



이은서
1971100



조장
이채은
1971086



서아름
1971236



송승민
1771378

악성 위협 감지 시스템

2022 사이버보안 캡스톤 디자인 2조



프로젝트 소개



코로나19와 같은 사회적 요인으로 업무 **전산화가 증가** 함에 따라 **사이버 공격 증가**

"APT 공격"

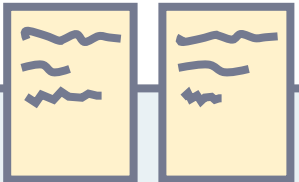
신규로 개발되는 기술과 전술을 이용해 다양하게 진화하는 공격

수일 ~ 수년 단위 지속되는 공격 유형

APT공격의 주요 타겟인 피싱 메일과 악성 파일을 검사하여 위협 감지, 예방할 수 있는 여러 기능을 제공하는 시스템



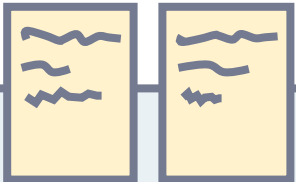
악성 위협 감지 시스템



제안발표 피드백 반영



1. 제시하고 있는 기능을 구현하기 위해서는 **타겟 브라우저를 제시**
 - 광고배너 차단 기능을 크롬 프로그램으로 개발하여 **타겟 브라우저를 chrome으로 설정하였습니다.**
2. 기존 유사 프로그램인 dangerzone에 HWP파일 탐지기능만 추가하는 것인지, **어떤 차별화가 있는지에 대한 구체적 제시**
 - 기존 Dangerzone 의 기능 또한 모두 가능하게 하며, **HWP파일 감염 여부 탐지 후**
감염된 파일일 경우, **안전한 Flat Document 형식인 PDF로 변환**하는 기능을 구현 중에 있습니다.
3. 프로젝트의 범위가 넓어 보이는 만큼 잘 **계획**을 세워서 진행
4. 인원이 많은 만큼 구현되는 부분에서 **역할의 분담**에 좀더 신경을 쓰기 바람
 - 피드백 확인 후 팀원들과 다시 계획과 역할을 세부 분담하여 진행

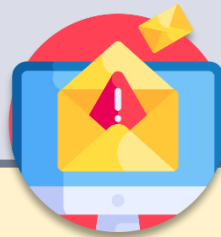


핵심 기능



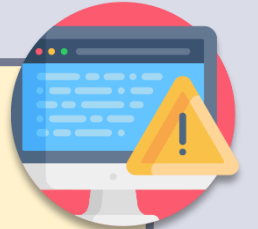
HWP 파일 변환

Active contents 무해화
contents 변환(그림, 글 등)



악성 파일 탐지

악성 매크로 탐지
악성 파일 유무 전달



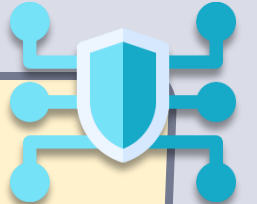
악성 URL 탐지

URL 위험도 알림



기록 저장

유해 URL 차단
차단 URL 기록 보관



기능 통합

Chrome 이용



기능 별 핵심 사항



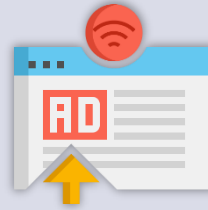
파일변환

감염 파일의 악성코드 삭제
문서 내용 무 손상 상태 반환



파일 검사 보고서

감염 파일이 보유한 악성 코드 데이터 제공



광고 배너 차단 기능

광고 배너를 가려 광고 없이 사이트 이용 가능

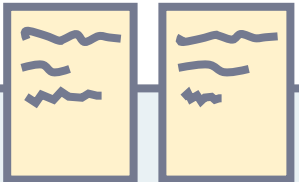
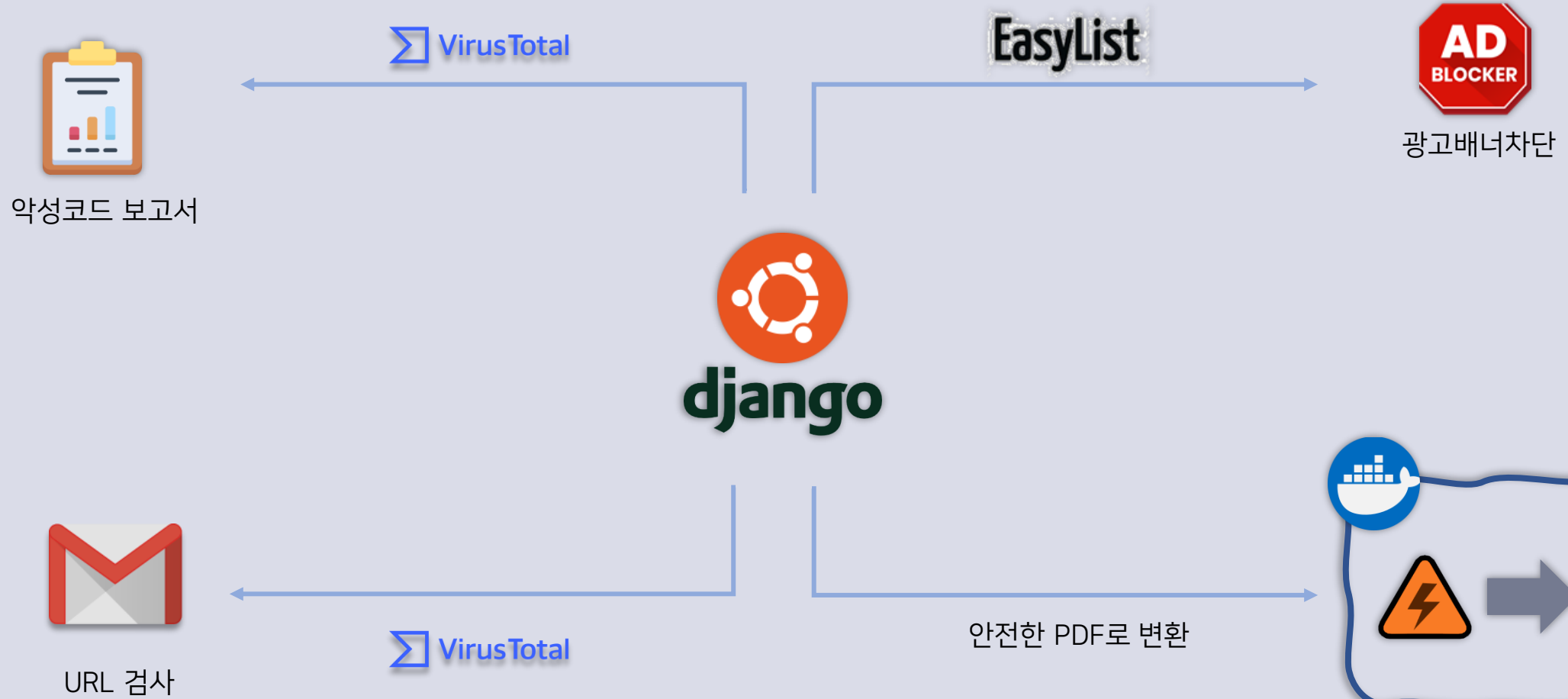


URL 검사 기능

URL 스캔 후 어떤 위협사항 보고




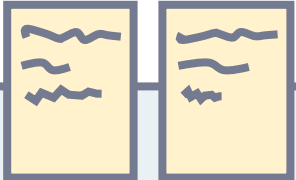
프로그램 개요



프로젝트 일정 계획 및 진척 상황 보고

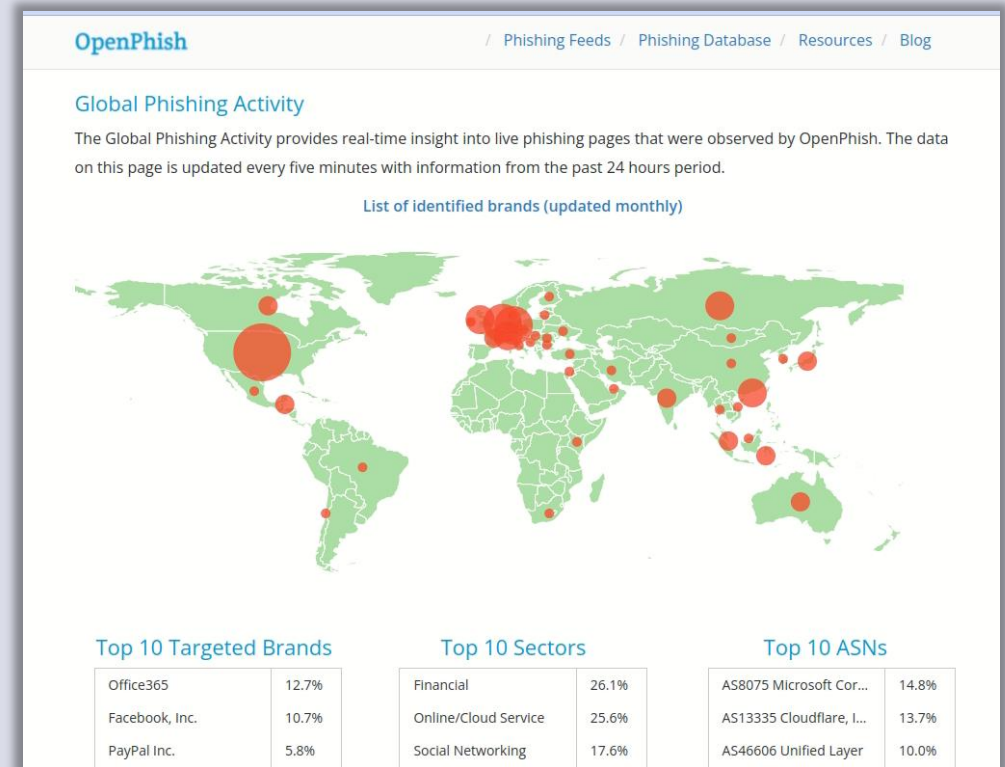
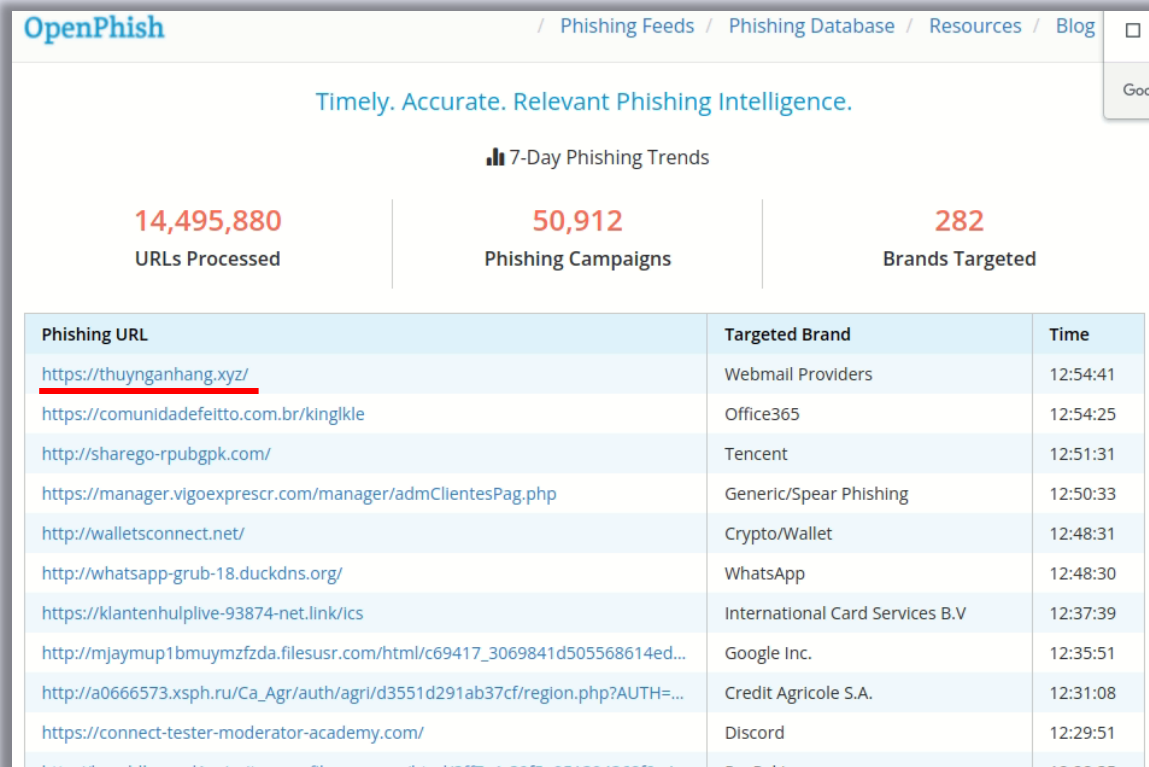
주차	수행내용
3주차	제안 발표
 4주차	URL 위험도 감지 기능 설계
 5주차	URL 위험도 감지 기능 설계
 6주차	광고배너 차단 기능 구현
 7주차	광고배너 차단 기능 구현
 8주차	악성 hwp 파일 ➡ PDF 변환
9주차	중간 발표

주차	수행내용
10주차	
 11주차	파일 감염 여부 검사 기능 구현
12주차	
13주차	
14주차	
15주차	프로젝트 최종 발표

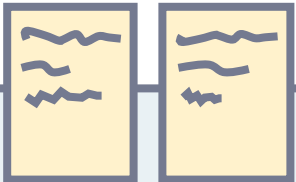


수행 내역 및 시연

1. URL scan

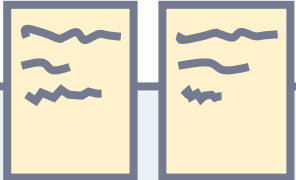
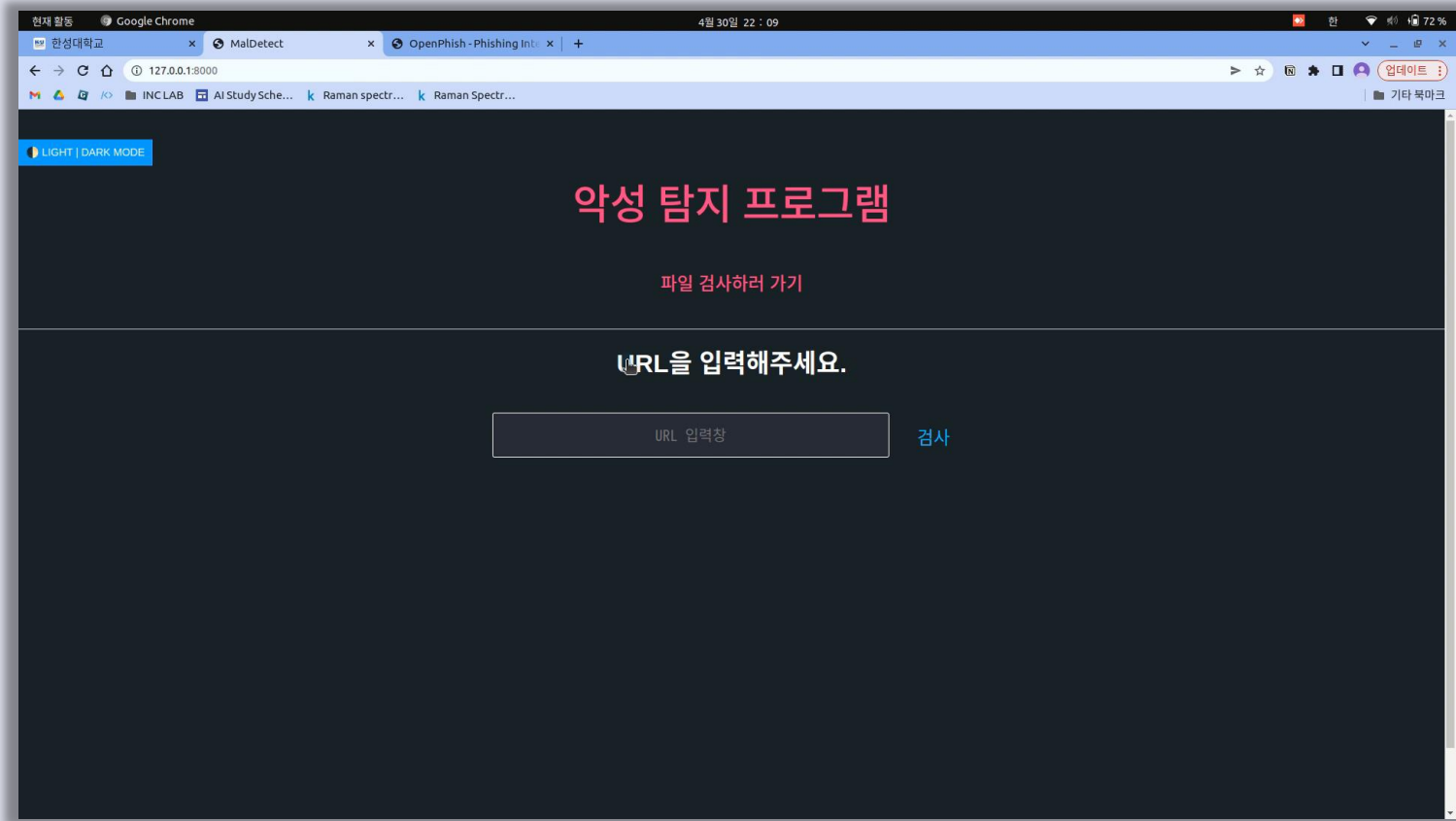


피싱 사이트, 악성코드 배포 사이트 목록을 실시간으로 업데이트 하는 사이트



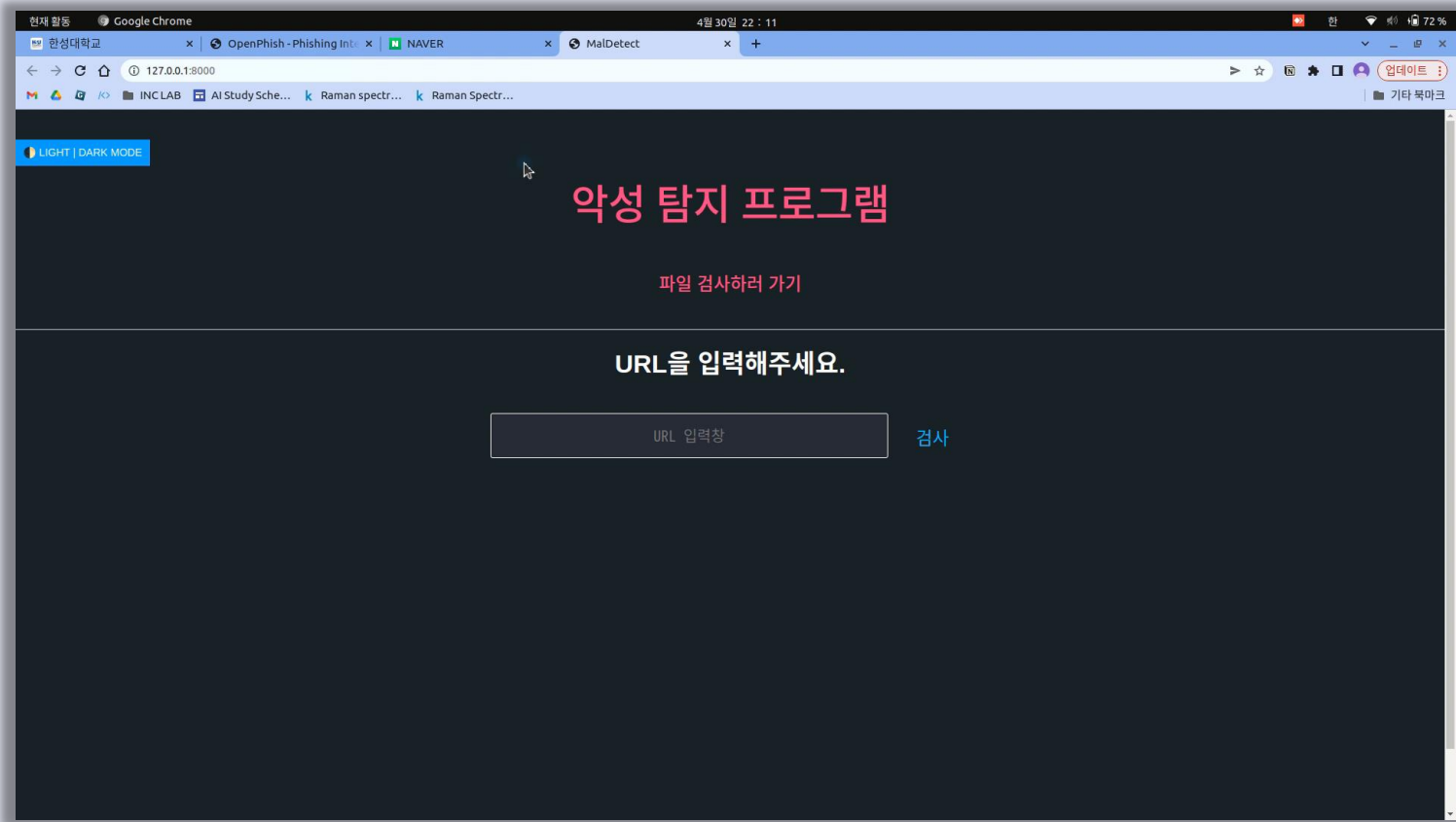
수행 내역 및 시연

1. URL scan - 위험 요소가 존재하는 URL



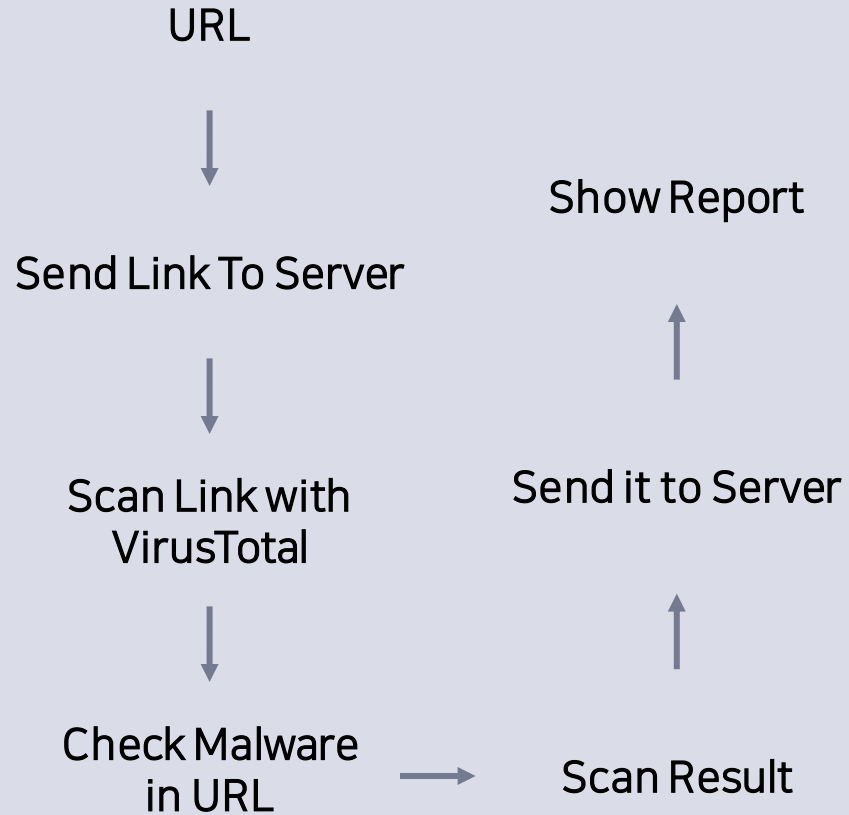
수행 내역 및 시연

1. URL scan - 안전한 URL



수행 내역 및 시연

1. URL scan



```
if form.is_valid():
    geturl = request.POST.get("url", "")
    if is_valid_url(geturl)==False:
        return render(request, 'urlerror.html')
    cmd = "curl -s --request GET --url 'https://www.virustotal.com/vtapi/v2/url/report?apikey="+apikey+"&resource="+geturl+"'"
    resp = os.popen(cmd)
    output = resp.read()
    #changing start
    my_apikey = "78a4f8a70dc6f5adb7a29b28f899746a39b009784f43579e5867acc3b45c5a1"
    my_url = geturl
    url_scan = 'https://www.virustotal.com/vtapi/v2/url/scan'
    scan_params = {'apikey': my_apikey, 'url': my_url}
    scan_response = requests.post(url_scan, data=scan_params)

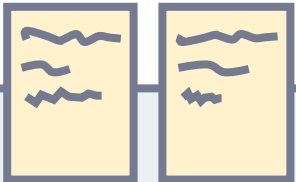
    print('VirusTotal URL SCAN START (60 Seconds Later) : ', my_url, '\n')

    #time.sleep(60)

    url_report = 'https://www.virustotal.com/vtapi/v2/url/report'
    report_params = {'apikey': my_apikey, 'resource': my_url}
    report_response = requests.get(url_report, params=report_params)
    report = report_response.json()
    report_scan_date = report.get('scan_date')
    report_scan_result = report.get('scans')
    report_scan_venders = list(report['scans'].keys())
    num = 1
    print(report.get('verbose_msg','\n'))
    print('scan date: ',report_scan_date)
```

```
try:
    imprint = json.loads(output)['positives']
except:
    return render(request, 'urlerror.html')
print(imprint)
if imprint==1:

    result2="주의가 필요한 사이트입니다."
    result4="<발견된 Malware>"
    result5="해당 사이트로 이동하기"
    k=0
    for vender in report_scan_venders:
        outputs=report_scan_result[vender]
        outputs_keys = report_scan_result[vender].get('result')
        if(outputs_keys=="malware site" or outputs_keys=="malicious site"):
            k+=1
            if (int(imprint) == int(k)) :
                result3=str(vender)
                f.write(result3)
            else:
                result3=str(vender)+", "
                f.write(result3)
```



수행 내역 및 시연

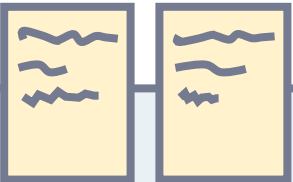
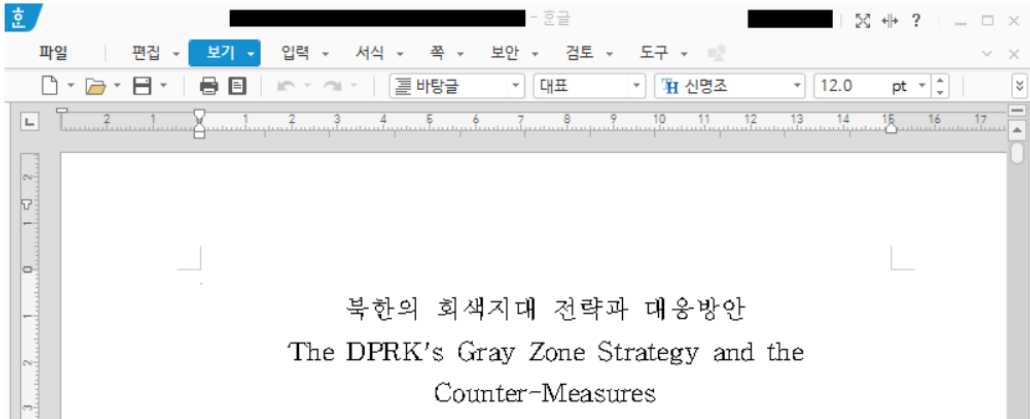
2. File scan

ASEC 악성코드 정보 조치 가이드 보안 위협 동향 AhnLab

Posted on 2020년 7월 3일

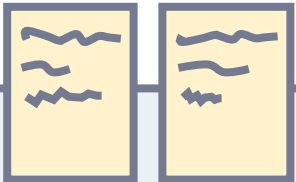
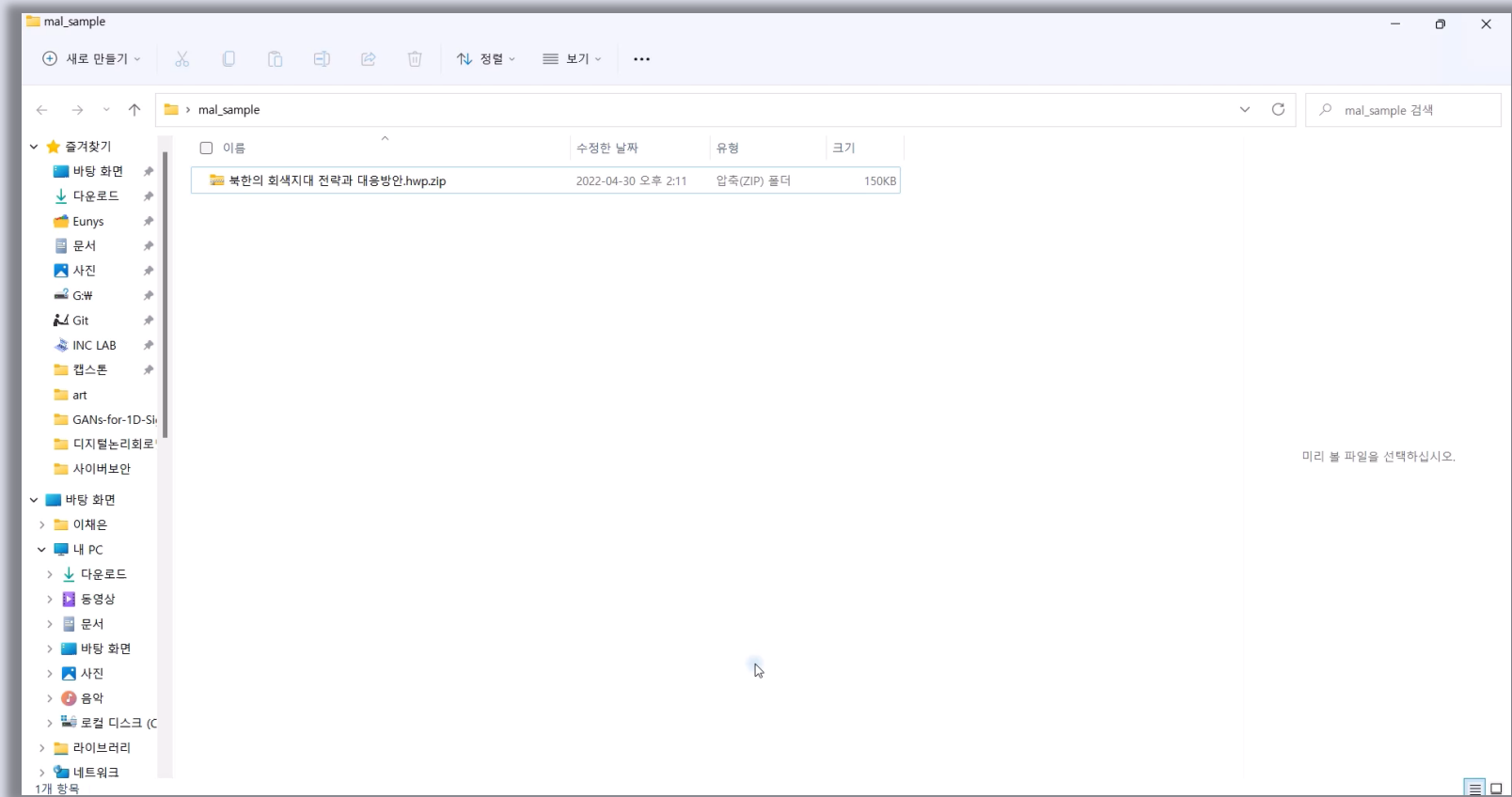
'북한의 회색지대 전략과 대응방안' 한글문서(HWP) 유포 중

ASEC 분석팀은 최근 '북한의 회색지대 전략과 대응방안' 내용의 한글 문서 파일이 유포 중인 것을 확인하였다. 한글문서는 2019년 10월 21일에 작성되었으며, 2020년 6월 23일에 공격자에 의해 수정된 것으로 추정된다. 마지막으로 해당 문서를 저장한 사람은 [Venus.H](#)로 확인되었다. 아래의 그림은 **EPS 취약점 스크립트를 포함한 악성 한글문서**의 본문 내용을 나타낸다.



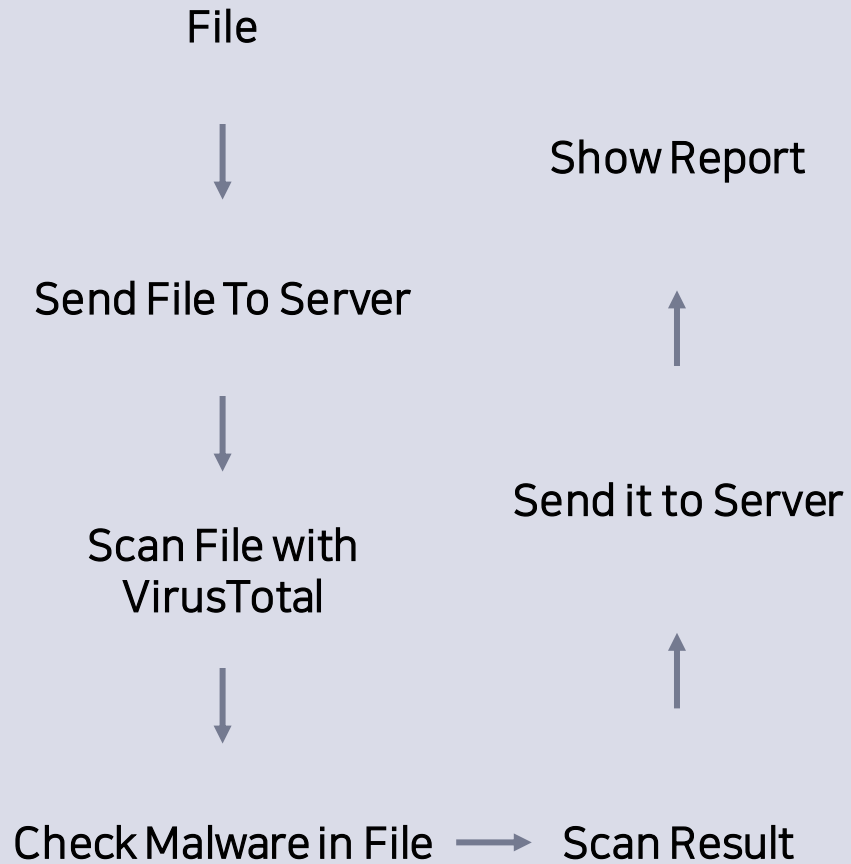
수행 내역 및 시연

2. File scan



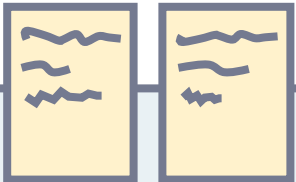
수행 내역 및 시연

2. File scan



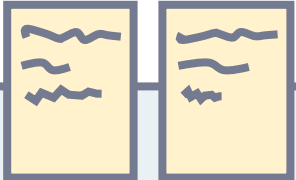
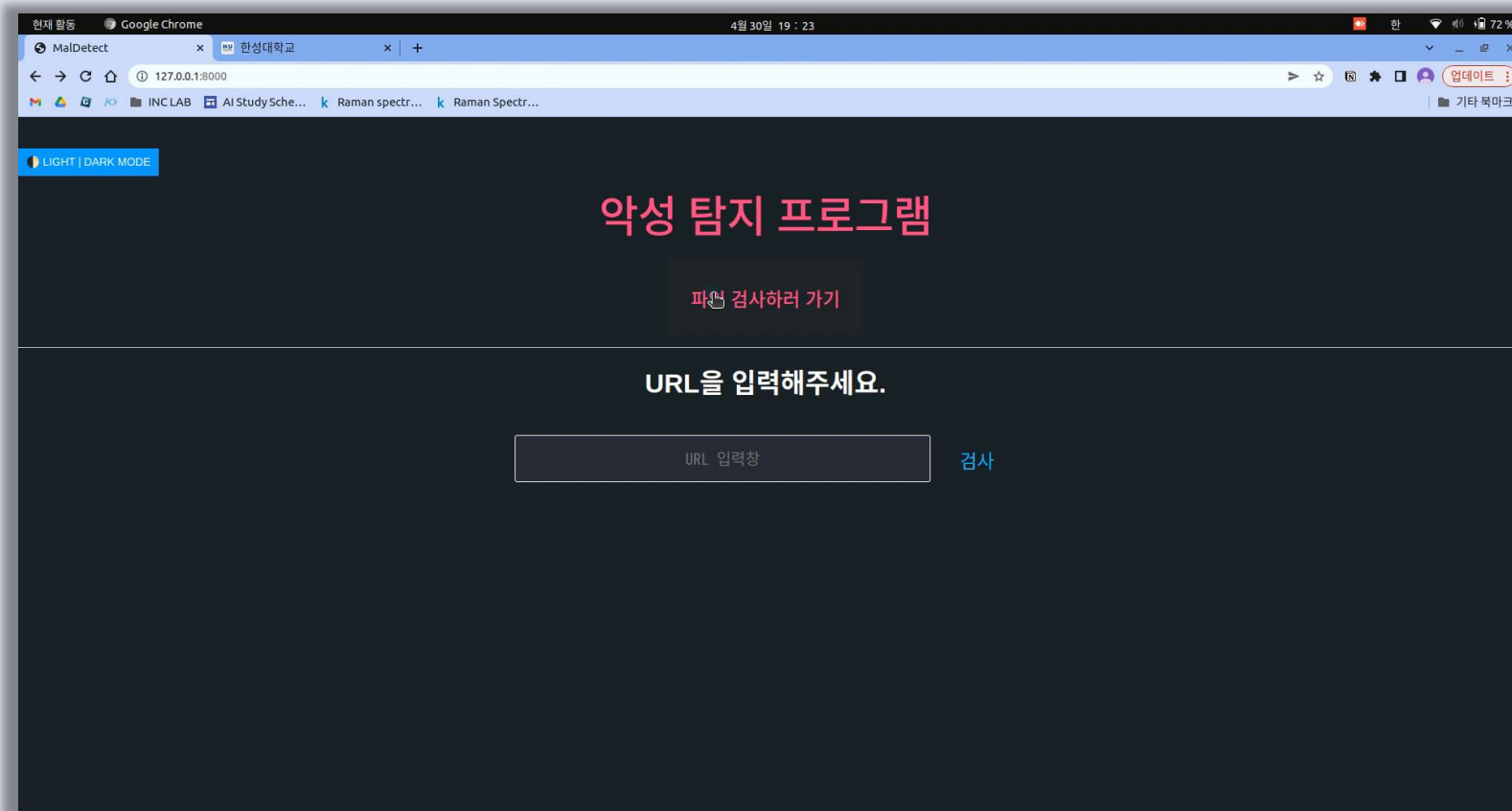
```
if request.method == 'POST':
    uploaded_file = request.FILES['document']
    fs = FileSystemStorage()
    name = fs.save(uploaded_file.name, uploaded_file)
    context['url'] = fs.url(name)
    file = uploaded_file.name
    files = {'file': (file, open(file, 'rb'))}
    url_scan = 'https://www.virustotal.com/vtapi/v2/file/scan'
    url_scan_params = {'apikey': apikey}
    response_scan = requests.post(url_scan, files=files, params=url_scan_params)
    result_scan = response_scan.json()
    scan_resource = result_scan['resource']
    print('Virustotal FILE SCAN START (60 Seconds Later) : ', file, '\n')

    url_report = 'https://www.virustotal.com/vtapi/v2/file/report'
    url_report_params = {'apikey': apikey, 'resource': scan_resource}
    response_report = requests.get(url_report, params=url_report_params)
    report = response_report.json()
    report_scan_date = report.get('scan_date')
    report_scan_sha256 = report.get('sha256')
    report_scan_md5 = report.get('md5')
    report_scan_result = report.get('scans')
    report_scan_vendors = list(report['scans'].keys())
    report_scan_vendors_cnt = len(report_scan_vendors)
    url_check=file
    num = 1
    print(report.get('verbose_msg'), '\n')
```



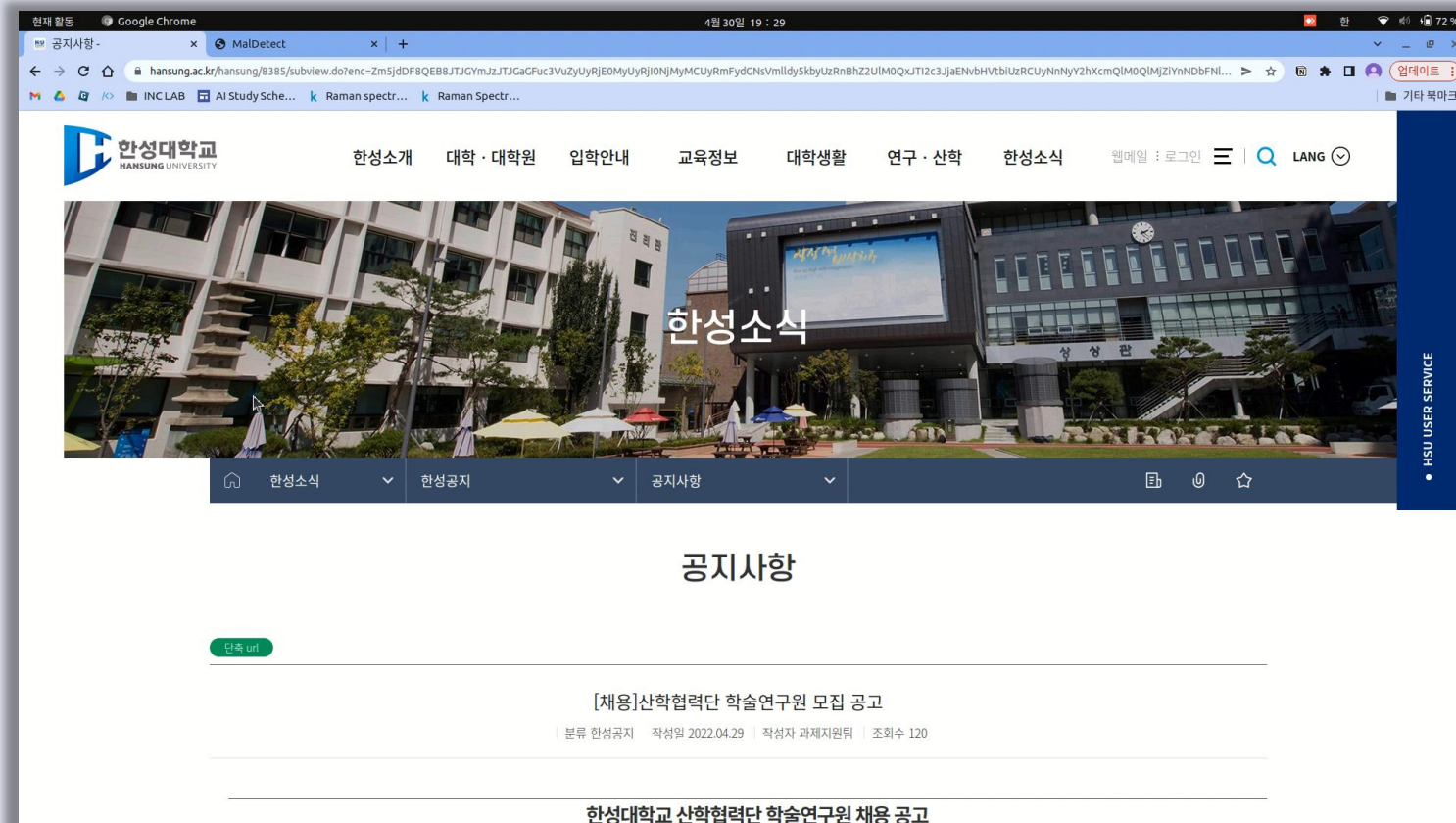
수행 내역 및 시연

2. File scan - 악성 파일 검사



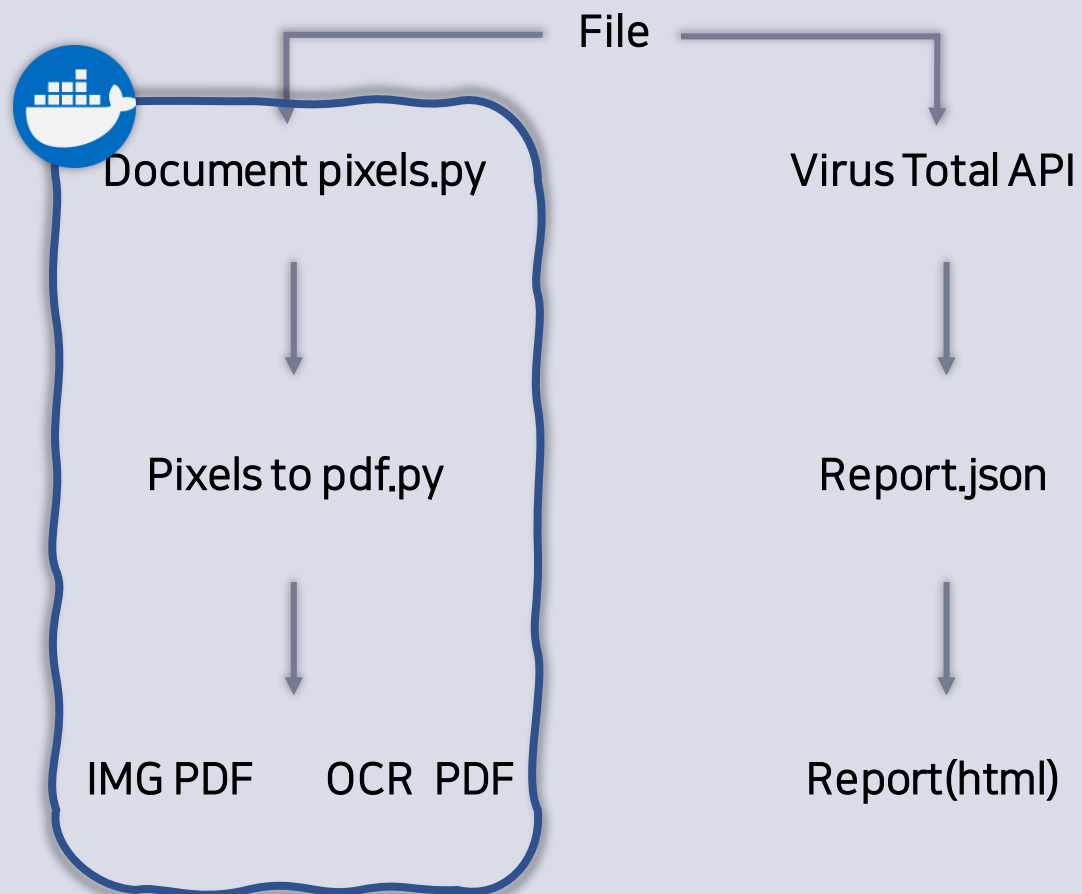
수행 내역 및 시연

2. File scan - 안전한 파일 검사

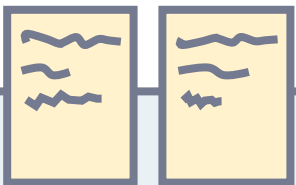


수행 내역 및 시연

3. 악성 File → Flat PDF 변환

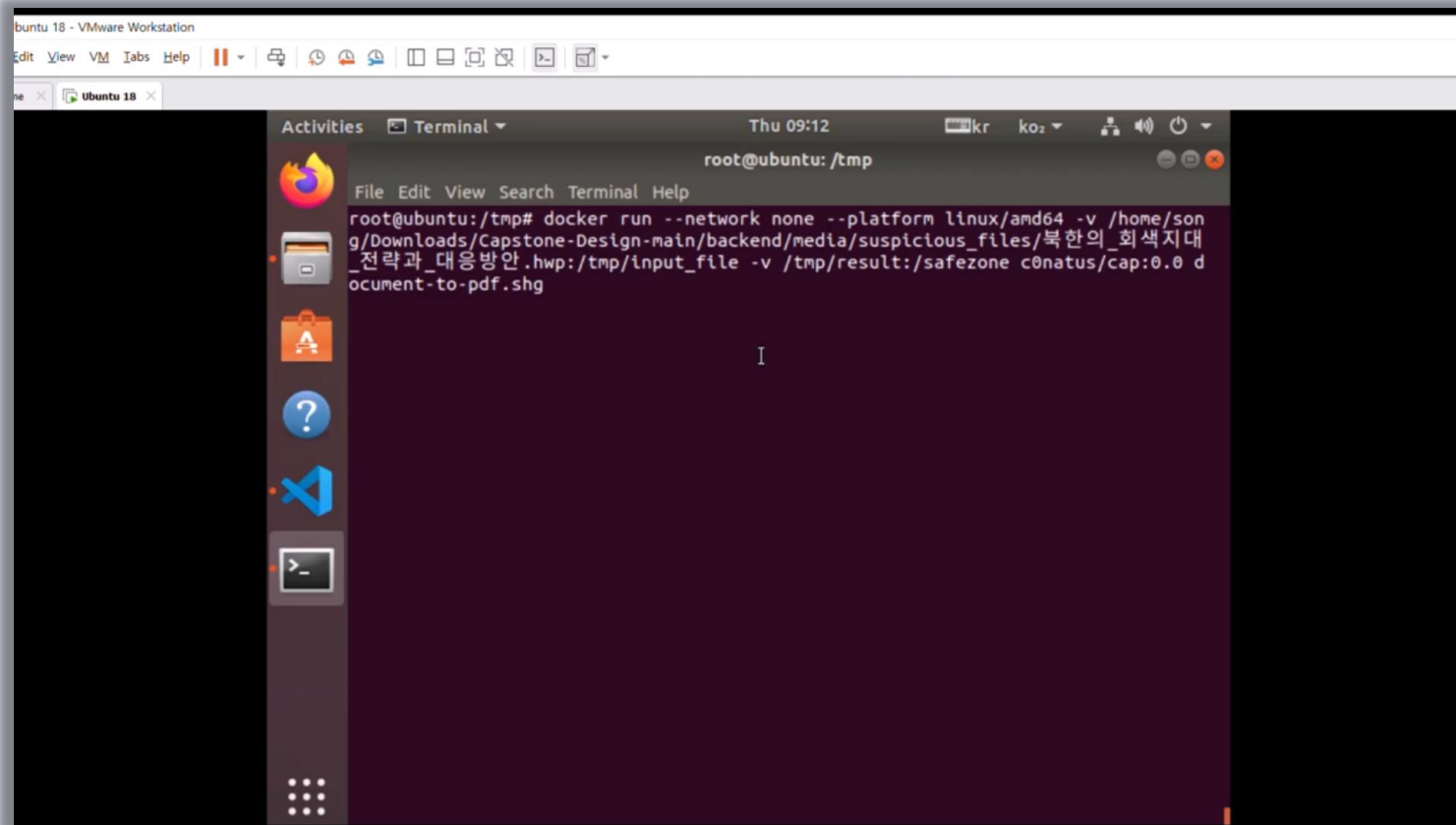


```
elif conversion["type"] == "pyhwp":
    print_flush(f"Converting to xhtml using pyhwp")
    args = [
        "hwp5html",
        "--output",
        "/tmp/xhtml",
        "/root/dangerzone/input_file",
    ]
    try:
        p = subprocess.run(args, timeout=60)
    except subprocess.TimeoutExpired:
        print_flush(
            "Error converting document to xhtml, pyhwp timed out after 60 seconds"
        )
        sys.exit(1)
    if p.returncode != 0:
        print_flush(f"Converting to xhtml failed: {p.stdout}")
        sys.exit(1)
    print_flush(f"Converting to PDF using wkhtmltopdf")
    args = [
        "xvfb-run",
        "wkhtmltopdf",
        "/tmp/xhtml/index.xhtml",
        "/tmp/input_file.pdf",
    ]
    try:
        p = subprocess.run(args, timeout=60)
    except subprocess.TimeoutExpired:
        print_flush(
            "Error converting document to PDF, wkhtmltopdf timed out after 60 seconds"
        )
        sys.exit(1)
    if p.returncode != 0:
        print_flush(f"Converting to PDF failed: {p.stdout}")
        sys.exit(1)
    pdf_filename = "/tmp/input_file.pdf"
else:
    print_flush("Invalid conversion type")
    sys.exit(1)
```

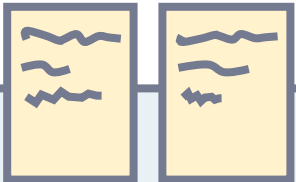


수행 내역 및 시연

3. 악성 File → Flat PDF 변환



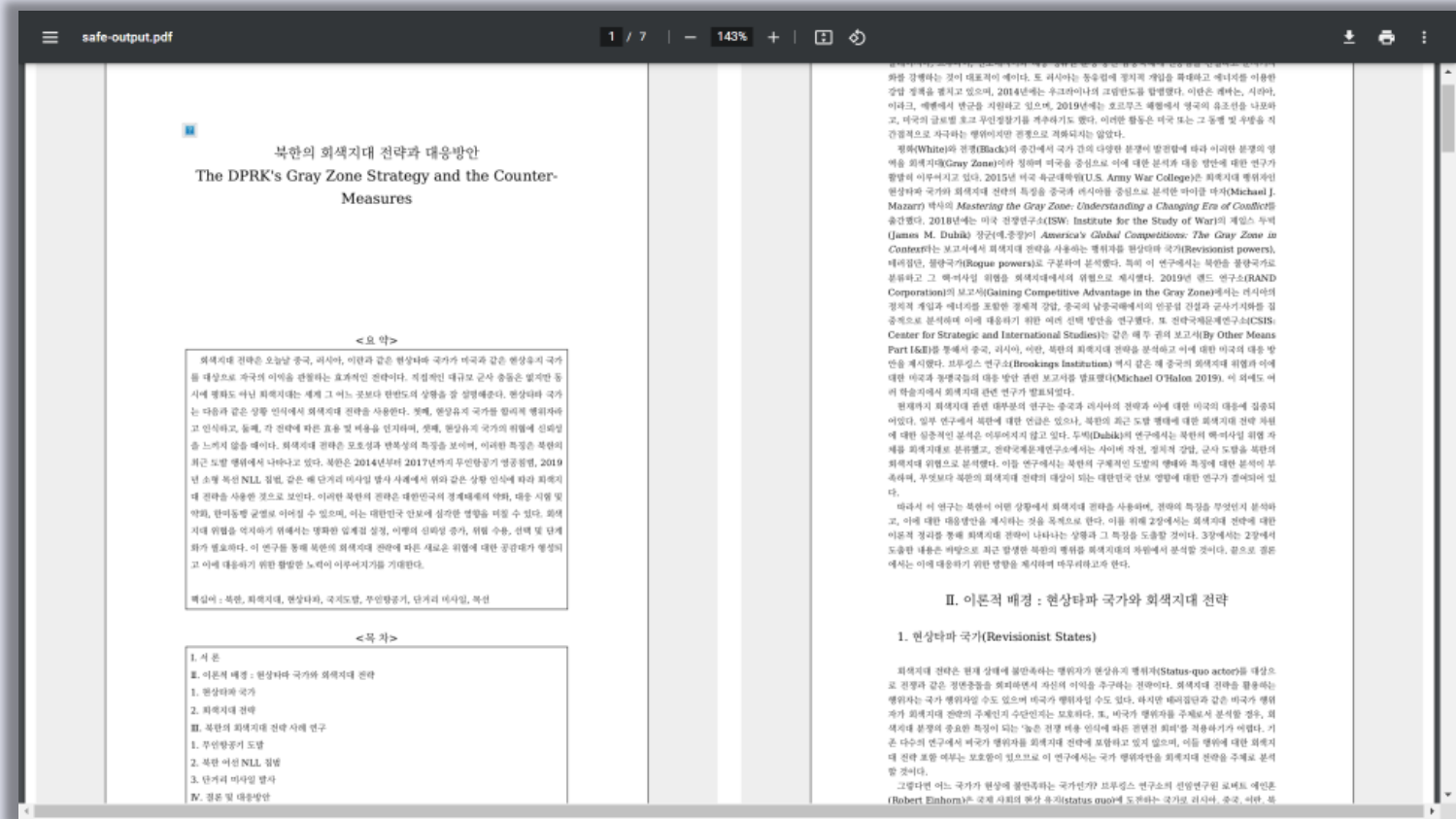
```
buntu 18 - VMware Workstation
Edit View VM Tabs Help
Ubuntu 18
Activities Terminal Thu 09:12 kr ko2
root@ubuntu: /tmp
File Edit View Search Terminal Help
root@ubuntu:/tmp# docker run --network none --platform linux/amd64 -v /home/son
g/Downloads/Capstone-Design-main/backend/media/suspicious_files/북한의_회색지대
_전략과_대응방안.hwp:/tmp/input_file -v /tmp/result:/safezone c0natus/cap:0.0 d
ocument-to-pdf.shg
```



수행 내역 및 시연

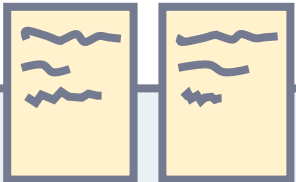
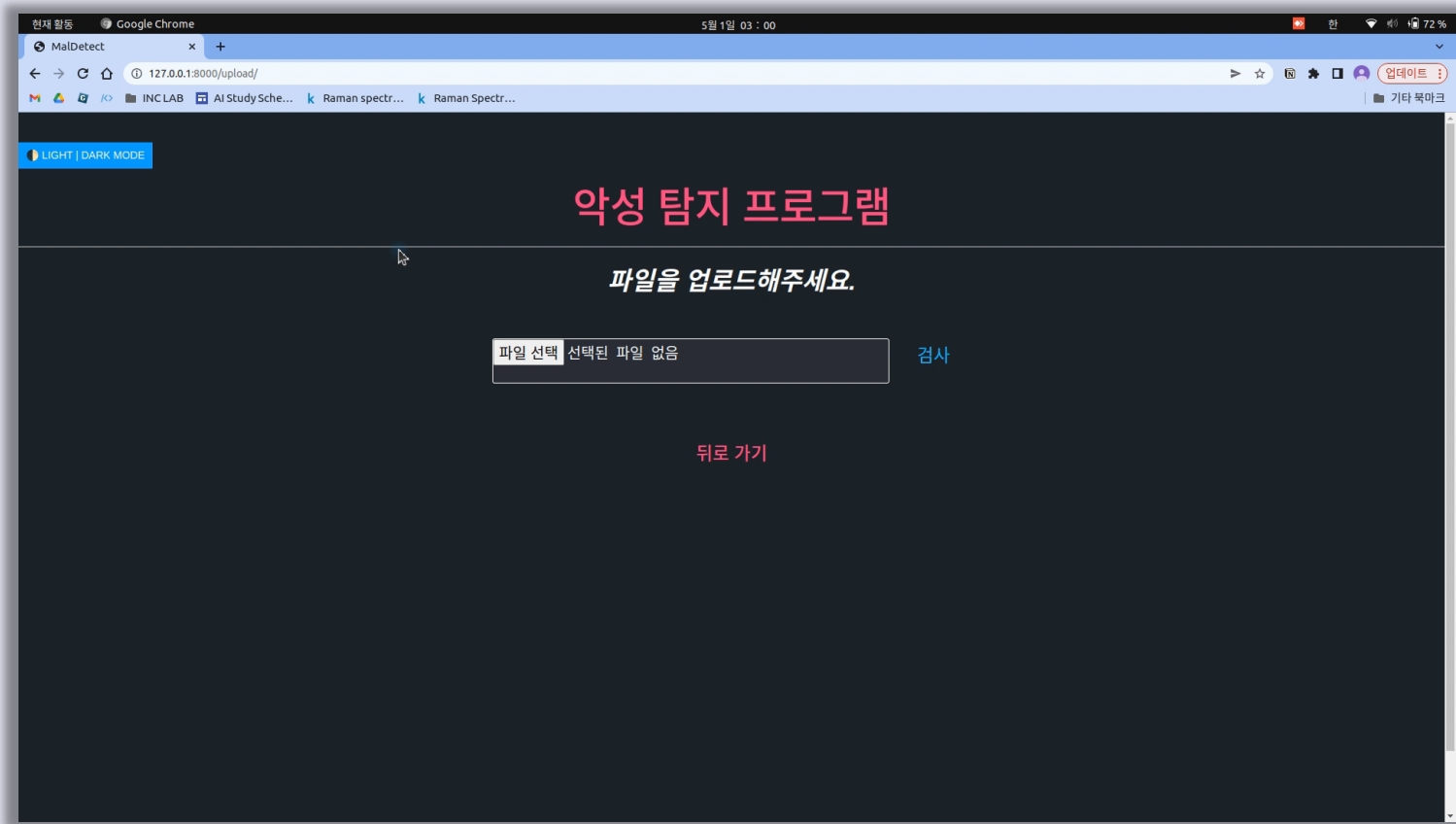
3. 악성 File → Flat PDF 변환 - 변환된 PDF

Safe-output.pdf



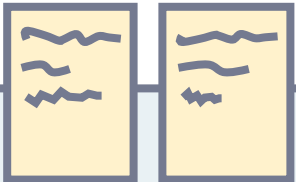
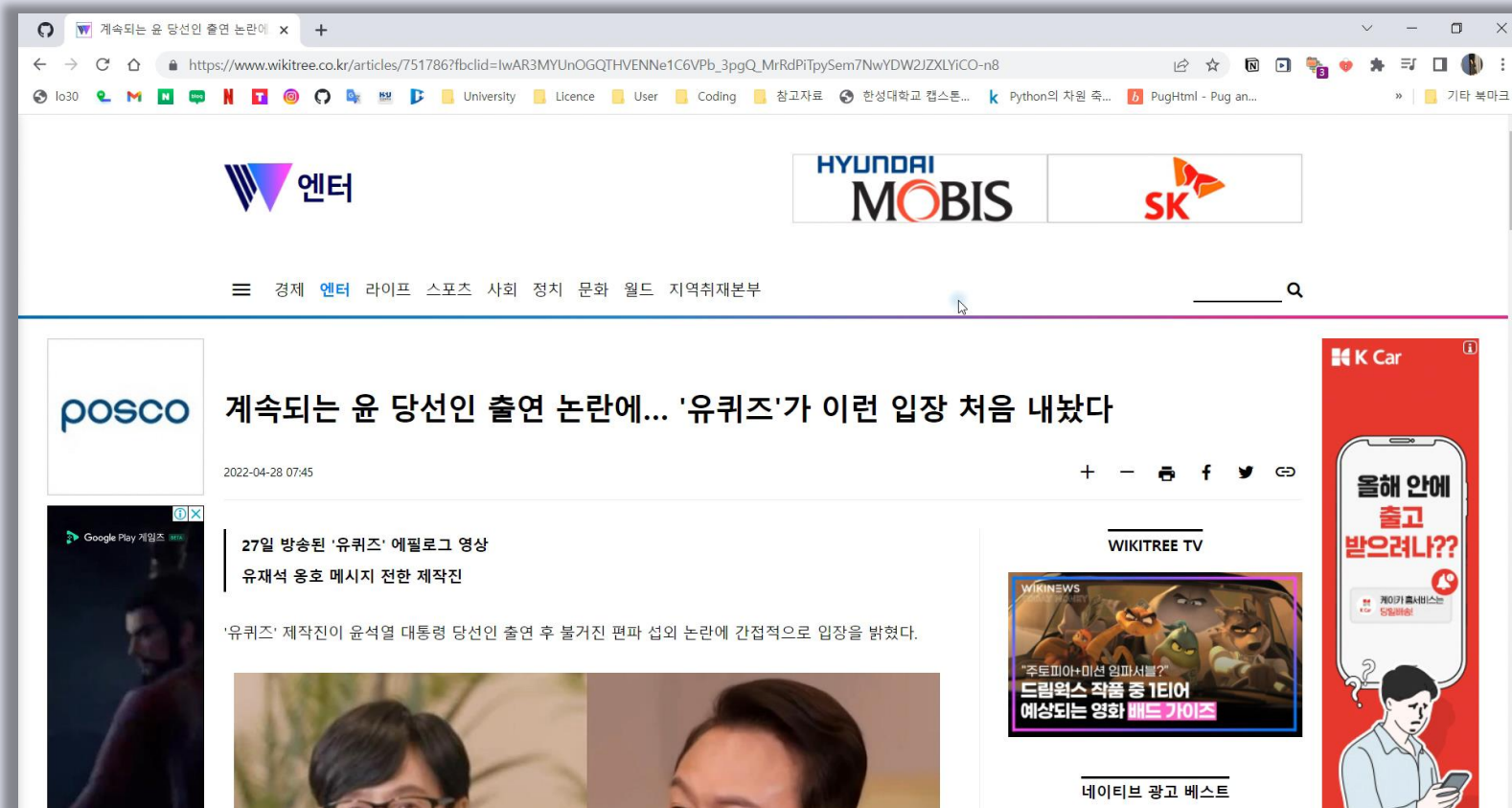
수행 내역 및 시연

3. 악성 File → Flat PDF 변환 - 변환된 PDF 감염 여부 검사



수행 내역 및 시연

4. 광고 배너 차단 - 기존 site 모습



수행 내역 및 시연

4. 광고 배너 차단

Visit the Web Page



Check URL in
Background.js



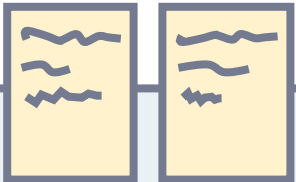
Detect Keyword or
URL in EasyList



Adblock Cover

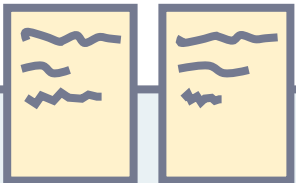
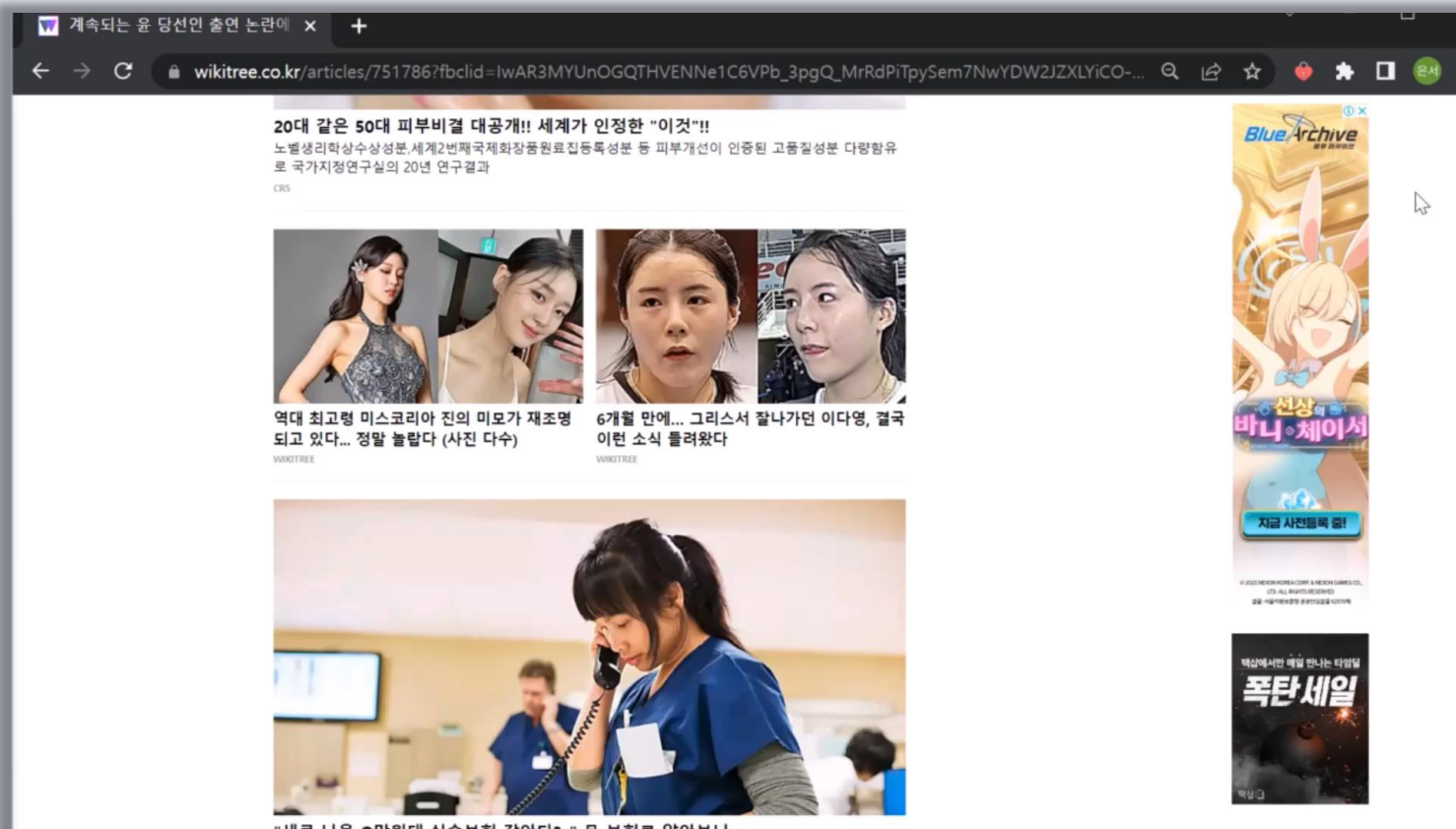
```
1 function getCurrentTabURL(callback){
2   var queryInfo = {
3     active: true,
4     currentWindow: true
5   };
6   chrome.tabs.query(queryInfo, function(tabs){
7     var tab = tabs[0];
8     var url = tab.url;
9     callback(url);
10  })
11 }
12 function renderURL(statusText){
13   document.getElementById("i_result").innerHTML = statusText;
14 }
15
16 document.addEventListener('DOMContentLoaded', function(){
17   chrome.tabs.executeScript(
18     function(result){
19       getCurrentTabURL(function(url){
20         renderURL(url);
21       });
22     }
23   )
24 });
```

```
1
2 chrome.webRequest.onBeforeRequest.addListener(
3   function(details) {
4     if(!enabled){
5       return { cancel: false };
6     }
7     console.log("I am going to block:", details.url)
8     return {cancel: true};
9   },
10  {urls: blocked_sites,
11    ["blocking"]
12  }
13 )
```



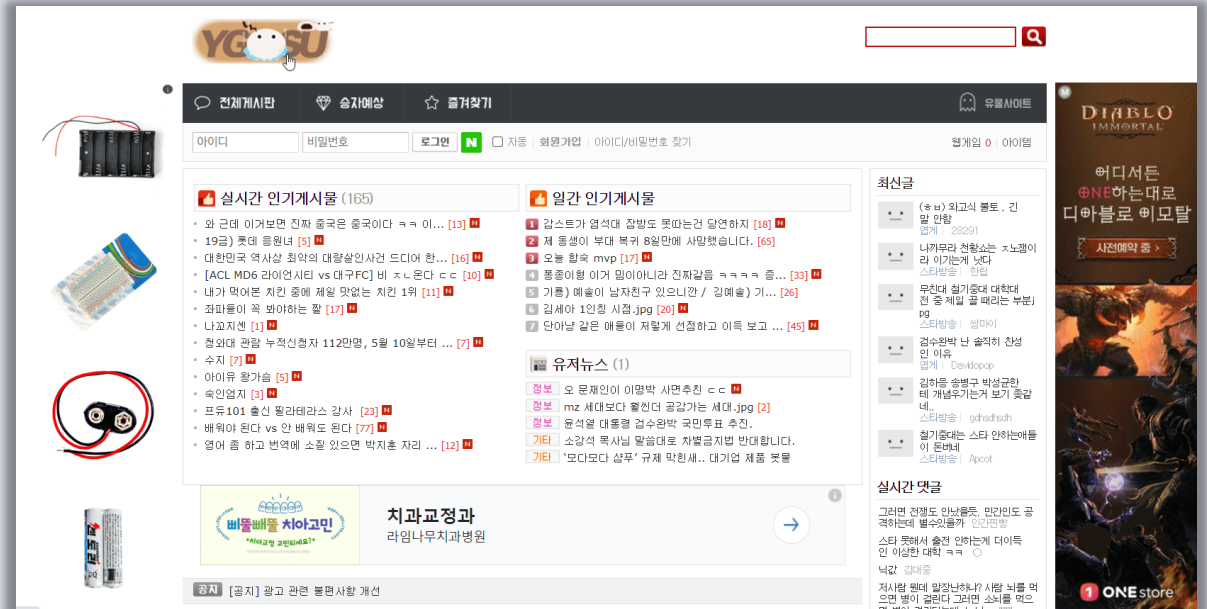
수행 내역 및 시연

4. 광고 배너 차단 - 실행

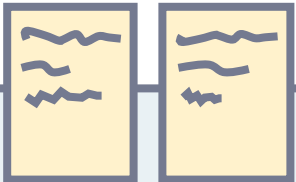


수행 내역 및 시연

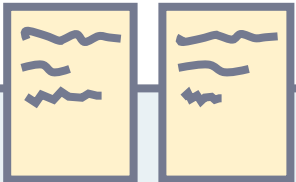
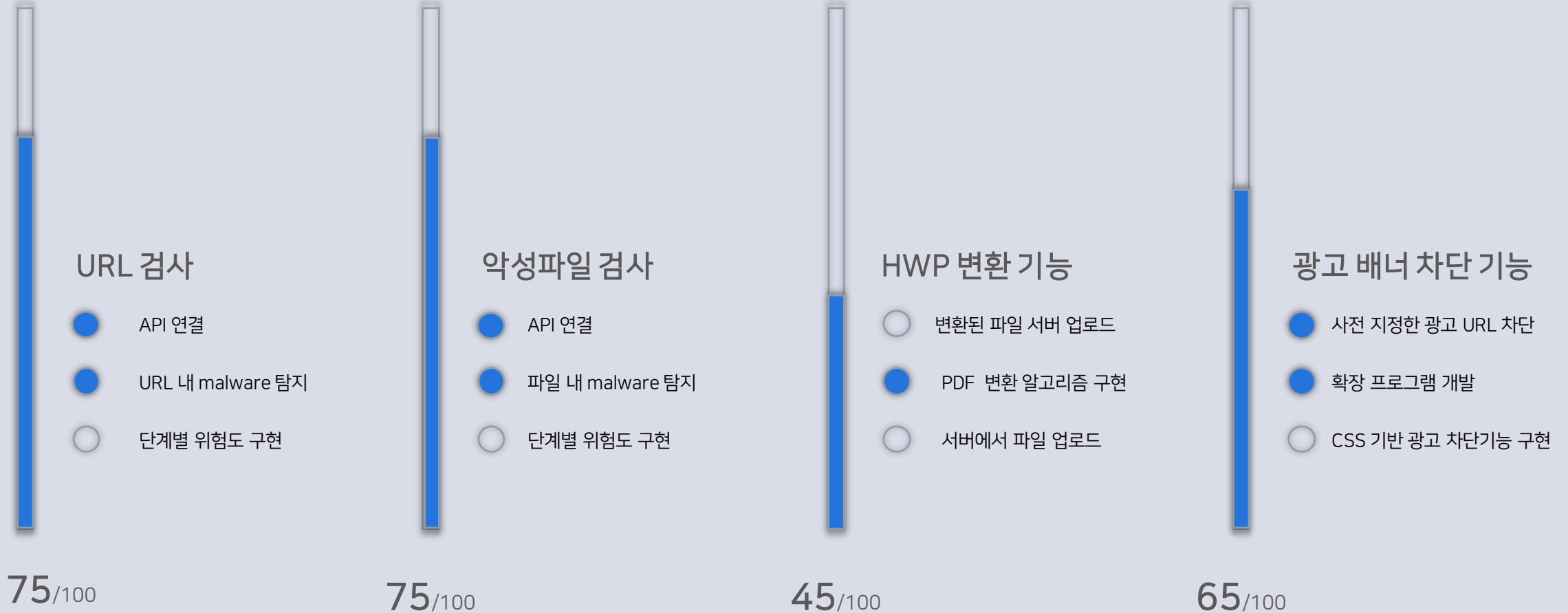
4. 광고 배너 차단 - 일부 사이트 적용 문제



- Anti-adblock이 구현된 사이트에 적용 불가 → "Shadow DOM"
- Cosmetic filtering = CSS 기반 광고 차단 방식 추가 구현 예정



개발 목표 대비 개발 진척도

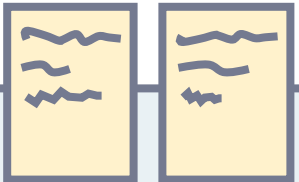



향후 개발 계획

주차	수행내용
10주차	악성 hwp 파일 ➡ PDF 변환 기능 구현
11주차	악성 hwp 파일 ➡ PDF 변환 구현 및 버그 수정
12주차	광고 배너 차단 기능 추가
13주차	최종 테스트 및 버그 수정, Web 디자인
14주차	프로젝트 완성 및 버그 수정
15주차	프로젝트 최종 발표

기존 프로그램들과 차별화

CSS 기반 광고 차단 가능하도록 추가 구현





감사합니다

2022 사이버보안 캡스톤 디자인 2조