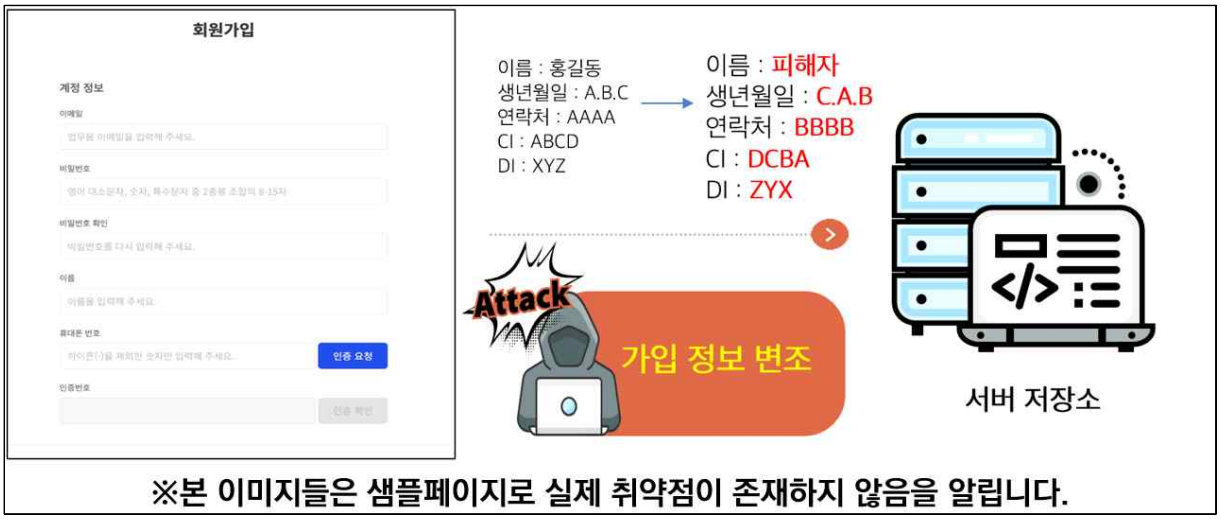
Name	Value
Keygb	1
Resultcd	0000
Mobilid	202401318354827
Mrchid	23120514
Ci	caef
Sex	f4b3
Foreigner	21e
Safeguard	
Cryptyn	Y
Mac	863
Commid	d48

② **(파라미터 변조)** 본인확인 이후 회원가입 과정에서 이용자가 입력한 데이터를 서버 또는 Web to Web으로 전송할 때, 본인확인 결과정보(이름, 생년월일 등)를 다른 정보로 변조할 수 없도록 조치하였는가?

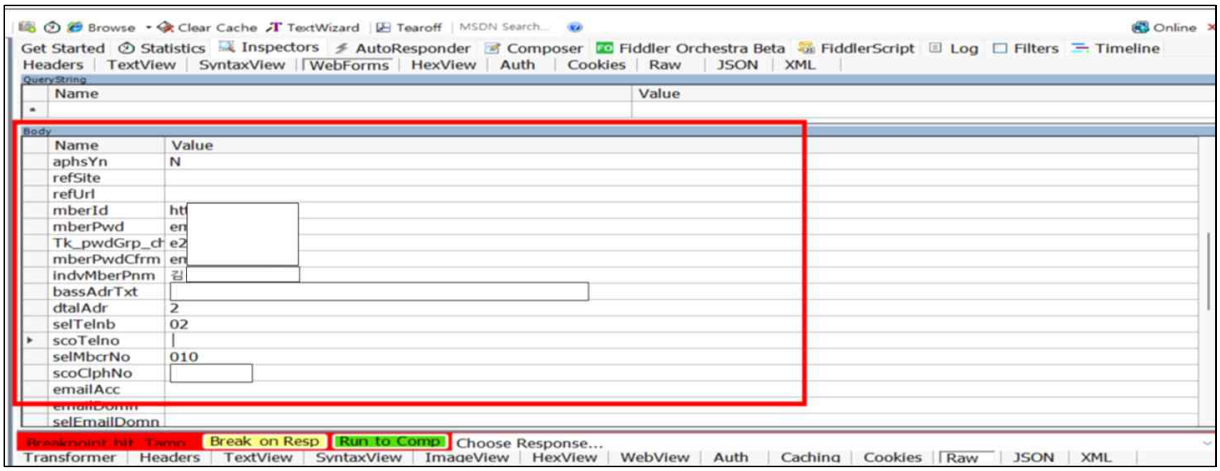
취약점 사례

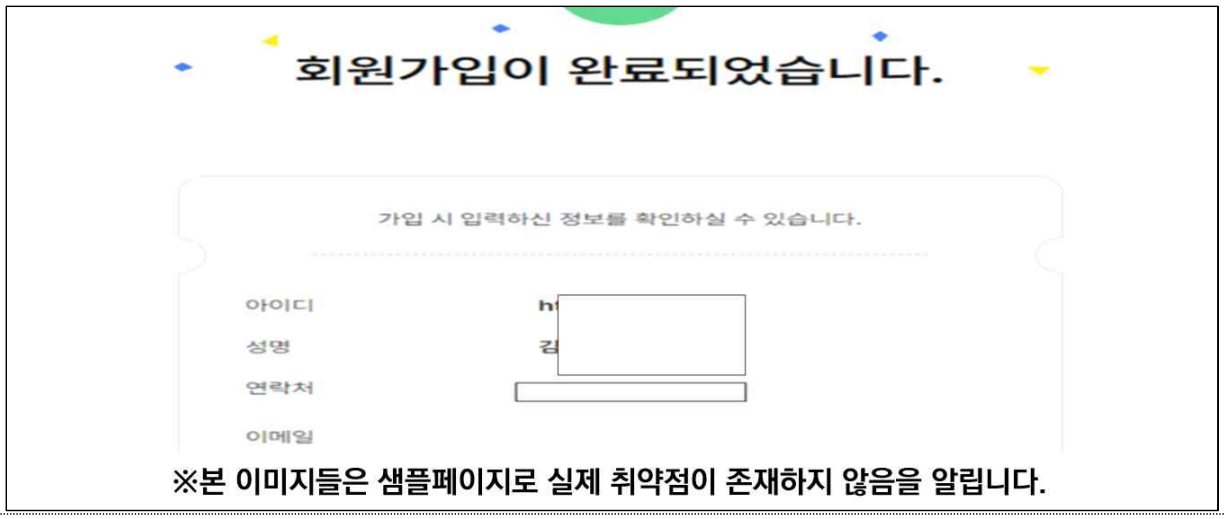
○ 이용자가 본인확인을 진행한 뒤 본인확인 결과정보를 웹페이지 이용기관의 서버로 전송할 때, 전송되는 패킷의 데이터 변조가 가능하여 본인확인 결과정보와 다른 정보가 서버에 전송될 경우, 타인명의로 가입이 가능

○ 이용자가 본인확인을 진행한 뒤 변경할 수 없는 Read-only 필드의 데이터를 포함하여 서버로 전송할 때, 해당 데이터들을 임의로 변조해 서버에 전송될 경우, 타인명의 가입이 가능



※본 이미지들은 샘플페이지로 실제 취약점이 존재하지 않음을 알립니다.





※본 이미지들은 샘플페이지로 실제 취약점이 존재하지 않음을 알립니다.

조치방안

○ 이용자가 회원가입 과정에서 변조하는 행위를 할 수 없도록 조치
  - 결과정보가 변조되지 않도록 난독화 같은 보호조치
  - 주요정보를 전송하지 않고 본인확인 결과에서 데이터를 가져오는 방식 등

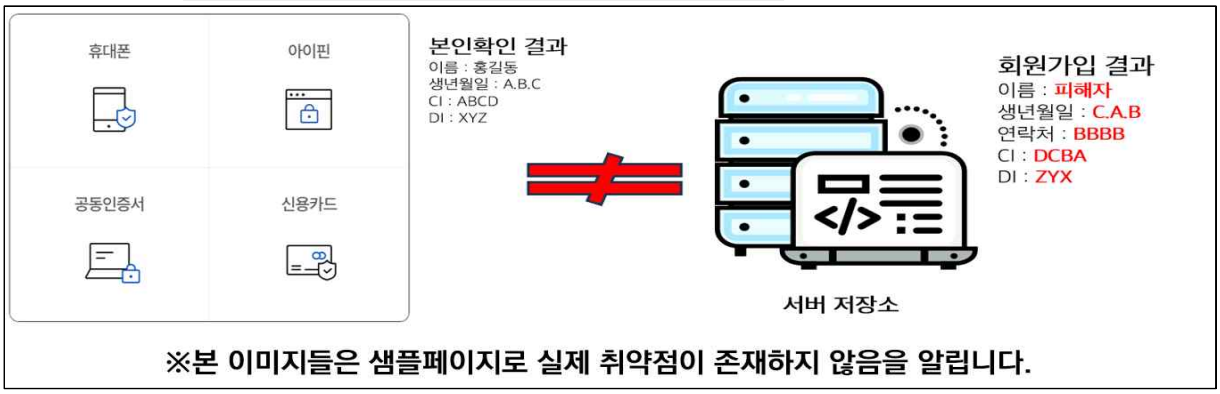
○ 이용자가 입력한 정보를 변조하지 못하도록 조치하고, 추가로 본인확인 결과와 서버로 전송된 데이터를 비교 검증하여 동일한 이용자임을 확인하도록 조치

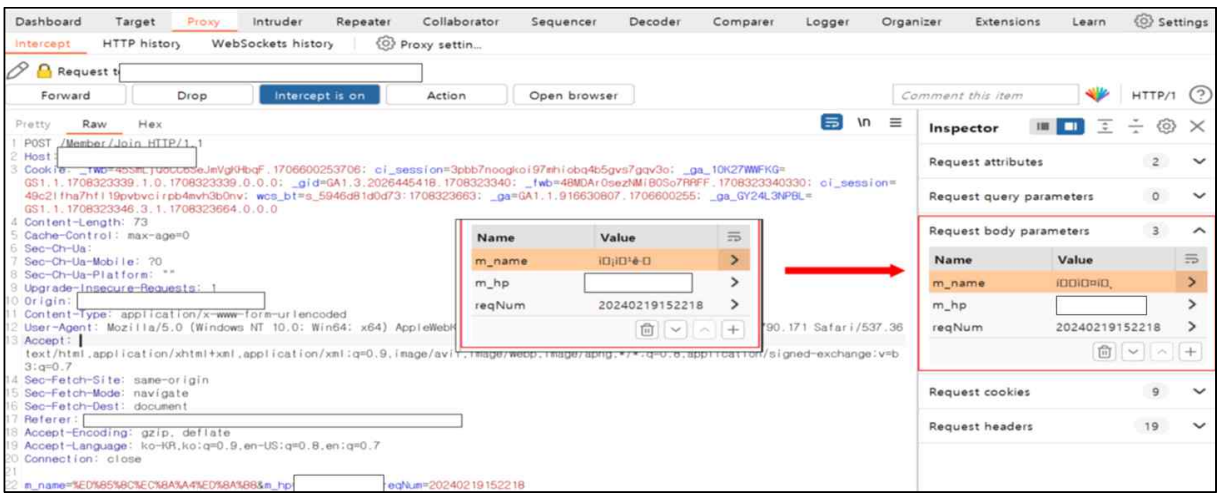
③ **(입력정보 일치여부)** 본인확인기관(또는 대행사)으로부터 수신한 결과 정보를 복호화한 값과 이용자가 입력한 값(회원가입 등) 간 일치 여부를 검증하였는가?

취약점 사례

○ 이용자가 본인확인 후, 본인확인 정보가 이용기관(웹사이트)에게 전달되어 복호화 후 DB에 저장되지만, 그 다음절차에서 이용자가 파라미터 변조 등으로 본인확인 결과와 상이한 정보를 입력해도 저장되는 경우,

○ 웹사이트에서 본인확인 결과와 DB에 입력한 정보간 동일인 여부를 검증해야 하나, 이를 검증하지 않아 명의 도용이 가능





조치방안

○ 이용자가 입력한 값과 인증결과가 동일한지 검증 필요

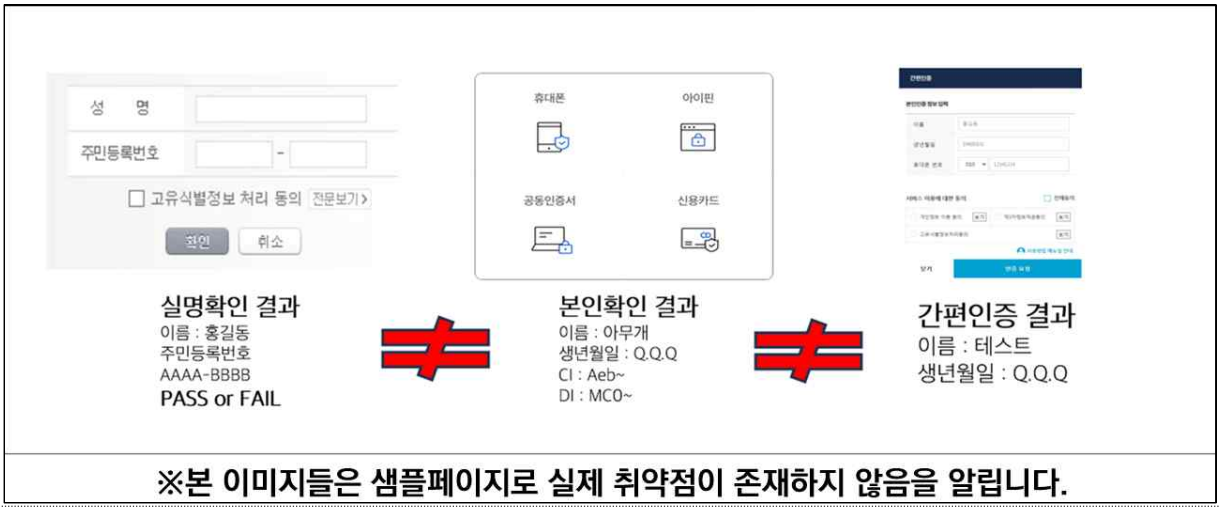
○ 본인확인 결과와 같은 신뢰 가능한 데이터를 기준으로 이용자가 입력한 데이터를 교차 비교하여 진위여부를 검증합니다. 이와같은 경우 이용자가 최초 입력한 정보와 해당 서비스의 마지막 단계 인증데이터와 비교를 진행하여 변조 여부를 서버에서 검증하도록 조치

④ **(복합인증 교차검증)** 신원확인 단계(실명확인, 본인확인, 계좌점유, 신분증 진위여부 등)에서 각각의 입력·응답 값의 일치 여부를 검증하였는가?

취약점 사례

○ 회원가입 단계 중 본인확인서비스 외 실명확인 방법이 추가로 있는 경우, 단순히 신원확인의 결과를 PASS&FAIL만 확인하고 각각의 결과값이 동일 이용자 인지 여부를 확인하지 않을 경우, 타인명의 가입이 가능

○ 서비스 이용을 위해 실명인증(주민등록번호)을 먼저 수행한 뒤 이후 단계에서 간편인증 혹은 본인확인을 수행하여 이용자를 식별하는 서비스에서 본인확인은 실명인증 결과와 비교를 하였지만 간편인증 결과와 실명인증 결과와 비교하지 않아 실명인증을 피해자 정보로 입력한 뒤 공격자의 간편인증을 수행하여 피해자의 명의로 서비스가 발급된 사례



※본 이미지들은 샘플페이지로 실제 취약점이 존재하지 않음을 알립니다.

조치방안

○ 실명확인 단계에서 발생하는 결과 값이 모두 동일한 이용자인지 확인을 위한 교차검증 수행

⑤ **(인증결과 우회)** 본인확인 결과를 처리하는 과정에서 인증 실패시, 실패코드를 변조 또는 우회할 수 없도록 조치하였는가?

취약점 사례

○ 본인확인 결과 '실패' 메시지를 '성공' 메시지로 위·변조하여 회원가입 할 경우 타인명의 가입이 가능

○ 이용자 조회, 비회원 조회와 같은 서비스를 본인확인 결과를 기반으로 제공하는 서비스에서 본인확인은 실패했지만 서버에서 PASS&FAIL만 검증하여 임의로 결과를 PASS로 변조하여 타인의 서비스 이용내역을 조회하는 사례

조치방안

○ 인증 결과에 대한 정보를 암호화하여 외부에서의 인증 결과에 대한 변조를 차단하고, 인증 결과를 검증할 수 있는 추가수단을 마련

⑥ **(데이터 재사용)** 동일 웹사이트에서 과거에 수집된 인증정보(암호화 데이터, 거래번호, 토큰, 세션 등)를 재사용하지 못하도록 조치하였는가?

취약점 사례

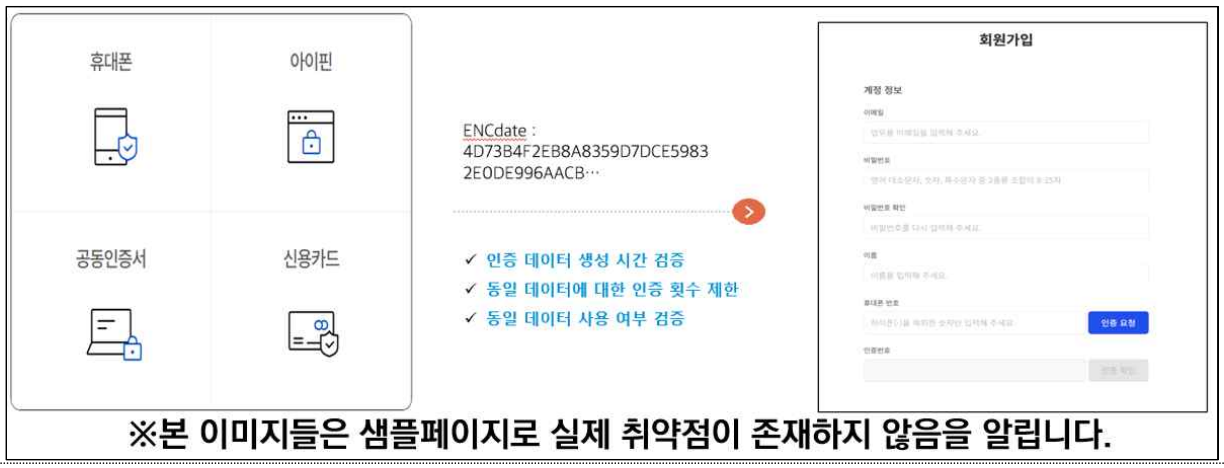
○ 공개된 공공 와이파이에 접근하여 이용자들의 본인확인 결과 데이터를 수집한 뒤 해당 데이터를 활용하여 이용자들이 사용하지 않은 서비스를 가입하거나 주문하여 본인확인 결과를 재사용 한 사례

○ 공공 와이파이 등 외부 환경에서 타인 이미 성공한 본인확인 결과 메시지를 탈취하여 동일 웹사이트에 재사용하여 타인이 회원가입 가능

※본 이미지들은 샘플페이지로 실제 취약점이 존재하지 않음을 알립니다.

조치방안

○ 데이터 재사용 방지를 위한 인증 횟수 및 만료 기간 설정 필요, 주기적인 암호키 갱신(통합모듈 표준규격 내 매 인증시 암호키 변경)



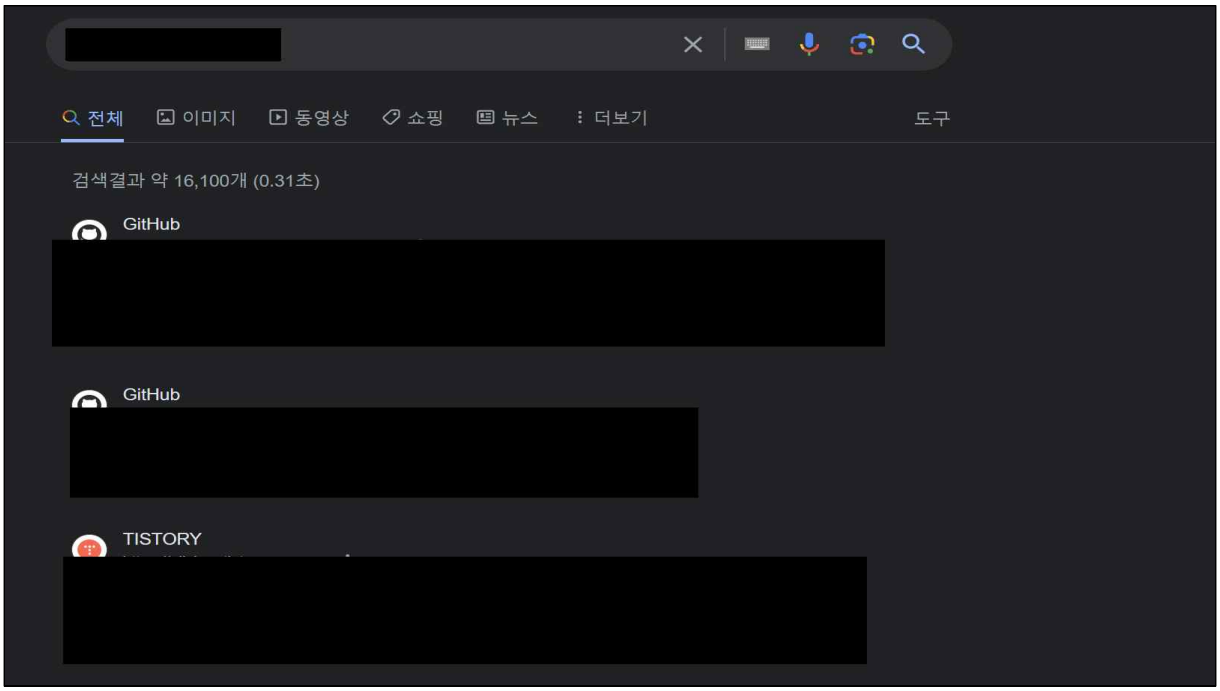
※본 이미지들은 샘플페이지로 실제 취약점이 존재하지 않음을 알립니다.

⑦ (**암호키 노출**) 본인확인서비스 테스트를 위한 샘플페이지 내 인증모듈 복호화 키와 실제 키가 동일하지 않도록 설정하고 해당 암호키가 노출되지 않도록 조치하였는가?

취약점 사례

○ 웹페이지 개발 및 테스트를 위해 만들어둔 임시 사이트가 남아있는 경우, 해당 사이트에 탑재된 본인확인서비스의 노출된 암호키를 호출하여 웹페이지와 이용자간 주고받는 본인확인 결과정보를 열람할 수 있으며, 이를 탈취하여 타인명의 회원가입 가능

○ 과거 본인확인서비스를 구축하는 개발자가 본인이 사용하는 모듈과 구축 방법을 자신의 블로그에 업로드하여 외부에 공개가 되어 해커는 해당 정보를 바탕으로 해당 페이지에서 사용하는 모듈의 암호 알고리즘과 키를 획득한 뒤 본인확인 결과데이터를 임의로 만들어 타인명의로 회원가입을 한 사례



조치방안

○ 고정된 키로 암호알고리즘을 사용할 경우 취약하여 주기적으로 암호키 갱신

○ 검색 엔진, 블로그, github 등과 같은 외부로 공개된 서비스에 본인확인 결과정보를 암/복호화 하는 키를 노출되지 않도록 설정하고 주기적으로 점검을 통해 확인

○ 크롤러가 웹 서버에 있는 중요 파일 경로에 접근하지 못하게 하도록 설정하거나 적절한 예외처리를 통하여 본인확인 관련 페이지 노출 차단

⑧ **(데이터 위·변조)** 본인확인 결과정보가 웹사이트가 수신하는 과정에서 데이터를 위·변조 할 수 없도록 조치하였는가?

취약점 사례

○ 본인확인이 완료된 후 결과정보가 이용자를 거쳐 웹페이지에 전달될 때, 이용자에게 불필요한 중요정보가 노출되어, 해당 CI/DI를 위·변조하여 회원가입 할 경우 타인명의 회원가입이 가능

조치방안

○ 이용기관별로 가맹점 고유 KEY를 등록하여 주기적으로 암호키 갱신

○ 주기적인 취약점 점검을 통한 외부에서 모듈에 대한 접근 및 노출에 대한 안전성 확보

○ 주요 본인확인 관련 데이터에 대한 접근 권한을 강화하여 인가된 이용자를 제외한 기타 이용자에 대한 접근 차단

⑨ **(관리자 페이지 노출)** 유추하기 쉬운 URL로 관리자 페이지가 노출되지 않도록 조치하였는가?

취약점 사례

○ 웹 관리자의 권한이 노출되어 웹사이트 변조 뿐 만 아니라 취약성 정도에 따라 웹 서버의 권한까지도 노출되어 홈페이지 내 개인정보 누출 가능성 존재

○ 쇼핑몰 사이트를 모두 관리 가능한 admin페이지를 유추하기 쉬운 경로인 ~/admin으로 설정되어 있는 사이트에서 해커가 해당 페이지에 접근하여 SQL Injection을 활용하여 관리자 권한으로 로그인 한 뒤 서버 DB에 접근하여 해당 사이트에 존재하는 모든 이용자의 개인정보를 탈취한 사례

조치방안

○ 일반 이용자의 접근이 불필요한 관리자 로그인 페이지 주소를 유추하기 어려운 이름으로 변경하고 관리자 페이지 접근 포트도 변경함

○ 관리자 페이지의 하위 페이지 URL을 직접 입력하여 접근하지 못하도록 페이지마다 세션 검증이 필요함

○ 관리자 페이지 이외에도 특정 이용자만 접근 가능한 페이지들은 정상적인 프로세스에 따라 접근할 수 있도록 페이지마다 세션 검증이 필요함

○ 웹 방화벽을 이용하여 특정 IP만 접근 가능할 수 있도록 룰셋 적용

⑩ **(전송구간 암호화)** 서버와 클라이언트 간 통신 시 전송구간을 암호화 하였는가?

취약점 사례

○ 웹상의 데이터 통신은 대부분 텍스트 기반으로 이루어지기 때문에 서버와 클라이어튼 간에 암호화 프로세스를 구현하지 않으면 회원가입 시 이용자의 중요정보가 간단한 도청(Sniffing)을 통해 탈취 및 도용될 수 있음

조치방안

○ 웹상에서의 전송 정보를 제한하여 불필요한 비밀번호, 주민등록번호, 계좌정보와 같은 중요정보의 전송을 최소화하여야 하며, 중요정보에 대해서는 반드시 SSL 등의 암호화 통신을 사용하여 도청으로부터의 위험을 제거함

○ 쿠키와 같이 클라이언트 측에서 노출되는 곳에 비밀번호, 인증인식 값, 개인정보 등의 정보를 기록하지 않음

○ 암호화 전송 시 프로토콜 설계의 결함이 있는 SSLv2, SSLv3, TLSv1.0, TLSv1.1은 비활성화 필수, TLSv1.2 이상 사용을 권장함

⑪ **(프로세스 검증 누락)** 인증이 필요한 웹 사이트의 중요 페이지(관리자 페이지, 회원변경 페이지 등)에 대한 접근통제를 수행하고 있는가?

취약점 사례

○ 인증이 필요한 웹 사이트의 중요페이지에 대한 접근 제어가 미흡할 경우 하위 URL 직접 접근, 스크립트 조작 등의 방법으로 중요한 페이지에 대한 접근이 가능하여 회원가입된 이용자의 개인정보 유출 가능성 존재

조치방안

○ 우회될 수 있는 플로우를 차단하여야 하며, 페이지별 권한 매트릭스를 작성하여 페이지에 부여된 권한의 타당성을 체크한 후 권한 매트릭스를 기준으로 전 페이지에서 권한 체크가 이뤄지도록 구현하여야 함

○ 인증이 필요한 모든 페이지에 대해 유효 세션임을 확인하는 프로세스 및 주요 정보 페이지에 접근 요청자의 권한 검증 로직을 적용함

○ 유효 세션의 검증 및 페이지에 대한 접근 권한을 Client Side Script에 의존할 경우 이용자가 임의로 수정할 수 있으므로 Server Side Script로 구현된 프로세스를 사용