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

프로젝트 제안 배경

infosec

SK 쉴더스

이메일을 통한 사이버 공격

공격 목적	메일 계정 탈취 (정보유출, 사칭 2차 공격)	악성코드 감염 (내부 정보 유출)	
공격 유형	피싱메일 - 포털운영진 사칭 - 파일 다운로드 위장 링크	악성파일 첨부 - 문서의 정상기능(매크로 등) 악용 - 문서 파일 위장 악성 실행파일	스피어피싱 - 특정 대상 공격 - 2단계 스피어피싱
공격 목적	사회공학적 기법 (사람의 심리 이용)	APT 공격 (지능형 지속 공격)	

〈 악성파일 확장자 정보 〉

infosec

구분	확장자 예시
실행파일	.exe .msi .bat .scr .pif .vbs .wsf 등
문서 아이콘 위장 이중 확장자	.pdf(공백).exe .pdf(공백).scr .docx(공백).exe .xlsx(공백).exe 등

〈 APT 공격 방어 방법 〉

infosec

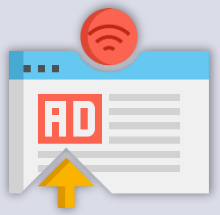
본 프로젝트의
URL 검사 및 파일 검사 이용 시
해결

- 출처가 불분명한 의심스러운 메일 내 링크 클릭 및 첨부파일 열람 주의
- 메일 본문 내 링크를 클릭하여 계정정보 입력 주의
- 메일 첨부파일 실행 시 파일 확장명 확인
- 메일 첨부문서 열람 시 매크로, 개체연결삽입(OLE) 허용 주의
- 백신 프로그램 최신 업데이트 및 실시간 감시 기능 활성화
- 메일 로그인 시 2단계 인증 적용
- 트위터, 인스타그램 등 SNS에 개인정보 노출 주의

핵심 기능

파일변환

- 다양한 문서 확장자를 가진 파일 지원
- 감염 파일의 악성코드 삭제
- 문서 내용 무 손상 상태 반환



광고 배너 차단 기능

- 광고 배너를 가려 광고 없이 사이트 이용 가능

파일 검사 기능

- 감염 파일이 보유한 악성 코드 데이터 제공

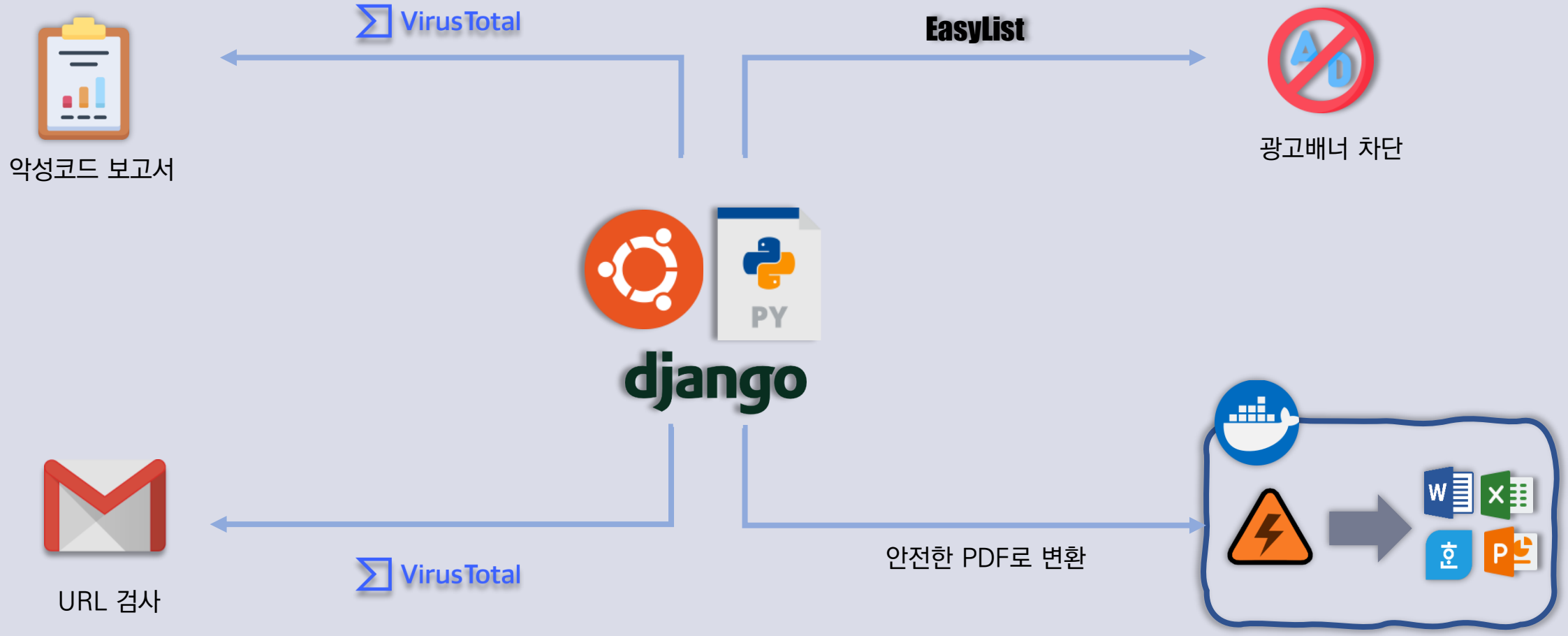
URL 검사 기능

- URL 스캔 후 탐지된 malware 보고

관련 유사 프로젝트 조사 및 차이점 분석

		HWP	Open Document Format	안전 파일 변환	악성 코드 유무	편리성	URL 검사
		X	O	O	X	X	
유료	SaniTOX 콘텐츠 악성코드 무해화(CDR) 서비스	O	X	O	O	X	
유료	McAfee WebAdvisor				O	O	X
	캡스톤 결과물	O	O	O	O	O	O

프로그램 개요


VirusTotal
EasyList
악성코드 보고서
광고배너 차단
PY
django
URL 검사
VirusTotal
안전한 PDF로 변환

개발에 사용된 기술 소개

1. Python

- Dangerzone 한글 확장자 추가
- Web 개발 (Django)

2. Ubuntu Server 20.04

3. Chrome Extension

- Javascript, HTML을 사용

4. 파일 검사 및 URL 검사

- VirusTotal OpenAPI

5. 악성 파일 변환

- CDR 기법을 적용한 DangerZone Opensource
- Docker

6. 광고 배너 차단

- Chrome Extension API
  - ✓ onBeforeRequest
- EasyList
- Membranes
- Shawdow DOM bypassing

수행 내역 및 시연

1. URL scan

URL

↓

Send Link To Server

↓

Scan Link with VirusTotal

↓

Check Malware in URL → Scan Result

↑

Send it to Server

↑

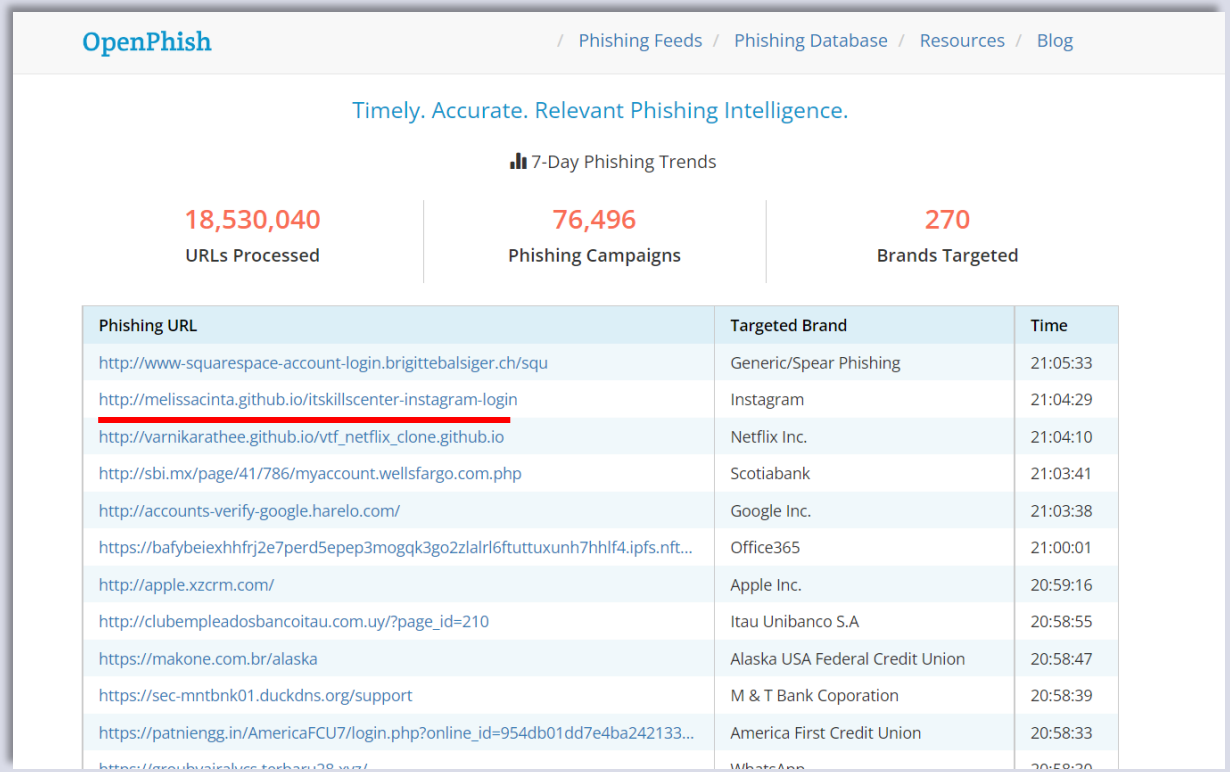
Show Report

```python
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 = "78a4f8a70dc6f5adbf7a29b28f899746a39b009784f43579e5867acc3b45c5a1"
    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)
```

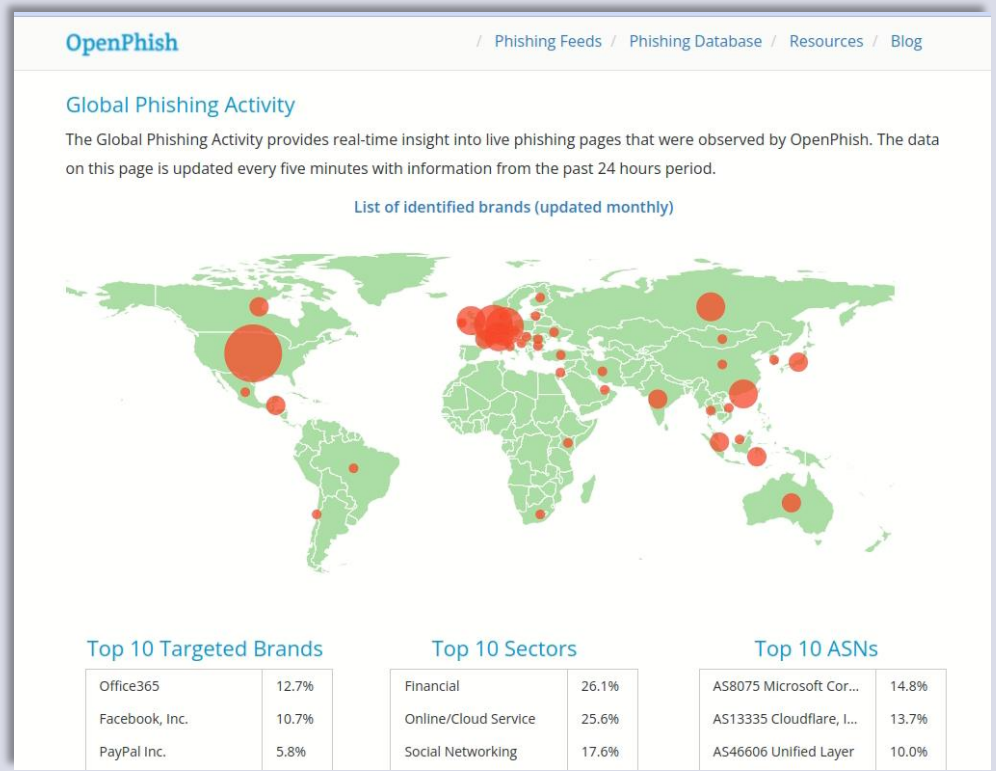
```python
try:
    imprint = json.loads(output)['positives']
except:
    return render(request, 'urlerror.html')
print(imprint)
if imprint==1:

    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

1. URL scan



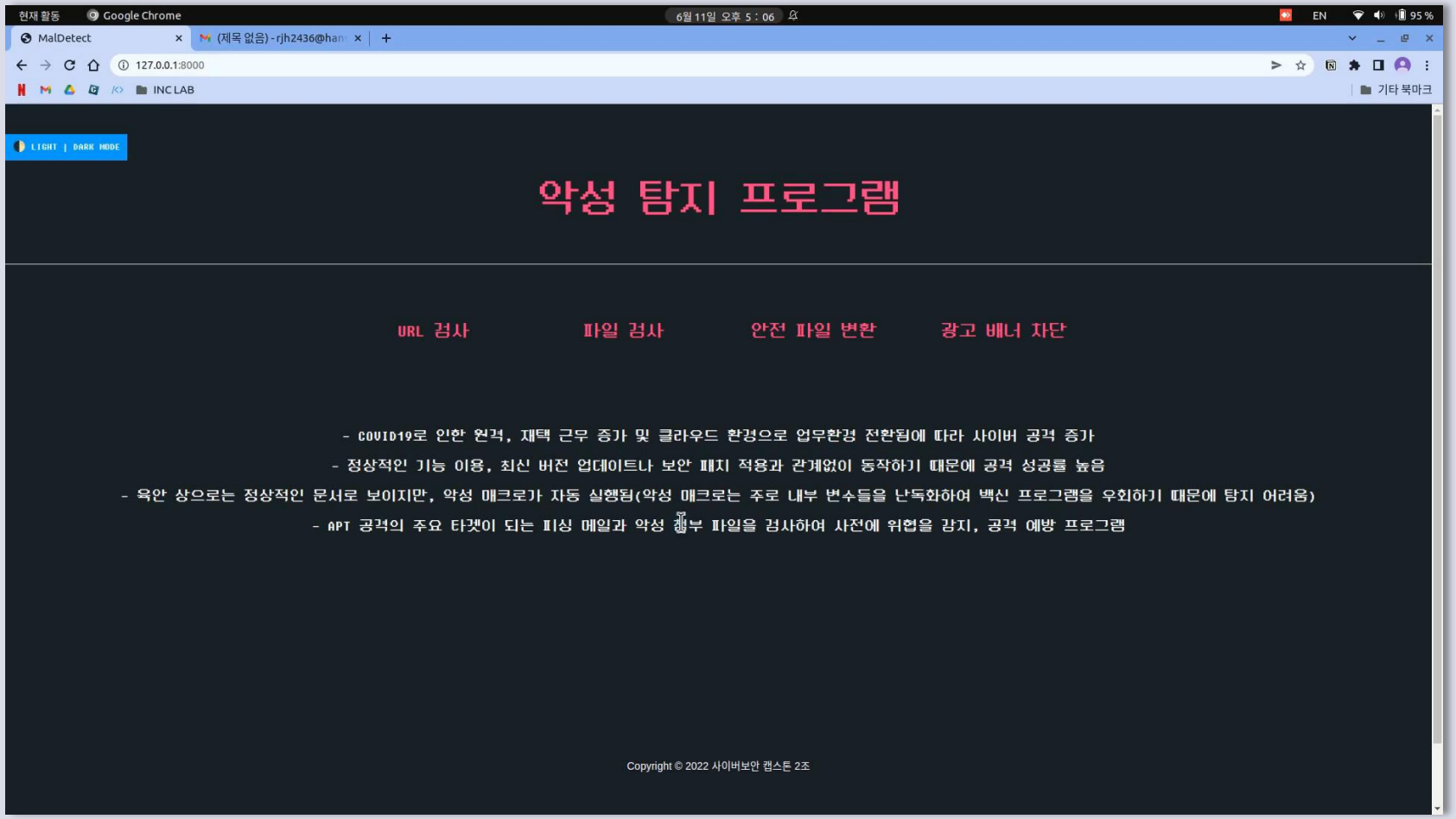
Phishing URL	Targeted Brand	Time
http://www-squarespace-account-login.brigittebalsiger.ch/squ	Generic/Spear Phishing	21:05:33
http://melissacinta.github.io/itskillscenter-instagram-login	Instagram	21:04:29
http://varnikarathee.github.io/vtf_netflix_clone.github.io	Netflix Inc.	21:04:10
http://sbi.mx/page/41/786/myaccount.wellsfargo.com.php	Scotiabank	21:03:41
http://accounts-verify-google.harelo.com/	Google Inc.	21:03:38
https://bafybeiexhhfrj2e7perd5epep3mogqk3go2zlalrl6ftuttuxunh7hhlf4.ipfs.nft...	Office365	21:00:01
http://apple.xzcrm.com/	Apple Inc.	20:59:16
http://clubempleadosbancoitau.com.uy/?page_id=210	Itau Unibanco S.A	20:58:55
https://makone.com.br/alaska	Alaska USA Federal Credit Union	20:58:47
https://sec-mntbnk01.duckdns.org/support	M & T Bank Coporation	20:58:39
https://patniengg.in/AmericaFCU7/login.php?online_id=954db01dd7e4ba242133...	America First Credit Union	20:58:33



피싱 사이트, 악성코드 배포 사이트 목록을 실시간으로 업데이트 하는 사이트

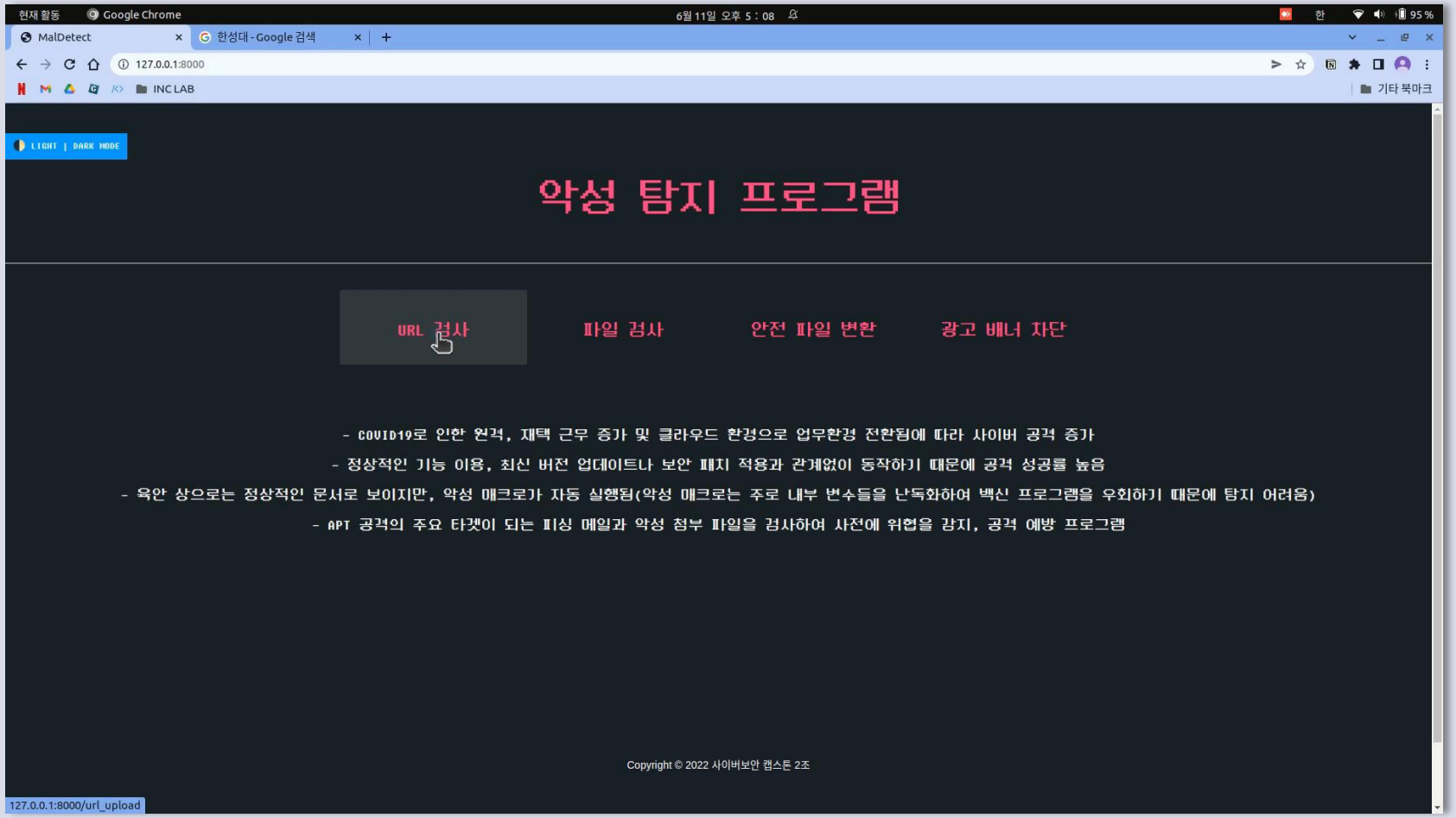
수행 내역 및 시연

1. URL scan – 위험 요소가 존재하는 URL



수행 내역 및 시연

1. URL scan – 안전한 URL



수행 내역 및 시연

2. File scan

File

↓

Send File To Server

↓

Scan File with VirusTotal

↓

Check Malware in File → Scan Result

↑

Send it to Server

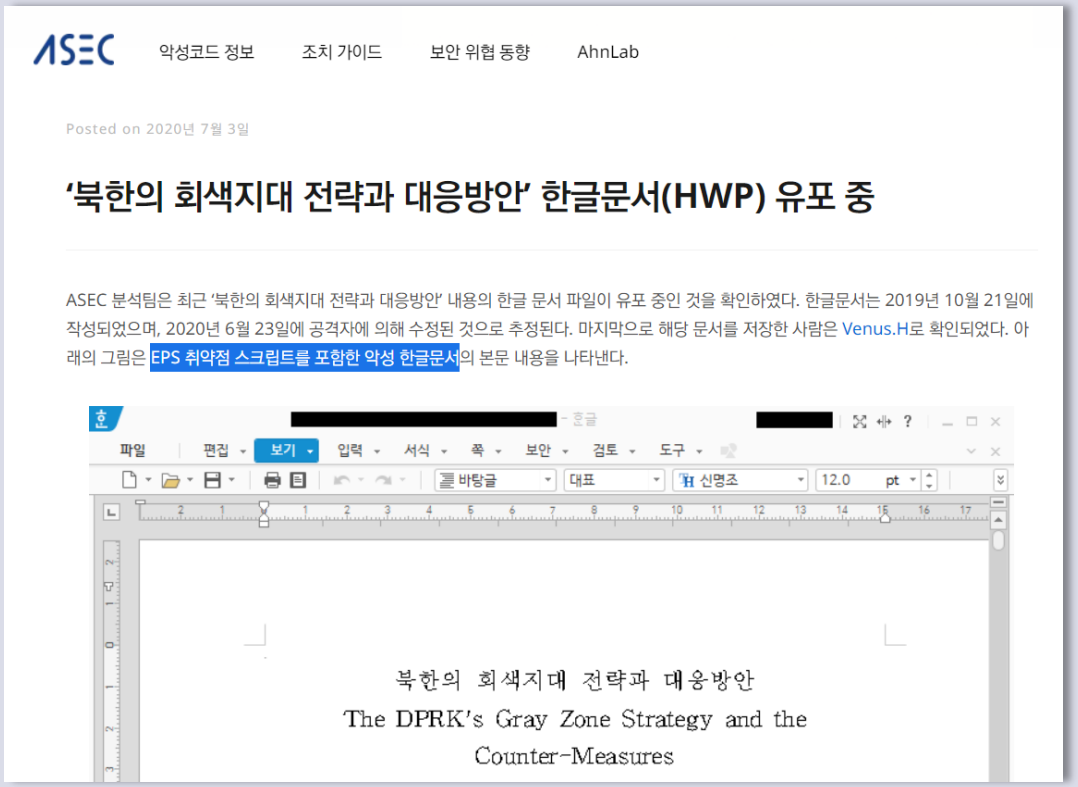
↑

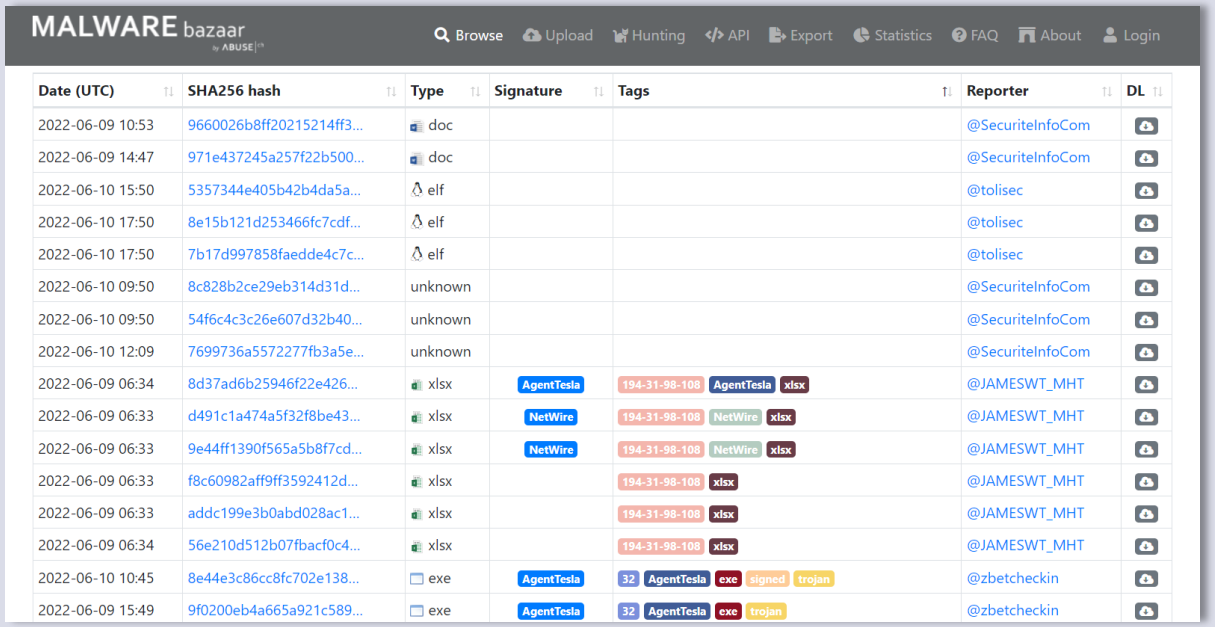
Show Report

```python
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



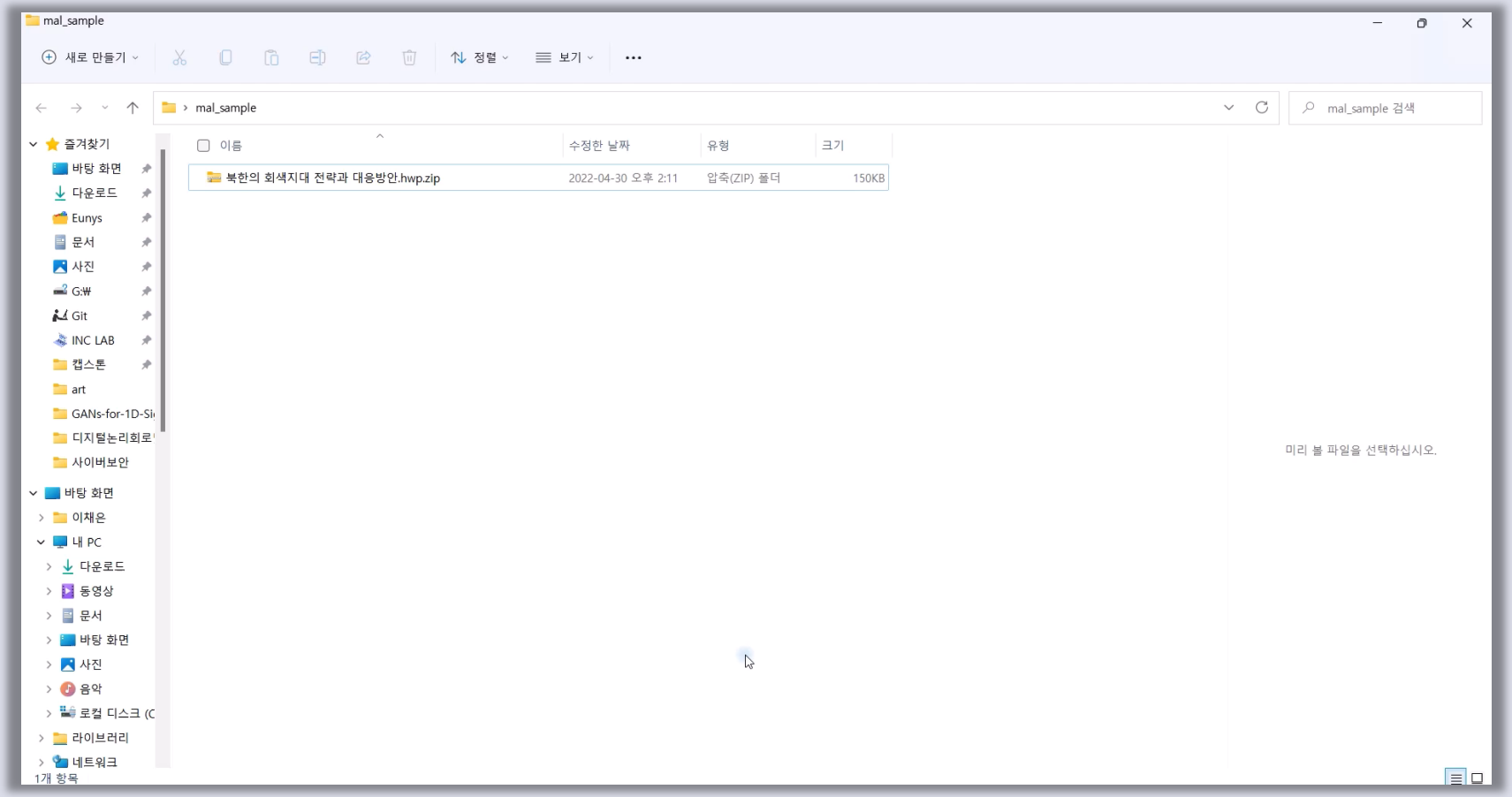


Date (UTC)	SHA256 hash	Type	Signature	Tags	Reporter	DL
2022-06-09 10:53	9660026b8ff20215214ff3...	doc			@SecuriteInfoCom	
2022-06-09 14:47	971e437245a257f22b500...	doc			@SecuriteInfoCom	
2022-06-10 15:50	5357344e405b42b4da5a...	elf			@tolisec	
2022-06-10 17:50	8e15b121d253466fc7cdf...	elf			@tolisec	
2022-06-10 17:50	7b17d997858faedde4c7c...	elf			@tolisec	
2022-06-10 09:50	8c828b2ce29eb314d31d...	unknown			@SecuriteInfoCom	
2022-06-10 09:50	54f6c4c3c26e607d32b40...	unknown			@SecuriteInfoCom	
2022-06-10 12:09	7699736a5572277fb3a5e...	unknown			@SecuriteInfoCom	
2022-06-09 06:34	8d37ad6b25946f22e426...	xlsx	AgentTesla	194-31-98-108 AgentTesla xlsx	@JAMESWT_MHT	
2022-06-09 06:33	d491c1a474a5f32f8be43...	xlsx	NetWire	194-31-98-108 NetWire xlsx	@JAMESWT_MHT	
2022-06-09 06:33	9e44ff1390f565a5b8f7cd...	xlsx	NetWire	194-31-98-108 NetWire xlsx	@JAMESWT_MHT	
2022-06-09 06:33	f8c60982aff9ff3592412d...	xlsx		194-31-98-108 xlsx	@JAMESWT_MHT	
2022-06-09 06:33	addc199e3b0abd028ac1...	xlsx		194-31-98-108 xlsx	@JAMESWT_MHT	
2022-06-09 06:34	56e210d512b07fbacf0c4...	xlsx		194-31-98-108 xlsx	@JAMESWT_MHT	
2022-06-10 10:45	8e44e3c86cc8fc702e138...	exe	AgentTesla	32 AgentTesla exe signed trojan	@zbetcheckin	
2022-06-09 15:49	9f0200eb4a665a921c589...	exe	AgentTesla	32 AgentTesla exe trojan	@zbetcheckin	

- 북한의 회색지대 전략과 대응방안.hwp
- 확장자 xlsx, doc, exe, etc..

수행 내역 및 시연

2. File scan

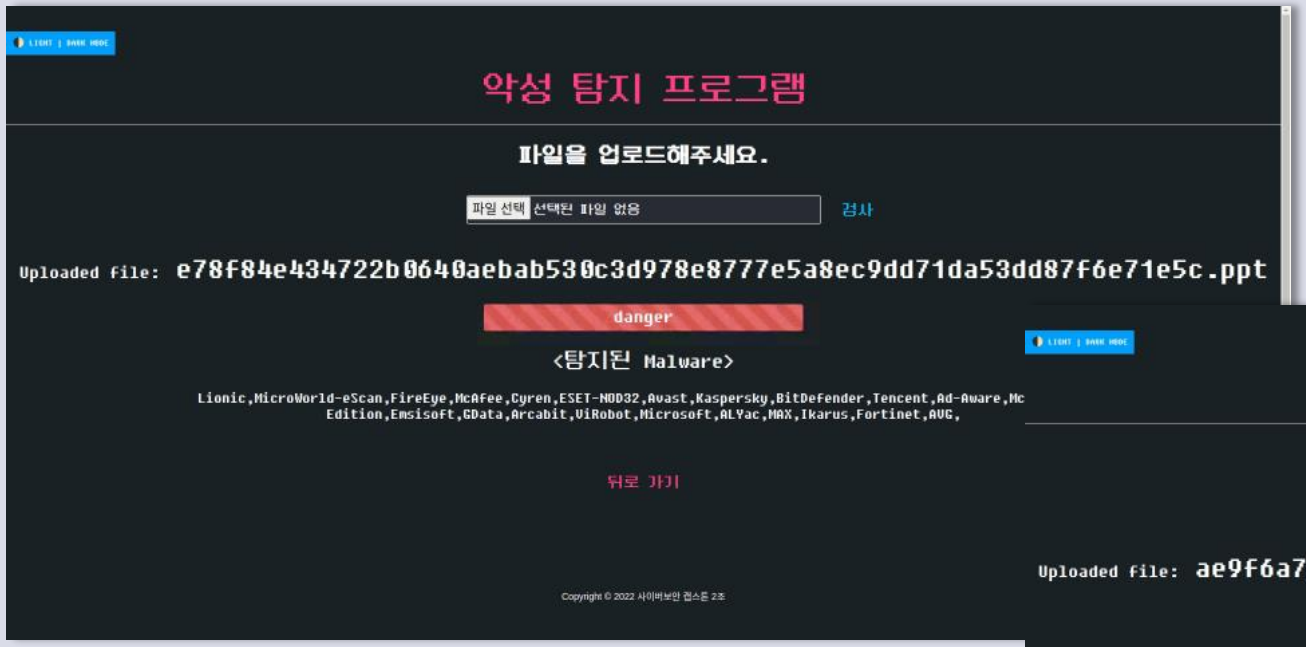


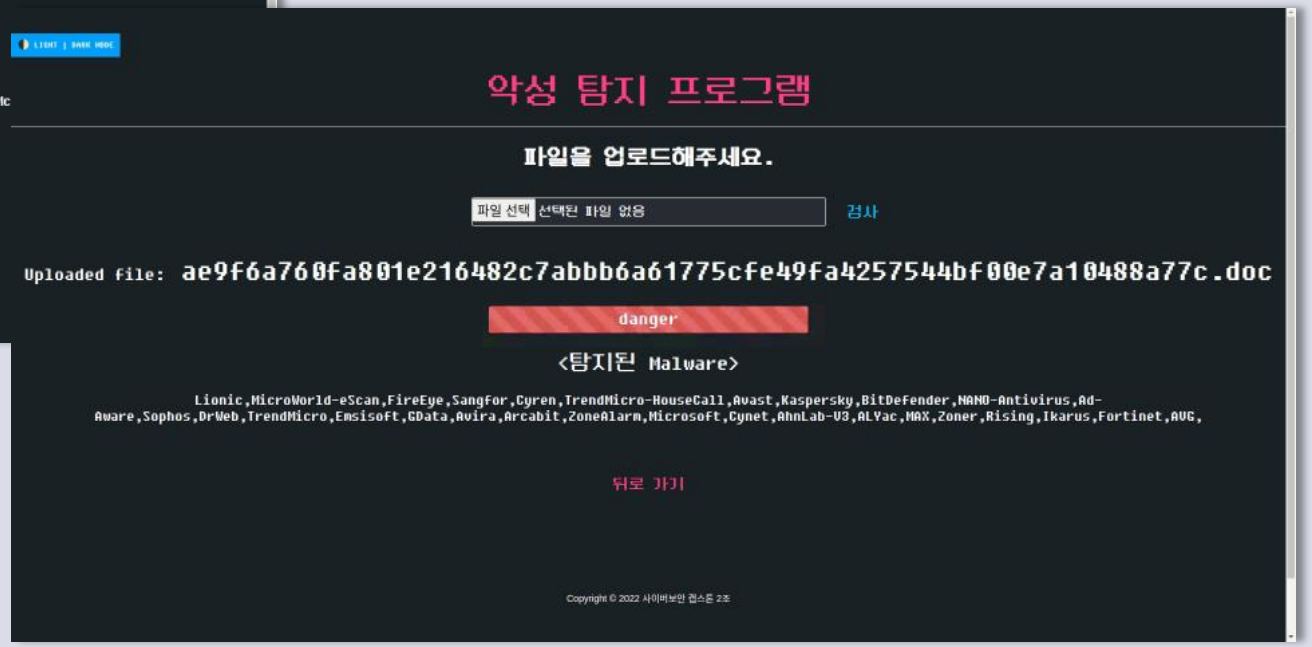
수행 내역 및 시연

2. File scan – 악성 파일 검사

수행 내역 및 시연

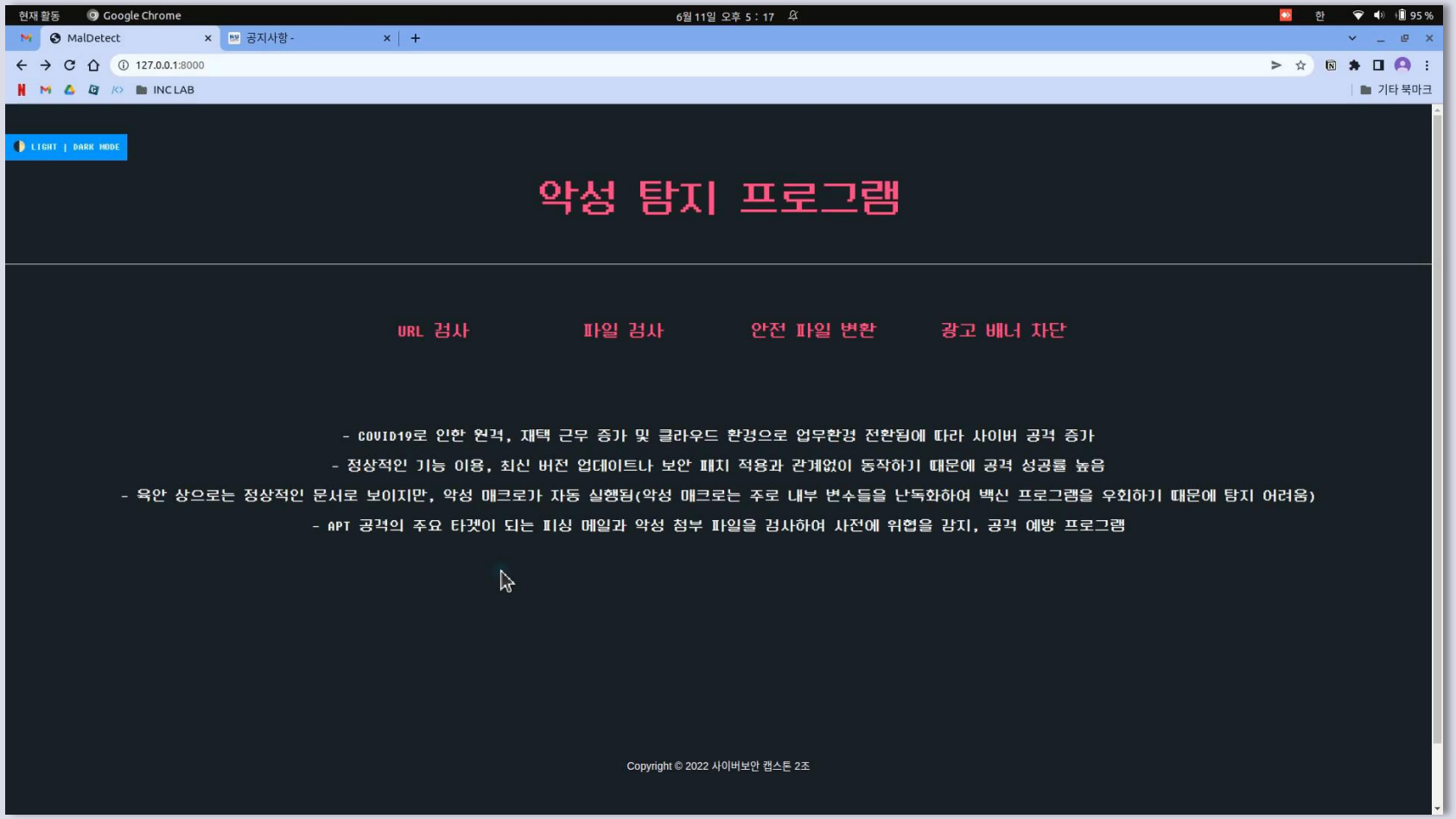
2. File scan – 악성 파일 검사





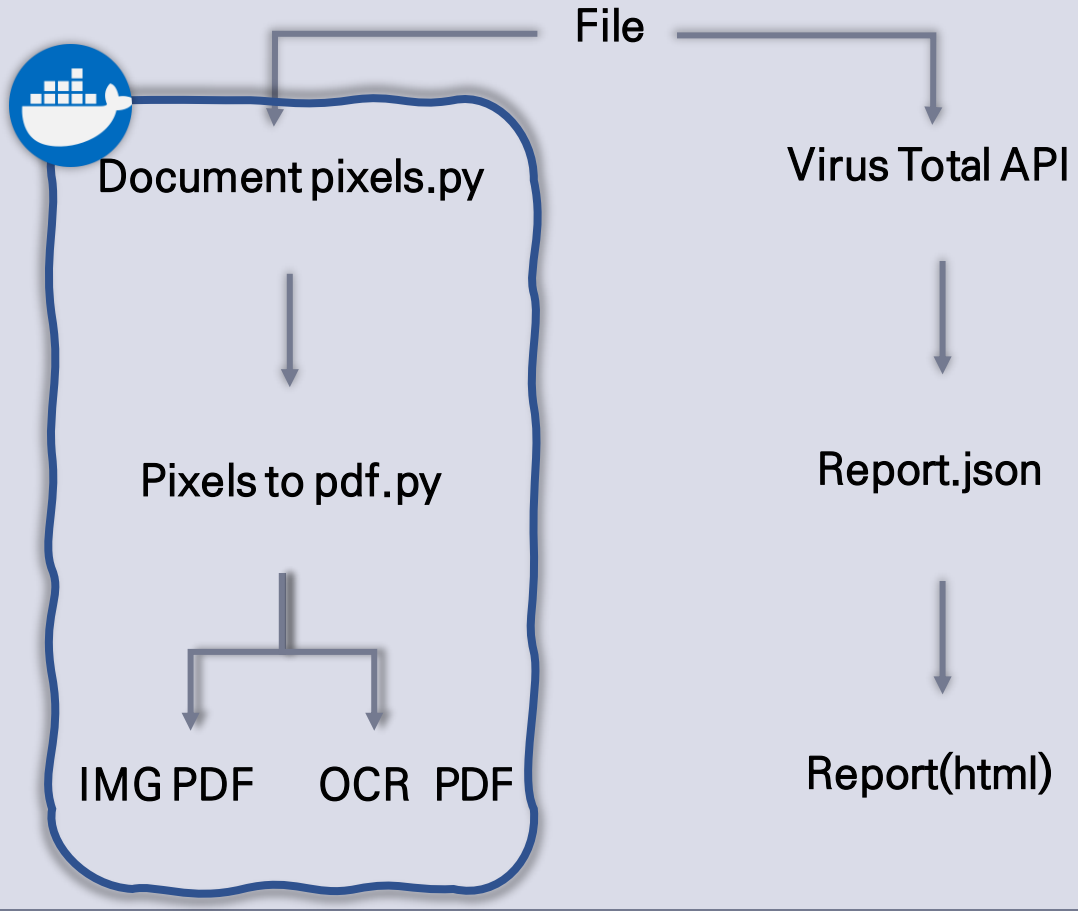
수행 내역 및 시연

2. File scan – 안전한 파일 검사



수행 내역 및 시연

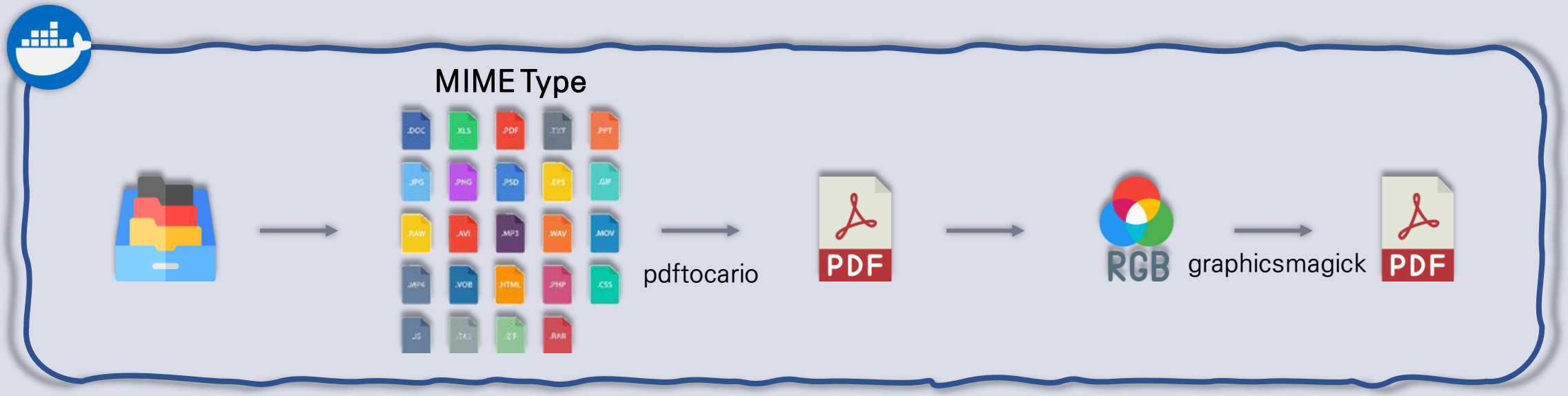
3. 악성 File → Flat PDF 변환




```python
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

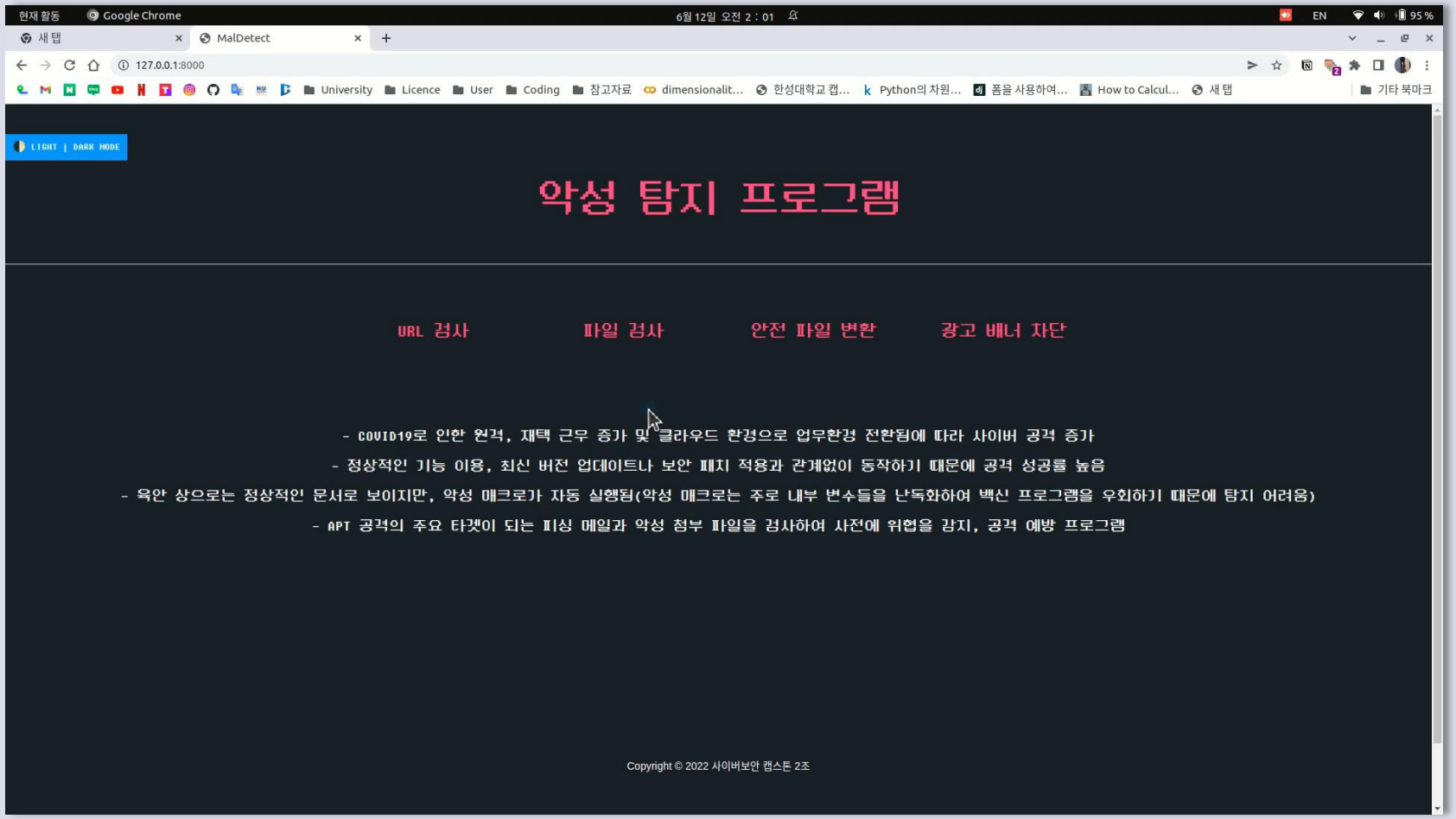


추가 사용 모듈

- Docker에서 HWP 파일 확장자 사용 : pyhwp, wkhtmltopdf, 한글 언어팩 사용
- MIME Type : libreofiice, pwhwp, graphicmagick 사용
- OCR (searchable PDF) : tesseract

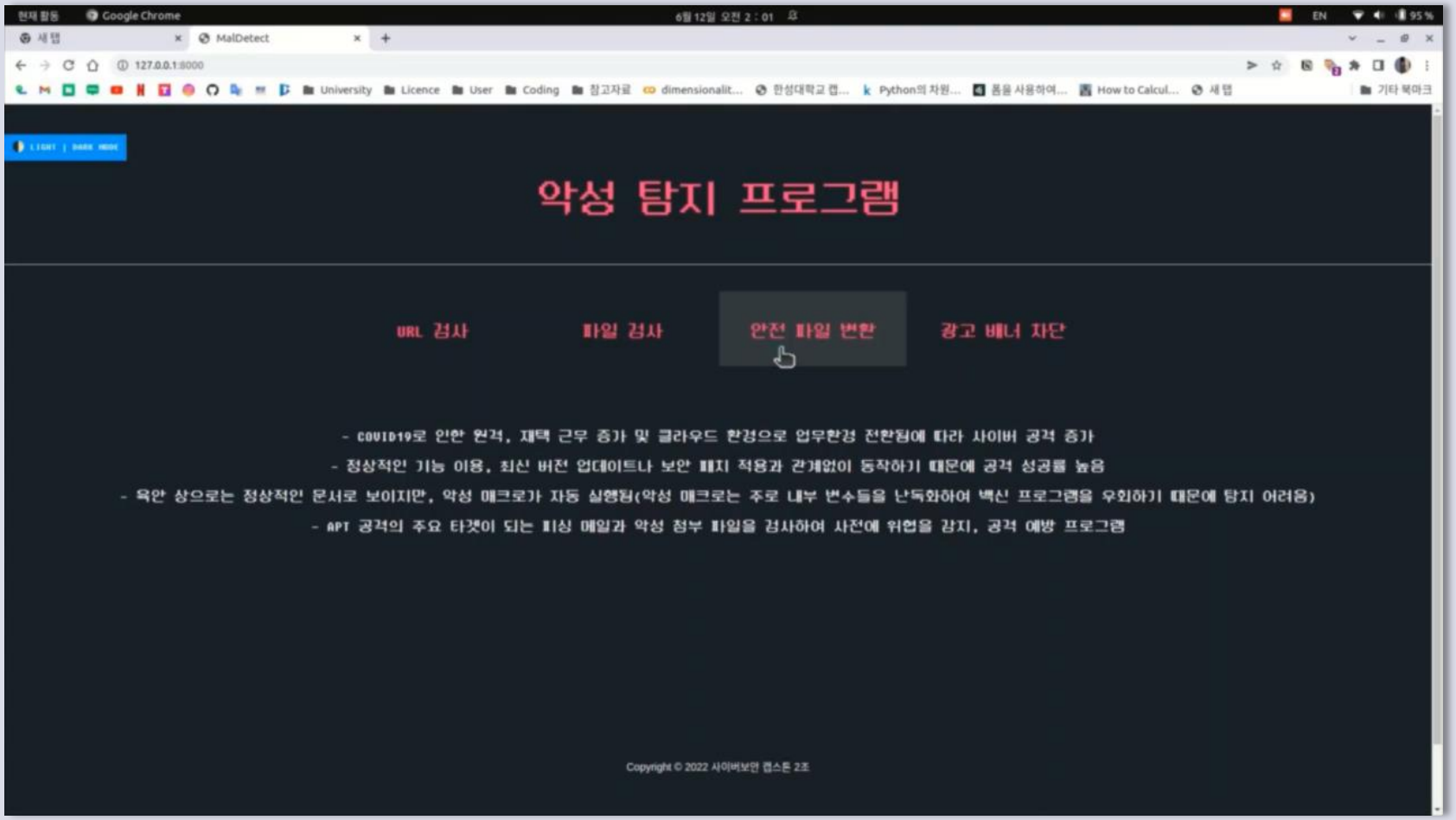
수행 내역 및 시연

3. 악성 File → Flat PDF 변환



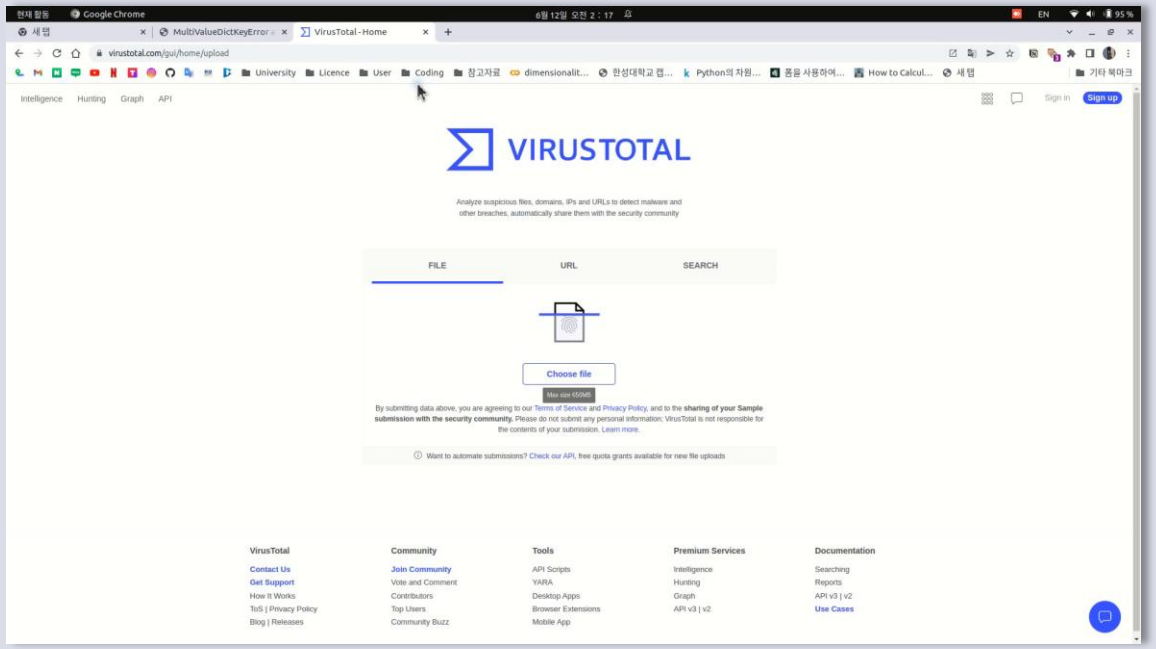
수행 내역 및 시연

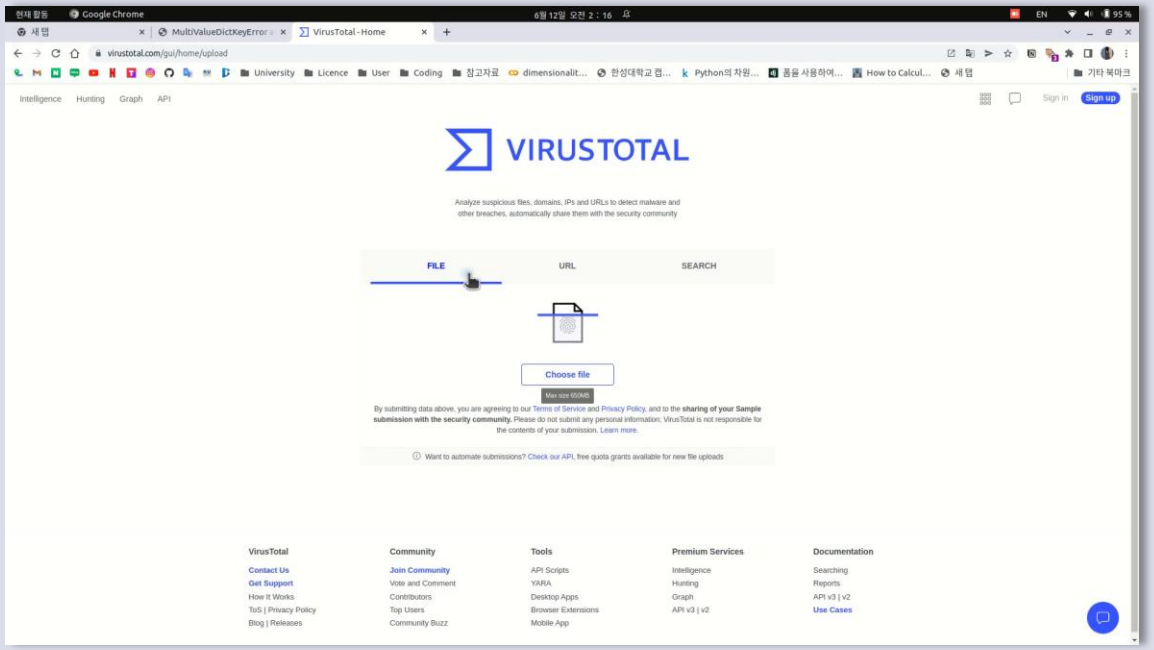
3. 악성 File → Searchable PDF 변환



수행 내역 및 시연

3. 악성 File → 변환된 PDF 감염 여부 확인





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

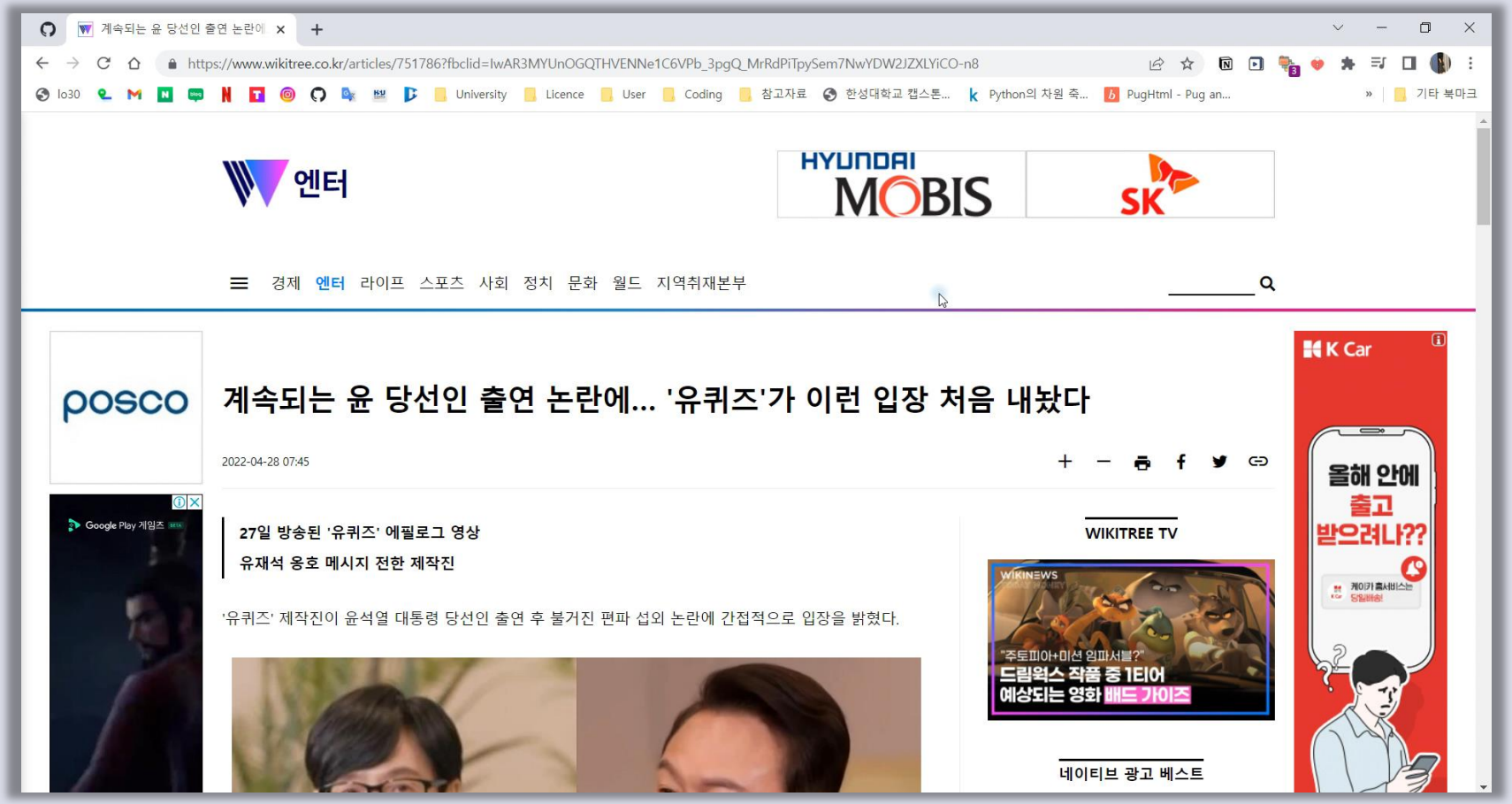
```
function getCurrentTabURL(callback){
  var queryInfo = {
    active: true,
    currentWindow: true
  };
  chrome.tabs.query(queryInfo, function(tabs){
    var tab = tabs[0];
    var url = tab.url;
    callback(url);
  })
}
function renderURL(statusText){
  document.getElementById("i_result").innerHTML = statusText;
}

document.addEventListener('DOMContentLoaded', function(){
  chrome.tabs.executeScript(
    function(result){
        getCurrentTabURL(function(url){
        renderURL(url);
    });
    }
  )
});
```

```
chrome.webRequest.onBeforeRequest.addListener(
    function(details) {
        if(!enabled){
          return { cancel: false };
        }
        console.log("I am going to block:", details.url)
        return {cancel: true};
    },
    {urls: blocked_sites},
    ["blocking"]
```

```
//css 기반 광고 차단
var a;
let b = new Uint8Array(32);
crypto.getRandomValues(b);
a = String.fromCharCode(...b.map(function(d) { return (26 * d >> 8) + 97 }));
chrome.runtime.sendMessage(a);
const c = document.createElement("script");
c.textContent = "(function(){'use strict';var f=Object.defineProperty,g=Object.getOwnPr
var ka=h.bind(q.push),la=h.bind(q.indexOf),r=h.bind(q.splice),ma=q.values,u=Error.captur
const x=Element.prototype;var Ga=x.hasAttribute,Ha=h.bind(x.setAttribute),Ia=h.bind(x.r
var Ta=w(Sa,'contentWindow'),Ua=w(Sa,'contentDocument'),Va=w(Sa,'src'),Wa=w(window,'leng
function w(a,b){return g(a,b).get};const hb=Node.ELEMENT_NODE;function C(a,b){return ta
class M extends null{constructor(a){const b=ba(M.prototype);b.g=a;return b}}function N(a
const sb=N(J.construct,function(a,b,c,d){return a.M(b,c,d)}),tb=N(lb,function(a,b,c,d){
function Ab(a,b,c){a=new rb(a,Ab,jb);try{var d=this.g;Bb=!0;d.H.g.delete(d.L);try{K(a,b
function Gb(a,b,c){f(b,c,g(a,c))}function S(a,b,c){a.hasOwnProperty(b)&&(a[b]=R(a[b],c)
function U(a,b){let c=C(a.i,b);if(F(c)){l(ya,b,['load',a.o]);try{let m=Ta.call(b);if('al
function Kb(a,b){a=a.B;for(let c=0,d=a.length;c<d;c++)a[c](b)}\n\
var Lb=class{constructor(a,b){this.g=a;this.C=b;this.B=[];this.o=fa.call(this.o,this);t
0;c<b;c++)U(this,this.I[c])}this.v=a}}};const Nb=\"623a14b8-771c-45f8-b0ce-d5fb4943ac12
const Vb=(a,b,c)=>{(F(b)||null===b)&&(b=window);var d=C(Pb,b);if(!d||0===c.length)return
function ac(a,b){if(b=a=C(a.g,b)){b=a;var c=A(b,Fa)?A(b,Aa):void 0;F(c)||c===document?b=
wc=(a,b,c)=>{a=K(a,b,c);if(va(qc.g,b)){let {u:d,m:e}=C(qc,b);'clip-path'===A(d,vc).valu
const Mc=(a,b,c)=>{a=K(a,b,c);b=la(a,P);-1!==b&&r(a,b,1);return a},Nc=(a,b,c)=>{a=K(a,b
function O(a){let b=a.h[P];b||(a.l=-1);return b}function Rc(a,b){var c=O(a);if(c){a:{var
const Uc=(a,b,c)=>{const d=C(Y,b);if(!d)return K(a,b,c);b=d.h;a=K(a,b,c);O(d)&&a--;retur
sb;c=new I(b,c);c=Fb(b,c);b.prototype.constructor=c;a.MutationObserver=c;b=a.EventTarge
lc(b,Lc,Kc);T(b,'length',Uc);S(b,'item',Vc);S(b,ja,Wc);b=a.Element.prototype;T(b,'inner
";
(document.head || document.documentElement).appendChild(c).remove();
```

수행 내역 및 시연

4. 광고 배너 차단 – 기존 Site 모습



수행 내역 및 시연

4. 광고 배너 차단 – 실행

수행 내역 및 시연

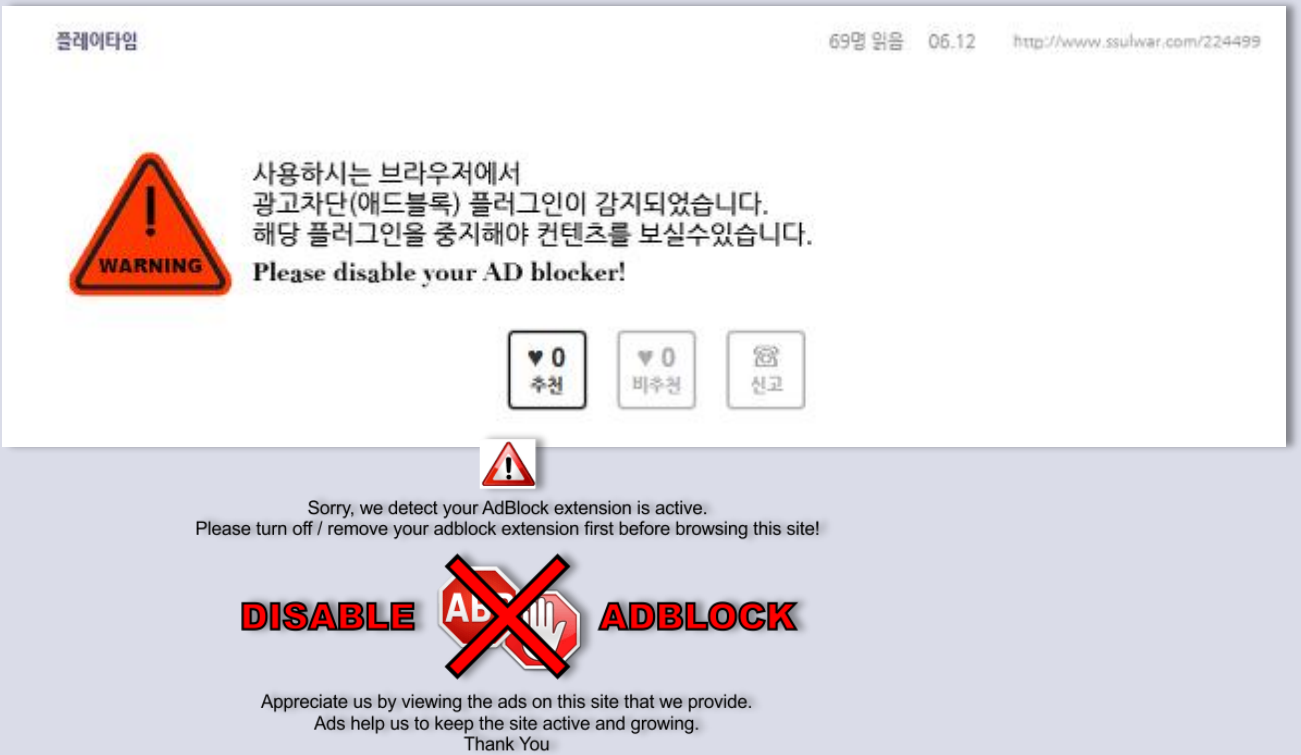
4. 광고 배너 차단 – 일부 사이트 적용 문제

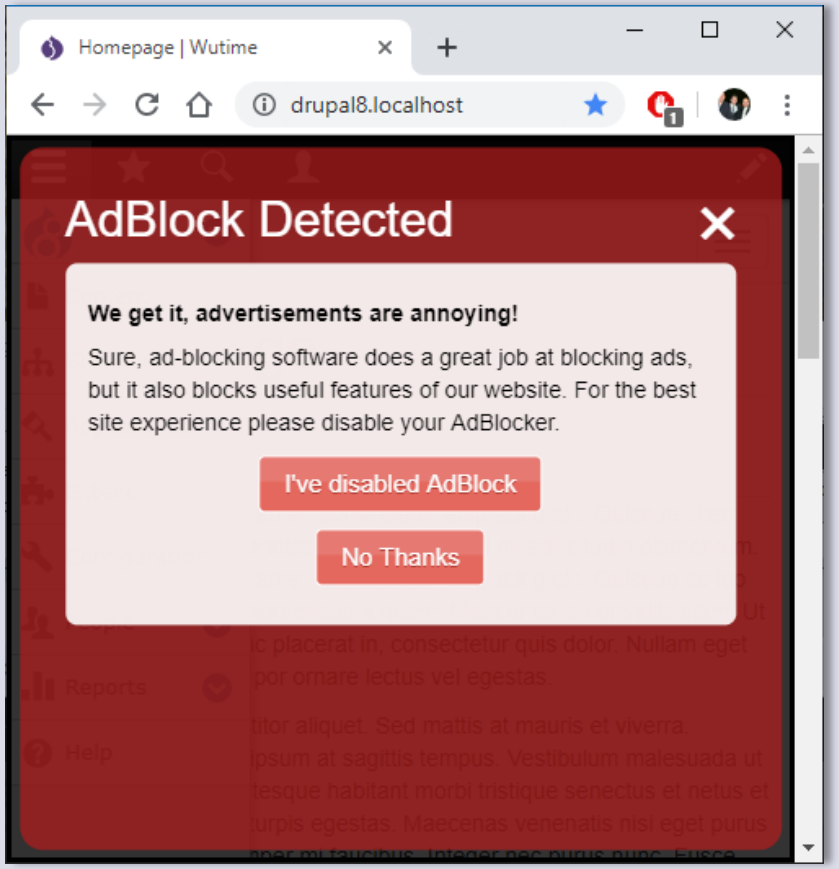
✓ ad-reinsertion(광고 차단 우회) 적용 사례

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

수행 내역 및 시연

4. 광고 배너 차단



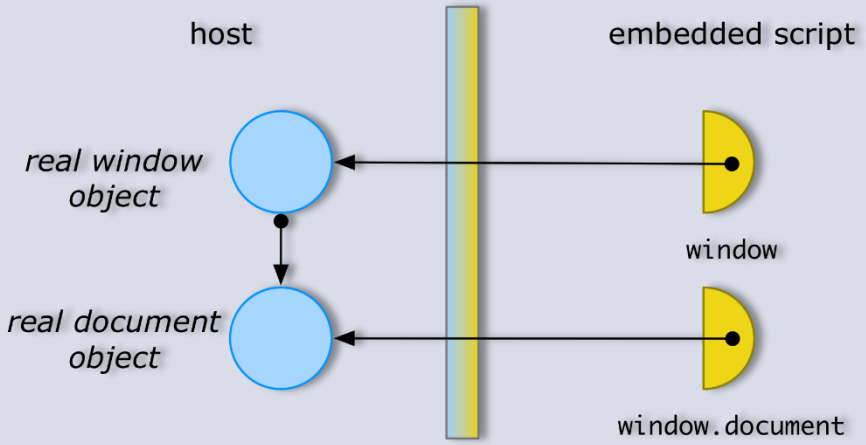


〈adBlock 차단 프로그램으로 이용이 불가한 사례〉

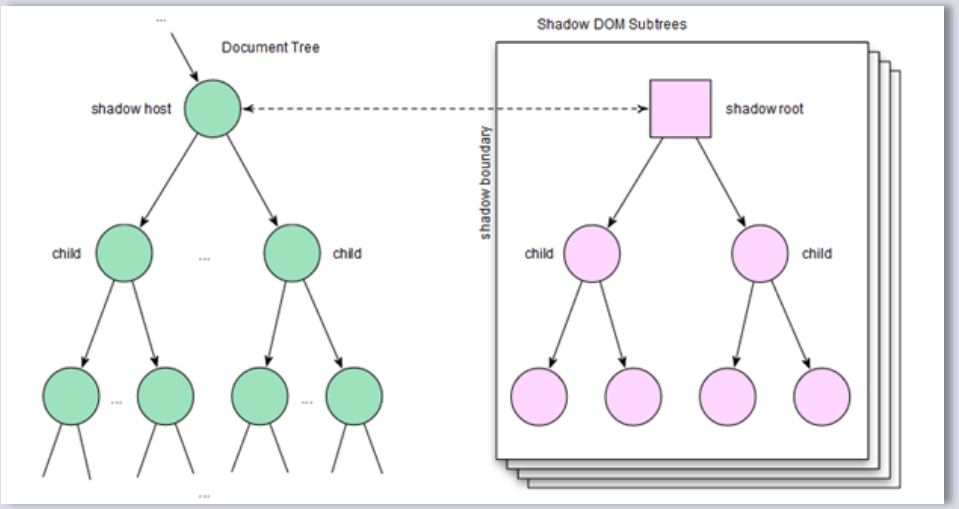
수행 내역 및 시연

4. 광고 배너 차단 – 일부 사이트 적용 문제 해결 방법

〈Membrane〉



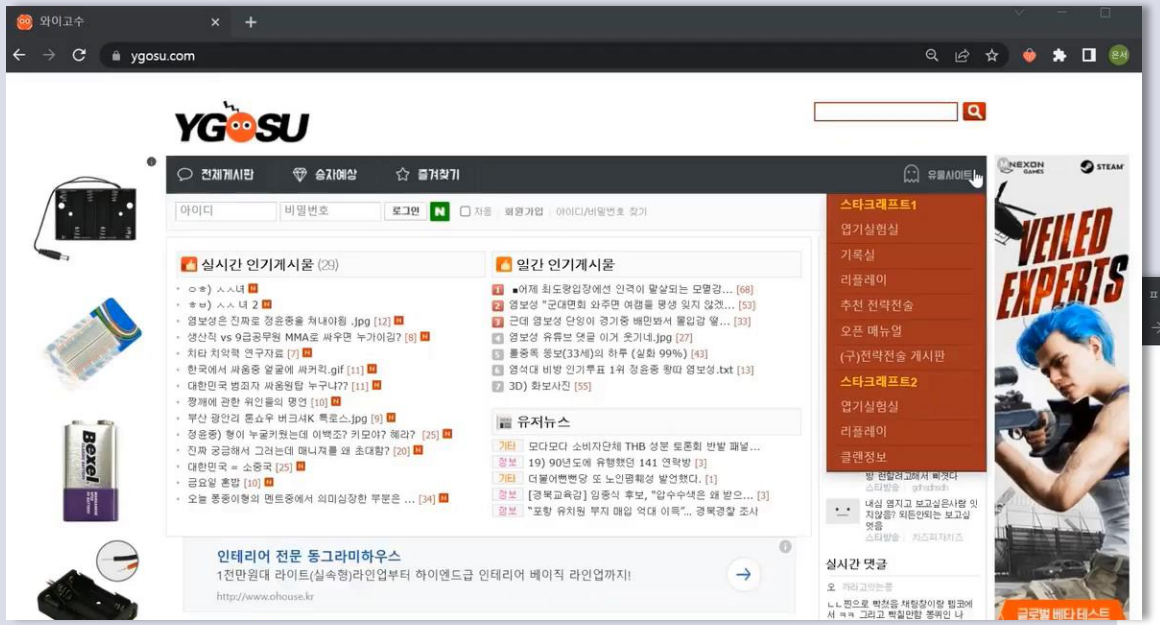
〈Shadow DOM〉

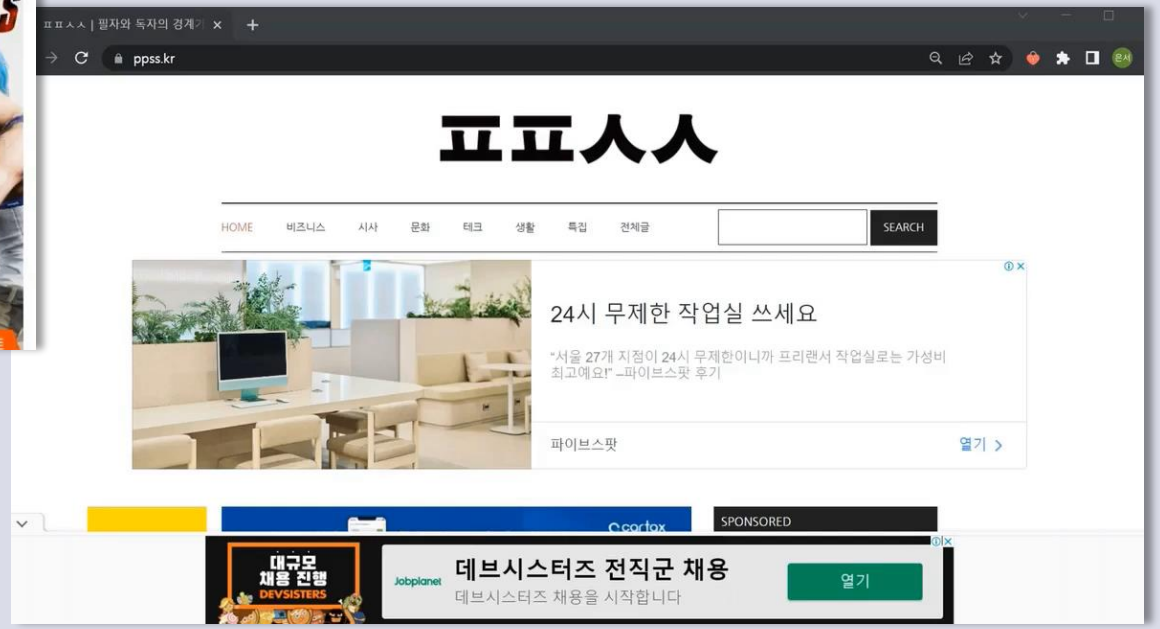


- employs membrane pattern to install hooks to every browser API that websites may use to query the adblocking status.
- By returning results indicating that ads are not being blocked, it neutralizes website's antiadblock.
- implements Shadow DOM bypassing, so that it can target ads in the presence of DOM obfuscation techniques using Shadow DOM.

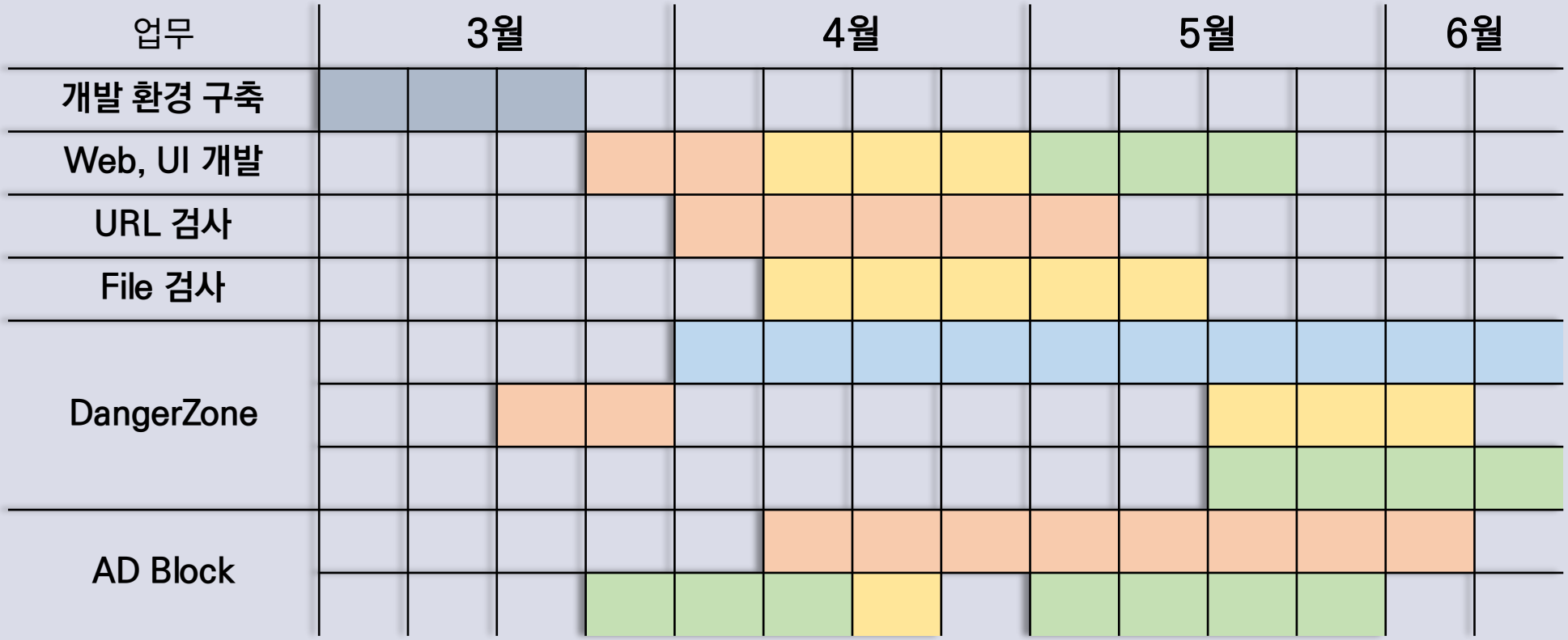
수행 내역 및 시연

4. 광고 배너 차단 – 일부 사이트 적용 문제 해결





팀원 별 담당 업무



이채은

이은서

서아름

송승민

모두

프로젝트 수행 후기

- 제안발표 당시 계획했던 **핵심 기능 구현 모두 완료**
- 많은 case들을 테스트 하며 정확한 알고리즘 개발에 노력

  → 특이 case에 적용이 되지 않는 문제에도 적용 가능하도록 **추가 구현 (Css 기반 광고차단)**
- 악성파일을 직접 다룸에 있어서 감염될 위협이 있어 고립된 환경 사용이 요구

  → 새로운 sandbox 환경의 **docker**라는 프로그램을 직접 학습하며 프로그래밍 실력 향상
- 기술적(기능적) 개발 요구 사항 외에 비기능적 요구사항(Usability, Reliability, Performance) 을 고려
- 개발자에게 꼭 필요한 협업 경험을 쌓을 수 있던 좋은 기회

감사합니다

2022 사이버보안 캡스톤 디자인 2조

악성위협 감지 시스템

이채은 이은서 서아름 송승민