

파일 완전 삭제 도구 개발

2020. 10. 23

지도교수 : 양환석 교수님

1 조

김성일
공종현
정성철
유민경
이준희
노인규



목 차

- ❖ 조원 편성
- ❖ 주제 선정
- ❖ 구 상 도
- ❖ 추진 경과
- ❖ 개발 환경 및 개발 내용
- ❖ 개발 시스템 운영
- ❖ 결론 및 기대 효과

조원 편성

이름	역할
김성일(팀장)	웹 프론트 개발, PPT 및 보고서 작성, 개발 총괄
노인규	웹 백 개발, DB 설계 및 구축
공종현	윈도우 프로그램 및 GUI 개발
정성철	안드로이드 어플리케이션 개발
유민경	안드로이드 어플리케이션 프론트 개발
이준희	DB 설계 및 구축

주제 선정(1/3)

컴퓨터 중고 하드디스크 개인정보 유출 통로?

입력 2013.06.18 (21:28) | 수정 2013.06.23 (16:11)

0 48

고화질

표준화질

자동재생 OFF



다가나가와현 납세 등 개인정보 유출... 하드디스크, 인터넷 경매로 팔려

등록 2019-12-06 14:51:56

납세기록 등 개인정보 담긴 하드디스크 9개 유출
유출된 하드디스크 용량만 27테라바이트

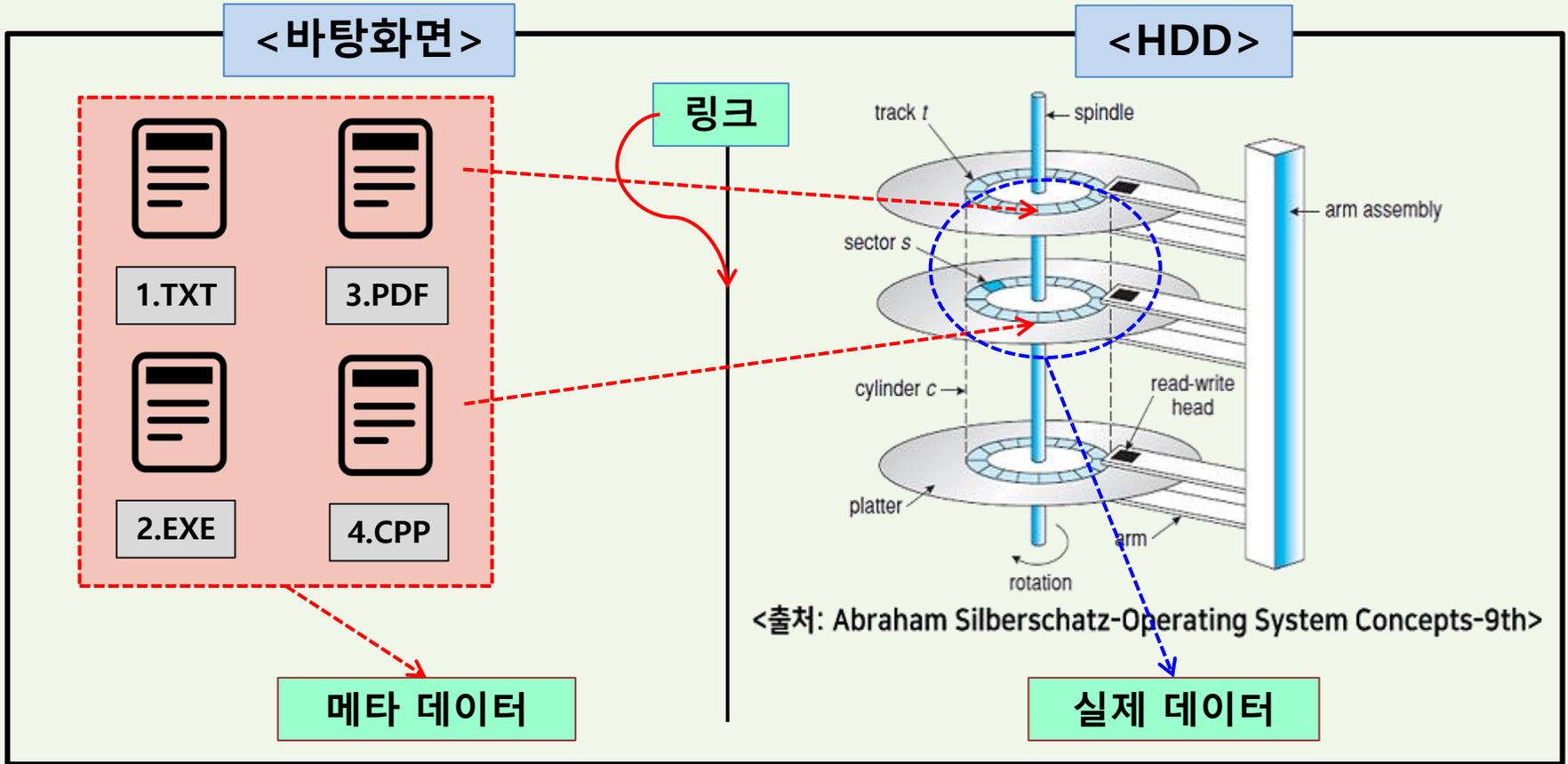
[서울=뉴시스] 김예진 기자 = 일본에서 가나가와(神奈川)현이 가지고 있던 납세 등에 대한 개인·비밀 정보가 대량 유출돼 파문이 되고 있다.

6일 아사히 신문은 가나가와 현청 서버에서 분리된 하드디스크(HDD)가 납세기록 등이 완전히 지워지지 않은 채 인터넷 경매에 팔려 중고품으로 나돌았다고 보도했다.

일반인과 기업의 개인정보 유출을 막기 위해 완전 삭제 프로그램을 개발

주제 선정 (2/3)

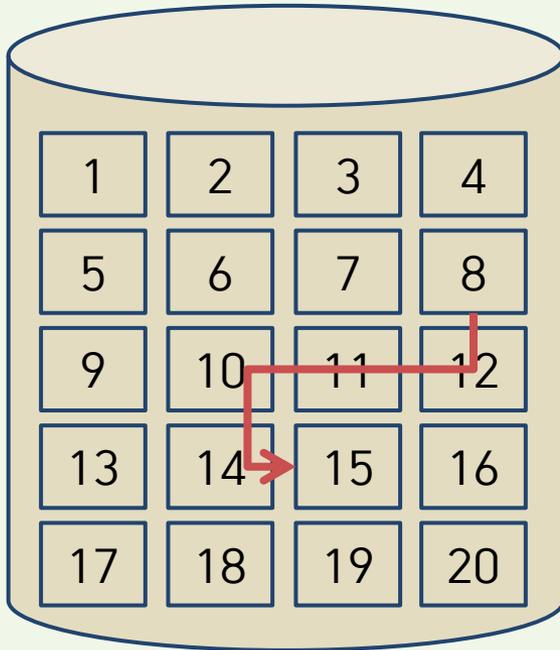
파일 구조



사용자가 생성한 파일이나 폴더를 삭제하면, 메타 데이터가 삭제되고 연결된 링크 또한 단절되나 실제 데이터는 삭제되지 않고 잔존 \Rightarrow 데이터의 복구가 가능

주제 선정 (3/3)

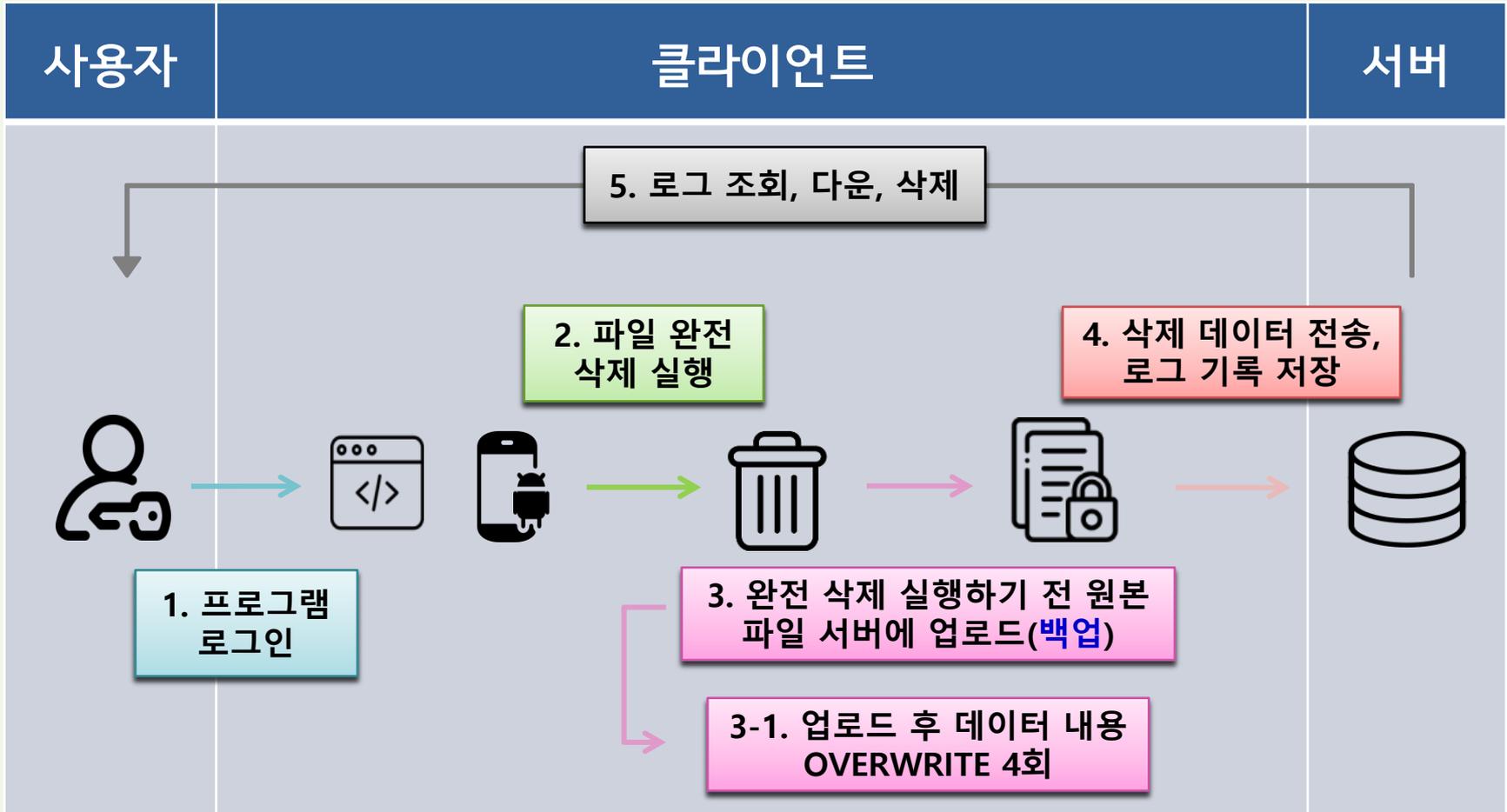
파일 저장 원리



Status	File Name	Start Block	Length
Create	1.txt	1	3
Create	2.exe	4	5
Create	3.pdf	9	2
Create	4.cpp	11	4
Delete	2.exe	-	-
Create	5.exe	Linked List 4	8

- ◆ Disk Space Allocation Algorithm에 의해 **연속할당방법**으로 빈 공간을 순차적으로 할당
- ◆ 중간에 파일을 삭제하면 빈 공간이 발생하고, 파일을 추가하면 중간의 빈 공간 부터 다시 채워지며, 파일의 길이가 길어지면 **연결할당방법**에 의해 공간을 할당

구상도



완전 삭제 실행전 원본 파일을 DB에 30일간 저장, 오삭제로 인한 복구에 대비

추진 경과

수행 업무	추진기간 (2020년)								
	3월	4월	5월	6월	7월	8월	9월	10월	
자료 조사 및 연구	■								
서버 구축		■							
실행파일 개발		■							
어플리케이션 개발		■							
웹 페이지 제작						■			
오류 수정 및 보완							■		

개발 환경 및 개발 내용 (1/10)

개발 환경

실행파일

EXE : C#

Android Application

APK : Java, XML
(Android Studio)

OS

Server : Heroku
Client : Windows 10

Web

Front : Javascript Angular
Back : Node.js

Database

Mongo DB

개발 총괄

총괄

○ 서버 구축

- Heroku(무료 호스팅서버) 와 MongoDB를 통한 서버 구축

○ 어플 및 프로그램 개발

- PC 및 Android 용 “완전 삭제” 프로그램, 어플리케이션 개발

※ FOP(Four Overwrite Pass) 알고리즘 구현

○ 웹 페이지 제작

- Angular를 통한 프로그램 다운로드 및 삭제 로그 확인 가능 웹페이지 제작

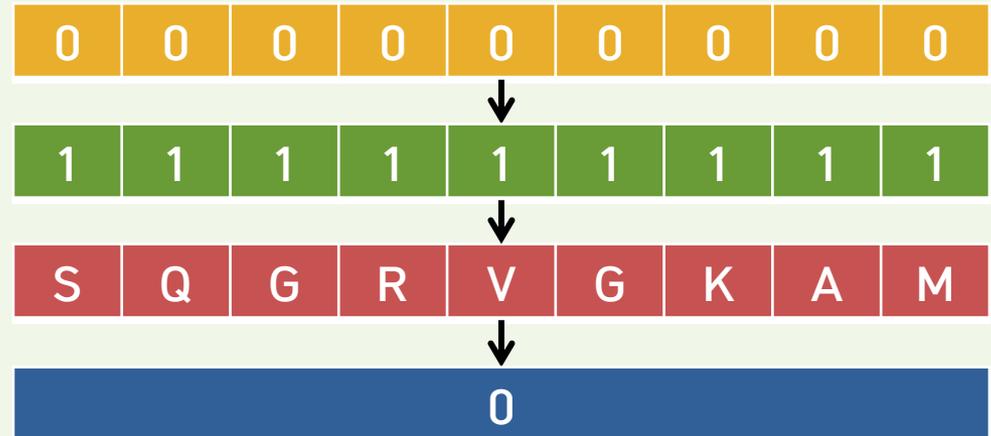
개발 환경 및 개발 내용 (3/10)

FOP 알고리즘

FOP Algorithm

- 국정원, 개인정보삭제 3 Pass 권장
- 삭제 데이터는 총 4회 Overwrite
 - 1. 0(파일길이)
 - 2. 1(파일길이)
 - 3. Random Byte(파일길이)
 - 4. 0(1번)
- 기존 Single Pass Algorithm 대비 데이터 복원율이 현저하게 감소

Erase Algorithm	내용	보안레벨
DoD 5220.22-M-ECE(3 Pass)	미국 국방성 문서에 3Pass 알고리즘을 명시해 놓았습니다. 전체섹터에 0을 기록하고 다시 1을 기록하고 임의의 문자로 채운 후 검증합니다. 대한민국 국가 정보원에서는 이 방식으로 개인정보를 삭제함을 권장합니다.	Midium



PC/Andorid 버전의 새로운 FOP Algorithm (**Four Overwrite Pass**) 구현

개발 환경 및 개발 내용 (4/10)

안드로이드 - 파일선택

```
public void request(){
    EditText edit_id = findViewById(R.id.id);
    EditText edit_pw = findViewById(R.id.pw);
    String url = "https://eraser2020.herokuapp.com/users/register";
    final JSONObject testjson = new JSONObject();
    final try {
        testjson.put("id", edit_id.getText().toString());
        testjson.put("password", edit_pw.getText().toString());
        String jsonString = testjson.toString();
        Log.e("tag", "jsonString: " + jsonString);
        //이제 전송해볼까?
        final RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
        final JSONObject requestBody = new JSONObject();
        try {
            Log.e("tag", "requestBody: " + requestBody.toString());
            Log.e("tag", "url: " + url);
            JSONObject jsonObject = new JSONObject();
            String selection = MediaStore.Images.Media.DATA + " = ?";
            String[] selectionArgs = {str}; // 실제 파일의 경로
            //getContentResolver().delete(rmuri, selection, selectionArgs);
        } catch (Exception e) {
            Log.e("tag", "Exception: " + e.getMessage());
        }
    } catch (Exception e) {
        Log.e("tag", "Exception: " + e.getMessage());
    }
}
```

로그인

회원가입

파일선택

로그인 및 삭제희망 파일 선택

개발 환경 및 개발 내용 (5/10)

안드로이드 - 파일전송

```
private String getRealPathFromURI(Uri contentURI) {  
    String result;  
    Cursor cursor = ...  
    if (cursor == null) {  
        result = contentURI.getPath();  
    } else {  
        cursor.moveToPosition(-1);  
        int idx = cursor.getColumnIndex(contentURI.getScheme());  
        result = cursor.getString(idx);  
        cursor.close();  
    }  
    return result;  
}  
  
public byte[] getFileDataFromURI(Uri contentURI) {  
    ByteArrayOutputStream outputStream = new ByteArrayOutputStream();  
    bitmap.compress(Bitmap.CompressFormat.PNG, 100, outputStream);  
    byte[] bytes = outputStream.toByteArray();  
    return bytes;  
}
```

파일 데이터 추출

```
private void uploadBitmap(final Bitmap bitmap) {  
    String url = "http://172.20.10.2:3000/logs/loguploads";  
  
    VolleyMultipartRequest volleyMultipartRequest = new VolleyMultipartRequest(  
        url, Method.POST, new Response.Listener<NetworkResponse>() {  
            @Override  
            public void onResponse(NetworkResponse response) {  
                // Success  
            }  
        }, new Response.ErrorListener() {  
            @Override  
            public void onErrorResponse(VolleyError error) {  
                Toast.makeText(getApplicationContext(), error.getMessage(),  
                    Toast.LENGTH_LONG).show();  
                Log.e("GotError", error.getMessage());  
            }  
        }) {  
        @Override  
        public void addToMultipartData(MultipartEntity entity) {  
            entity.addBinaryBody("image", bitmap.toBytes());  
        }  
    };  
    volleyMultipartRequest.addRequestListener(new RequestListener() {  
        @Override  
        public void onSuccess() {  
            // Success  
        }  
        @Override  
        public void onError() {  
            // Error  
        }  
    });  
    volleyMultipartRequest.execute();  
}
```

파일 전송

파일 데이터를 추출 ⇨ 원본 파일 서버에 전송 ⇨ 파일 완전 삭제

개발 환경 및 개발 내용 (6/10)

안드로이드 - 파일삭제

```
private void delete() throws IOException {  
    File file = new File(str);  
    Long len = file.length();  
    byte[] b = new byte[len.intValue()];  
    changedata(b, file);  
    Arrays.fill(b, (byte) 1 );  
    changedata(b, file);  
    new Random().nextBytes(b);  
    changedata(b, file);  
    byte[] b1 = {0};  
    changedata(b1, file);  
  
    Intent mediaScanIntent = new Intent(  
        Uri contentUri = Uri.fromFile(file)  
    );  
    mediaScanIntent.setData(contentUri);  
    this.sendBroadcast(mediaScanIntent);  
  
    contentUri = MediaStore.Images.Media.  
    String selection = MediaStore.Images.Media.  
    String[] selectionArgs = {str};  
    getContentResolver().delete(contentUri, selection, selectionArgs);  
    Log.e("debug", "delete");  
}
```

삭제 알고리즘 - 1

```
private void changedata(byte[] b,  
    try {  
        FileOutputStream fos = new FileOutputStream(file);  
        fos.write(b);  
        fos.close();  
    } catch (IOException e) {  
        Log.e("tag: IOException", e.toString());  
    }  
}
```

삭제 알고리즘 - 2

HEX 값

```
FF D8 FF E1 FF FF 45 78 69 66 00 00 49 49 2A 00 08 00 00 00 0C 00 00 01 04  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01  
EC 96 FF 1E 08 A7 04 DF D3 42 E2 0C 04 39 6C 52 D3 E0 DC 33 30 26 04 F5 BC 85 83 DA 57 32  
00
```

총 4번에 걸친 Overwrite(FOP Algorithm)를 통해 데이터 변조 및 결과 확인

개발 환경 및 개발 내용 (7/10)

웹 서버 - 이메일 인증

```
// email
router.post('/emailregister', (req, res, next) => {

  let Email = require('./models/email');
  smtpTransport = require('nodemailer-smtp-transport');

  const mailOptions = {
    from: Email.gid,
    to: Email.email,
    subject: '이메일 인증',
    text: randomString(10),
    // html: '...'
  };

  smtpTransport.sendMail(mailOptions, (err) => {
    if (err) {
      console.log(err);
    } else {
      console.log('이메일 전송 성공');
      Euser.findOne({ email: Email.email }, (err, user) => {
        if (err) {
          console.log(err);
        } else {
          if (user) {
            console.log('이메일 인증 완료');
            res.json({ success: true, msg: '이메일 인증 완료' });
          } else {
            console.log('이메일 인증 실패');
            res.json({ success: false, msg: '이메일 인증 실패' });
          }
        }
      });
    }
  });

  console.log(randomString);
  console.log(Email.ecert);
  if (randomString == Email.ecert) {
    console.log("일치");
    res.json({ success: true, msg: '이메일 인증 완료' });
  } else {
    console.log("불일치");
    res.json({ success: false, msg: '이메일 인증 실패' });
  }
});

const smtpTransport = new smtpTransport(smtpOptions, {
  debug: true
});
```

SMTP 기본 설정

인증코드 전송 후 DB에 저장

인증코드 비교

Math 함수 사용해서 랜덤값 생성 ⇨ 이메일 전송 ⇨ 인증코드 DB 저장 ⇨ 비교

개발 환경 및 개발 내용 (8/10)

웹 서버 - 로그 서비스

```
// 로그 업로드
router.post('/logupload', (req, res) => {
  // 로그 업로드
  // DB 저장
  // 로그 다운로드
  router.get('/logdownload/:_id', (req, res) => {
    // 로그 다운로드
    // 로그 삭제
    router.get('/logdelete/:username/:_id', (req, res) => {
      // 로그 삭제
      var username = req.params.username;
      var id = req.params._id
      gfs.exist({ _id: id }, (err, file) => {
        if (err || !file) {
          res.status(404).send('File Not Found');
          return;
        }
        gfs.remove({ _id: id }, (err) => {
          if (err) res.status(500).send(err);
          res.redirect('http://localhost:4200/logs/' + username);
        });
      });
    });
  });
});
```

완전 삭제한 파일 조회 페이지 구현 / 업로드, 다운로드, 삭제 기능 구현
(파일은 AES256으로 암호화 후 DB에 저장)

개발 환경 및 개발 내용 (9/10)

윈도우 - 파일검색

```
public bool LoginCheck(string id, string password)
{
    var client = new RestClient("http://eraser2020.herokuapp.com/users/authentication");
    client.Timeout = -1;
    var request = new RestRequest(Method.POST);
    request.AddHeader("Content-Type", "application/json");
    var json_data = new JObject();
    json_data.Add("userid", id);
    json_data.Add("password", password);

    string str_json = JsonConvert.SerializeObject(json_data);
    request.AddParameter("json_data", str_json, ParameterType.RequestBody);
    IRestResponse response = client.Execute(request);
    string json_text = response.Content;
    JSdata jsdata = JsonConvert.DeserializeObject<JSdata>(json_text);
    if (jsdata.success.Equals(true))
    {
        Form1.user_id = id;
        return true;
    }
    else
        return false;
}
```

로그인

```
public UInt32 _fnBigToLittleEndian_UInt(byte[] data)
{
    if (BitConverter.IsLittleEndian)
        Array.Reverse(data);
    string hex = "";
    for (int i = 0; i < data.Length; i++)
    {
        public int _findFileNameByMFT(byte[] MFT_DATA, byte[] FileName_Data)
        {
            int c = MFT_DATA.Length - FileName_Data.Length + 1;
            int j;
            for (int i = 0; i < c; i++)
            {
                if (MFT_DATA[i] != FileName_Data[0])
                    continue;
                for (j = FileName_Data.Length - 1; j >= 1 && MFT_DATA[i + j] == FileName_Data[j]; j--)
                {
                    if (j == 0)
                        return 0;
                }
            }
            return -1;
        }
    }
    UInt32 result;
    return result;
}
```

엔디안 변경

파일명 찾기

로그인 ⇨ 볼륨 내 데이터를 LittleEndian으로 변경 ⇨ MFT안에서 파일명 검색

개발 환경 및 개발 내용 (10/10)

윈도우 - 파일삭제

```
public uint _Search_Sector_Value(uint MFT_VALUE, Dictionary<string, string> driveInfo)
{
    string physical_drive_list = "";
    byte[] input_data = _Convert(Path.GetFileName(filepath).Text); // test.txt
    DriveInfo[] drives = DriveInfo.GetDrives();
    foreach (DriveInfo drive in drives)
    {
        if (drive.IsReady)
        {
            string physical_drive = drive.Name;
            string drive_letter = drive.DriveLetter;
            string drive_path = drive.RootDirectory.FullName;
            string drive_info = drive.PhysicalName;
            Console.WriteLine($"[ {drive_letter} ] {drive_path} {drive_info}");
            foreach (var kvp in driveInfo)
            {
                if (kvp.Key == drive_letter)
                {
                    string drive_path = kvp.Value;
                    SafeFileShare share = new SafeFileShare(drive_path);
                    if (!File.Exists(drive_path + input_data))
                    {
                        Console.WriteLine($"[ {drive_letter} ] {drive_path} {input_data}");
                    }
                }
            }
        }
    }
}

public static void Main(string filepath, string filename, string url)
{
    var client = new RestClient("http://eraser2020.herokuapp.com/logs/logupload");
    client.Timeout = -1;
    var request = new RestRequest(url, Method.POST);
    request.AddFile(filename);
    request.AddParameter("filepath", filepath);
    IRestResponse response = client.Execute(request);
    Console.WriteLine(response.Content);
}

private void DeleteFile(string filename)
{
    Random ran = new Random();
    var fileInfo = new FileInfo(filename);
    var directory = fileInfo.DirectoryName;
    var name = ran.Next();
    var prefix = "";
    var newFilename = Path.Combine(directory, string.Format("{0}{1}", prefix, name));
    while (File.Exists(newFilename))
    {
        if (nameSeed == null) { nameSeed = new Random(); }
        name += nameSeed.Next();
        if (name > int.MaxValue / 10)
        {
            prefix += "x";
            name = ran.Next();
        }
    }
}
```

섹터값 검색

파일 전송

삭제 알고리즘

볼륨 안에 파일명으로 섹터값 검색 ⇨ 원본 파일 전송 ⇨ 파일 완전 삭제

개발 시스템 운영(1/11)

웹 서버 - 이메일 인증(1/2)

회원가입 완료되었습니다.

확인

회원가입

이메일 인증 후 가입이 완료됩니다.

로그인

인증번호가 발송되었습니다.

인증번호 요청

확인

Eraser

Eraser

Eraser.T

김성일

asd

...

...

01047672015

asd

...

kdsko123@naver.com

인증번호 요청

인증번호

회원가입 ⇨ 로그인 ⇨ 인증번호 요청 (인증하지 않으면 로그인 불가)

개발 시스템 운영 (2/11)

웹 서버 - 이메일 인증 (2/2)

```
_id: ObjectId("5f85b7576bcada7860b15e69...")
username: "김성일"
userid: "asd"
userpassword: "$2a$10$dYt7UvcdeE2.jcMp0vI51ePAmMui7"
userphone: 1047672015
semail: "kdsko123@naver.com"
ecert: "oirm996sh81"
auth: false
__v: 0
```

DB 저장

이메일 확인

인증 비교

☆ 인증번호 요청이 발송되었습니다.

가입이 완료되었습니다.

보낸사람 VIP Erase
받는사람 <kdsko123@naver.com>

확인

인증 완료

```
_id: ObjectId("5f85b7576bcada7860b15e69...")
username: "김성일"
userid: "asd"
userpassword: "$2a$10$dYt7UvcdeE2.jcMp0vI51ePAmMui7"
userphone: 1047672015
semail: "kdsko123@naver.com"
ecert: "oirm996sh81"
auth: true
__v: 0
```

인증코드 DB 저장 ⇨ G메일의 SMTP를 이용 전송 ⇨ 인증 코드 비교 ⇨ 완료

윈도우 - 파일생성

\$RECYCLE.BIN 2020-10-14 오후 4:24 **파일 생성**
System Volume Information 2020-09-26 오후 12:06

Evidence Tree File List **MFT 추출**

G:\Test [NTFS] [root]

Name	Size	Type
\$Extend	1	Directory
\$RECYCLE.BIN	1	Directory
System Volume Infor...	1	Directory
\$AttrDef	3	Regular File
\$BadClus	0	Regular File
\$Bitman	160	Regular File

파일 생성 확인

	A	B	J	K	L	
1	RecordOff	Signature	FN_FileName	FilePath	HEADER_F	RecordActive FileSizeByt
42	0x0000A0	(GOOD	KongJongHyun.txt	:\#KongJongHyu	FILE	ALLOCATED 32
43	0x0000A4	(GOOD	\$RECYCLE.BIN	:\#\$RECYCLE.BIN	FOLDER	ALLOCATED
44	0x0000A8	(GOOD	S-1-5-21-2363174748	:\#\$RECYCLE.BIN	FOLDER	ALLOCATED
45	0x0000AC	(GOOD	desktop.ini	:\#\$RECYCLE.BIN	FILE	ALLOCATED 129
46	0x0000B0	(GOOD	Test5555.txt	:\#Test5555.txt	FILE	ALLOCATED 14
47	0x0000B4	(GOOD	Test6666.txt	:\#Test6666.txt	FILE	ALLOCATED 14

파일 생성 및 삭제를 확인하기 위한 G(볼륨)의 MFT 추출

개발 시스템 운영(4/11)

윈도우 - 파일삭제

The screenshot illustrates a sequence of operations in a Windows environment:

- 로그인 (Login):** A text box contains user credentials: `_id: ObjectId("5f89d977894ed40017e2b0fb")`, `username: "지우개 팀"`, `userid: "kong"`, `userpassword: "$2a$10$//F/TQgx87o7vXp.DmbLwut5oj8HnSEqRLW9X7xPosY6ugQu0wv46"`, `userphone: 1067105029`, `semail: "test12345@ruu."`, `ecert: "xs4fka5ufr"`, `auth: true`, and `_v: 0`.
- 파일 삭제 (File Deletion):** A file explorer window shows the path `G:\₩KongJongHyun.txt` selected.
- 파일 확인 (File Confirmation):** A table lists files with their names and modification dates:

이름	수정한 날짜
Test6666	2020-10-14 오후 11:10
Test5555	2020-10-14 오후 11:10
- 원본 업로드 (Original Upload):** A '로그 목록' (Log List) table shows the upload of a file:

파일 이름	날짜 / 시간	다운	삭제
Test77777.txt	20-10-15 / 01:35	다운	삭제
KongJongHyun.txt	20-10-14 / 23:14	다운	삭제

로그인 ⇨ 파일 완전 삭제 ⇨ 삭제된 파일 서버에 업로드 ⇨ 로그 확인

윈도우 - 복원시도

	A	B	J	K	L
1	RecordOff	Signature	FN_FileName	FilePath	HEADER_
42	0x0000A0C	GOOD	1728074700	..WDFEL401172807470	FILE DELETED
43	0x0000A4C	GOOD			
44	0x0000A8C	GOOD			
45	0x0000ACC	GOOD			
46	0x0000B0C	GOOD			
47	0x0000B4C	GOOD			

MFT 내용 확인

파일 복구

Test (G:) Scan

Filename Path

No files were found

No files were found. Would you like to scan again with Deep Scan enabled? Note this can take over an hour on a large drive.

프로그램을 통한 파일 완전삭제 후 MFT 재 추출 하여 데이터 확인
⇒ 복원프로그램을 통한 복원시도(실패)

개발 시스템 운영(6/11)

윈도우 - 파일 재 생성

이름	수
Test6666	2020-10-14 오후 11:10
Test5555	2020-10-14 오후 11:10
\$RECYCLE.BIN	2020-10-14 오후 11:10
System Volume Information	2020-10-14 오후 11:09
Test77777	2020-10-14 오후 11:22

새로운 파일 생성

	A	B	J	K	생성된 파일 확인	
1	RecordOff	Signature	FN_FileName	FilePath	HEADER_F	RecordAct
42	0x0000A0	GOOD	Test77777.txt	:₩Test77777.txt	FILE	ALLOCATE
43	0x0000A4	GOOD	\$RECYCLE.BIN	:₩\$RECYCLE.BIN	FOLDER	ALLOCATE
44	0x0000A8	GOOD	S-1-5-21-2363174748-	:₩\$RECYCLE.BIN₩S-	FOLDER	ALLOCATE
45	0x0000AC	GOOD	desktop.ini	:₩\$RECYCLE.BIN₩S-	FILE	ALLOCATE
46	0x0000B0	GOOD	Test5555.txt	:₩Test5555.txt	FILE	ALLOCATE
47	0x0000B4	GOOD	Test6666.txt	:₩Test6666.txt	FILE	ALLOCATE

DELETE 되었던 MFT Entry에 새로운 파일 생성 확인

개발 시스템 운영(7/11)

안드로이드 - 회원가입

The image displays three sequential screenshots of an Android application's sign-up process:

- Sign Up / 회원가입:** A form with a blue header. Fields include: User Name (성철), ID (cheol), Password (masked with dots), Confirm Password (masked with dots), and Phone (01012341234).
- Email / 이메일 인증:** A screen with a blue header. It shows an Email field containing 'kach3242@hanmail.net' and a Cert Number field. A blue arrow button is next to the email field.
- 인증번호 요청:** A dark grey screen with a white dialog box. The dialog box contains the text '알림' (Notice) and '인증번호가 발송되었습니다.' (Verification number has been sent). A '예' (Yes) button is at the bottom right.

회원가입, 이메일인증을 통한 계정 생성 및 앱 사용 가능

개발 시스템 운영 (8/11)

안드로이드 - 로그인

The image shows two screenshots of an Android application. The left screenshot is a login screen with a blue header and a white background. It features a 'Cert Number' field with the value 'qckwk07bwcg' and a 'Log In' button. A red box labeled '인증번호 확인' (Verify Certificate Number) is positioned above the cert number field. A grey box labeled 'DB 저장' (DB Save) is positioned above the 'Log In' button. A white box containing JSON data is overlaid on the left side of the screen, with a red box labeled '로그인' (Login) positioned above it. The JSON data is:

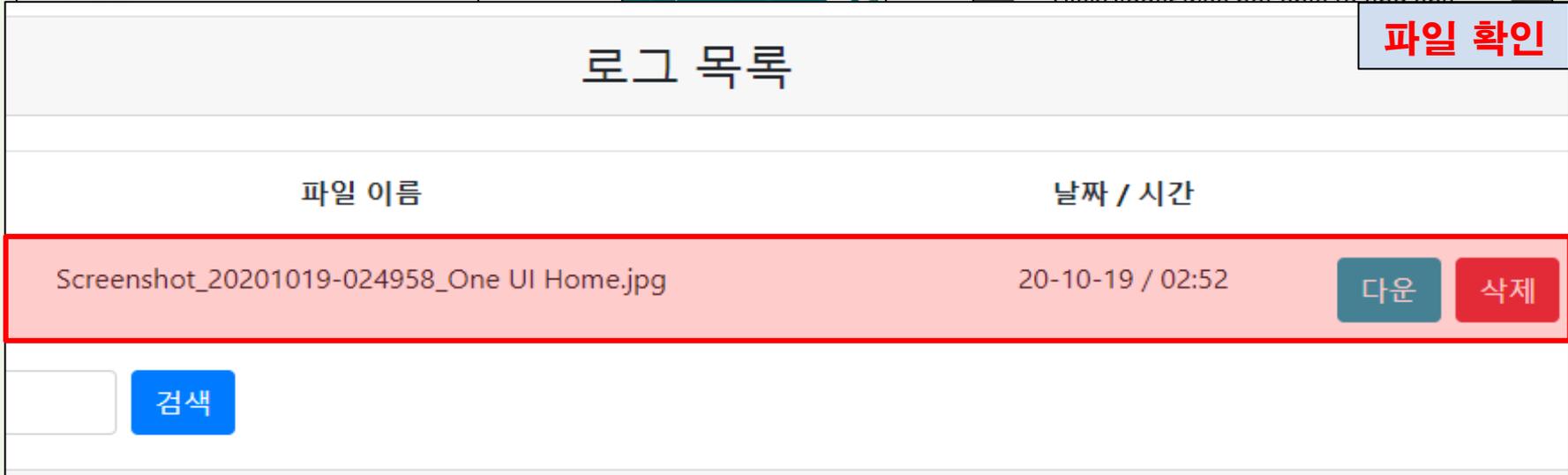
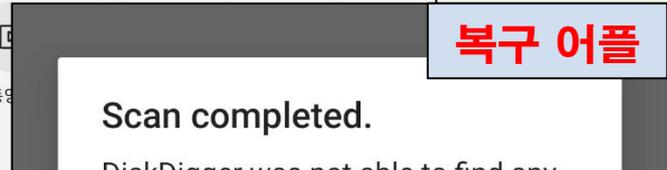
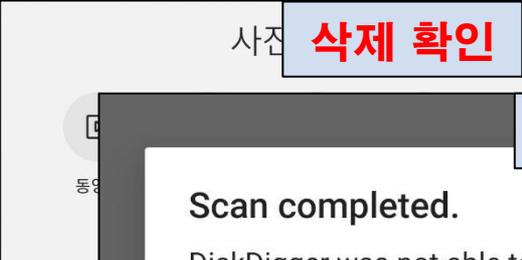
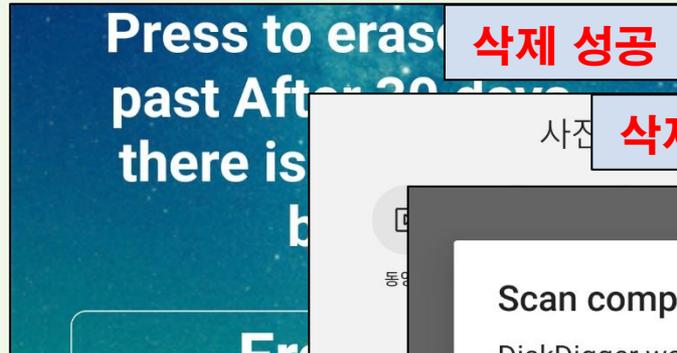
```
{  "_id": ObjectId("5f8c8d6ed39652..."),  "username": "성철",  "userid": "cheol",  "userpassword": "$2a...$",  "userphone": "0101234...",  "semail": "kach3242@...",  "ecert": "qckwk07bwcg...",  "auth": true,  "__v": 0}
```

 A red arrow points from the 'Log In' button to the right screenshot. The right screenshot is a confirmation screen with a dark blue background and white text. It says 'Press to erase y...' and 'past After 30 days, there is no turning back'. A red box labeled '메인' (Main) is positioned above the text. A white button labeled 'Erasing' is centered on the screen. A dark grey notification bubble at the bottom says '성철님이 로그인 되었습니다.' (Seongcheol has logged in).

로그인 ⇨ 파일 완전삭제 기능 페이지로 이동

개발 시스템 운영(9/11)

안드로이드 - 파일삭제



파일 선택 및 삭제 ⇨ 삭제 성공, 확인 ⇨ 복원 프로그램 설치 후 복원 시도

개발 시스템 운영 (10/11)

웹 서버 - 로그 서비스 (1/2)

The screenshot shows a web application interface with several components:

- POST 전송**: A button or label indicating a POST request.
- DB 저장**: A button or label indicating data is being saved to the database.
- 로그 목록**: A title for the log list page.
- 로그 페이지**: A label for the log page.
- 검색**: A search button.
- Page Navigation**: A set of buttons for navigating between pages: First, Previous, 1 (selected), Next, Last.

The log list table contains the following data:

파일 이름	날짜 / 시간
test.png	20-10-13 / 22:19

The background shows a terminal window with the following command and output:

```
MongoDB Enterprise > db.fs.files.find().pretty()
{
  "_id" : "test.png",
  "filename" : "test.png",
  "uploadDate" : ISODate("2020-10-13T22:19:00.000Z")
}
```

POST로 파일 전송 ⇨ DB에 파일 저장 ⇨ 로그 페이지에 저장된 파일 조회

결론 및 기대 효과

○ 결 론

- 기존 알고리즘을 기반으로 자체 연구한 알고리즘에 의거 완전 삭제된 파일은 파일 복구 도구로 복원이 불가능한 시스템을 구현
- 기존의 사용하는 핵심적인 포렌식 기술인 파일 복구 기술의 정반대되는 기술을 구현해 차별성있는 시스템 개발

○ 기대 효과

- 파일이 완전 삭제되어 포렌식 도구나 파일 복구 도구 등으로 복원이 되지 않아 개인정보 유출을 방지하는 시스템으로 활용 기대
- 특히 예산이 항상 부족한 학교, 군부대 등에서 저렴한 가격으로 보안기능이 강화된 시스템으로 유용성 증대 가능. 끝.

Q & A

감사합니다.