

**Sujet d'épreuves des Sélections régionales
de la 47^e Compétition des Métiers**

MÉTIER N°54

CYBESECURITE

MODULE A ET B

Soumis par :

Samy SCANNA, Expert WorldSkills France

Equipe Métier :

François CAPON, Equipe Métier Cybersécurité, concepteur Module B

TABLE DES MATIERES

1.	EXPLICATION DU SUJET.....	3
2.	PLANNING JOURNALIER	11
3.	MATERIAUX ET CONSOMMABLES	12
4.	OUTILLAGE PERSONNEL	ERREUR ! SIGNET NON DEFINI.
5.	BARÈME DE CORRECTION.....	13
6.	ANNEXES	13

1. EXPLICATION DU SUJET

DURÉE TOTALE DE L'ÉPREUVE :

4h30 heures

DIFFUSION DU SUJET :

Découvert le jour de la compétition

PREAMBULE

INTRODUCTION

WorldSkills France accueille la compétition mondiale en 2024 à Lyon, dans ce contexte les Administrateurs Systèmes & Réseaux de WorldSkills ont fait évoluer l'infrastructure informatique afin de permettre un travail des collaborateurs dans les meilleures conditions.

En votre qualité d'auditeur Cybersécurité, WorldSkills France vous sollicite pour réaliser un test d'intrusion sur sa nouvelle infrastructure, pour auditer certains codes applicatifs et afin d'anticiper toute future attaque informatique, vous demande d'auditer une machine qui serait à priori infectée.

CONSEILS ET CONSIGNES DANS LE CADRE DE LA RÉALISATION DU SUJET

En tant qu'équipe prétendante au titre de champion régional dans votre métier ainsi qu'à la représentation de votre région aux prochaines finales nationales, votre victoire passera inéluctablement par la compréhension des éléments suivants.

Le sujet est rédigé de sorte que les réponses attendues aux questions soient pour la majorité des réponses uniques et simples, c'est-à-dire que les réponses seront les mêmes pour tout le monde et aucune variation ne sera possible.

Ainsi soyez très attentif aux éléments de réponse, si le résultat attendu est « **explorer.exe** », ne donnez pas « **Explorer.exe** » comme réponse, cela sera considéré comme faux. L'objectif ici n'est pas de vous embêter dans votre épreuve mais de vous préparer au mieux à la compétition internationale, car certaines questions/configurations seront corrigées automatiquement et attendront un résultat strictement identique à ce qui est prévu.

Soyez donc attentifs aux majuscules/minuscules et aux accents.

Si vous pensez ne pas pouvoir finaliser le sujet, concentrez-vous sur ce que vous maîtrisez. **Votre objectif pour la compétition est de capitaliser un maximum de points.**

Concentrez-vous sur votre réalisation et non celle des autres.

On peut parfois être tenté de regarder l'état d'avancement des autres compétiteurs, mais sachez que dans notre métier, dû à son manque de concret visuel, les personnes qui semblent les plus avancées ne sont pas forcément celles qui capitaliseront le maximum de points.

Enfin, **n'oubliez pas que vous formez une équipe**, alors ne vous précipitez pas. Prenez le temps de lire le sujet une première fois dans son **intégralité**, **communiquez** avec votre co-équipier et **partagez**-vous les tâches, **optimisez** votre temps.

DESCRIPTION DU SUJET

Cette première partie comprends deux modules à réaliser d'une traite en **3 heures** :

- **Module A** – Test d'intrusion (Windows & GNU/Linux)
- **Module B** – Revue de code vulnérable (Python & PHP)

Ces deux modules se réalisent sans **aucune connexion à Internet**, seul l'infrastructure à tester, une machine Kali (VM Auditeur) et le code à auditer vous sont fournis.

MODULE A – TEST D'INTRUSION (WINDOWS & GNU/LINUX)

Ce module consiste à mener un test d'intrusion en boîte noire dans un environnement Windows & GNU/Linux.

Vous disposez d'une **VM Kali complète** avec tous ses outils par défaut et cette VM est directement connectée au même réseau local que les machines à attaquer.

Le plan d'adressage et les adresses IP des machines à attaquer ne vous sont pas communiqués, c'est à vous de le découvrir.

Les machines à attaquer sont les suivantes :

- **WS-SRVWIN01**
- **WS-CLTWIN01**
- **WS-SRVLIN01**

L'ensemble des machines (VM Kali et LAB à attaquer) sont disponibles sur vos deux postes physiques, directement dans le logiciel VMWare Workstation, vous pouvez ainsi travailler à deux sur une même machine sans risquer de gêner l'autre. Les machines virtuelles sont préconfigurées (processeur, ram, réseau) et vous n'avez pas à modifier la configuration. Enfin, ne perdez pas de temps à tenter de monter les disques des VM sur vos machines pour y rechercher les flags, les machines sont chiffrées.

Restez concentré sur votre objectif, trouver des flags.

Pour réaliser un maximum de points, vous devez trouver un certain nombre de flag, ces flags ont été disposés sur les trois machines présentes dans le lab et chaque flag vaut un certain nombre de point en fonction de la difficulté nécessaire pour les obtenir.

Vous prendrez-soin de noter l'ensemble des flags trouvés dans un document texte nommé « flag_equipeXX.txt » (où vous remplacerez XX par votre numéro d'équipe) que vous laisserez bien en évidence sur le bureau de chacune de vos machines physiques (compétiteur 1 et compétiteur 2).

Accès à la VM Kali :

- Nom d'utilisateur : **kali**
- Mot de passe : **kali**

A vous de jouer !

MODULE B – REVUE DE CODE :

Pour ce module, deux codes développés spécialement pour WorldSkills France vous sont mis à disposition.

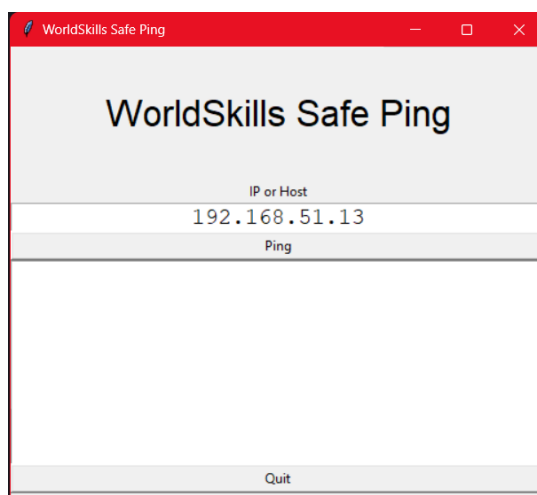
Vous devez réaliser un audit de ces codes afin d'identifier de potentielles vulnérabilités, et pour chaque code pour répondre à un ensemble de questions dont vous prendrez soin de consigner les réponses dans deux fichiers distincts (revue_code_1_equipeXX.txt et revue_code_2_equipeXX.txt) et laissés à disposition sur le bureau de vos machines physiques.

REVUE DE CODE 1 :

Afin d'aider les équipes Administration Réseau dans la résolution de problématiques techniques, un développeur a réalisé une application permettant aux équipes de réaliser de manière simplifiée des tests de ping pour s'assurer que les machines sont bien disponibles sur le réseau.

Cette application sera placée sur une machine rebond sous GNU/Linux qui disposera des accès réseau nécessaires pour réaliser les pings sur le réseau SRV-PROD. Et seule cette application sera accessible par les équipes.

Par conséquent, avant de mettre en production cette application, WorldSkills France souhaite que vous auditez cette application afin de s'assurer qu'elle ne présente aucun risque de sécurité.



Q1 – Identifiez la/les lignes de code vulnérable qui présentent un risque de sécurité. (Donnez les numéros de ligne).

Q2 – Nommez l'attaque qui est réalisable dans ce code vulnérable.

Q3 – Expliquez la vulnérabilité et donnez un exemple d'exploitation.

Q4 – Expliquez comment corriger la vulnérabilité.

Q5 – Fournir la/les lignes de code corrigées contre la vulnérabilité.

```

1  #!/usr/bin/env python
2
3  import tkinter as tk
4  import re
5  import os
6  import time
7
8  TITLE = 'WorldSkills Safe Ping'
9
10 def isPingSafe(ping):
11     reBeforeAndAfter = r'^[^\&|>|`]'
12     reMiddle = r';|&|\\|\\n|>|>>|&&|`'
13     rePattern= reBeforeAndAfter + reMiddle + reBeforeAndAfter
14     rePattern = '(' + rePattern + ')|(\$\()'
15     return not(bool(re.search(rePattern, ping)))
16
17 def doPing():
18     entryIP.configure(state='disabled')
19     pingCommand = 'ping -c 4 ' + variableEntryIp.get()
20     textOutput.configure(state='normal')
21     textOutput.delete(1.0, tk.END)
22     if (isPingSafe(pingCommand)):
23         pingCommand += ' 2>&1'
24         master.update()
25         pipe = os.popen(pingCommand)
26         for line in pipe:
27             textOutput.insert(tk.END, line)
28             master.update()
29     else:
30         textOutput.insert(tk.END, 'Alert: Unsafe Ping')
31         textOutput.configure(state='disabled')
32         entryIP.configure(state='normal')
33
34 if __name__ == "__main__":
35     master = tk.Tk()
36     master.title(TITLE)
37     tk.Label(master, text="", font=('Sans Serif',20)).pack(fill=tk.X)
38     tk.Label(master, text=TITLE, font=('Sans Serif',25)).pack(fill=tk.X)
39     tk.Label(master, text="", font=('Sans Serif',20)).pack(fill=tk.X)
40     tk.Label(master, text="IP or Host").pack(fill=tk.X)
41     variableEntryIp = tk.StringVar()
42     entryIP = tk.Entry(master, textvariable=variableEntryIp, justify=tk.CENTER, font=('Courier',15))
43     entryIP.pack(fill=tk.X)
44     buttonPing = tk.Button(master, text='Ping', command = doPing)
45     buttonPing.pack(fill=tk.X)
46     textOutput = tk.Text(master, height=10, width=150, state=tk.DISABLED,font=('Courier',12))
47     textOutput.pack(fill=tk.X)
48     buttonQuit = tk.Button(master, text='Quit', command=master.quit)
49     buttonQuit.pack(fill=tk.X)
50
51     tk.mainloop()

```

REVUE DE CODE 2 :

Afin de faciliter la transmission des factures aux partenaires de WorldSkills France, nous avons commandé le développement d'une application web permettant d'afficher les factures pour un client donné. Cette application web permet au client d'accéder à une interface web qui affiche les factures le concernant et peut filtrer ses factures en fixant une date minimale et/ou un montant minimum.

Cette application a été commandée il y a quelques mois maintenant et au vu de l'urgence nous avons dû la mettre en production sans pour autant vérifier les potentielles vulnérabilités.

Notre plus grosse crainte sur cette application, c'est qu'un tiers malveillant puisse extraire la liste de nos clients.

Dans un souci de confidentialité nous ne pouvons vous fournir l'entièreté de l'application, ainsi les extraits nécessaires à la réalisation de l'activité vous sont fournis ci-après :

- **Structure de la table [Customer]**
- **var_dump de \$customerInvoices pour un client donné.**
- **Code permettant d'afficher les factures pour un client donné.**

Q1 – Identifiez la/les lignes de code vulnérable qui présentent un risque de sécurité. (Donnez les numéros de ligne).

Q2 – Nommez l'attaque qui est réalisable dans ce code vulnérable.

Q3 – Expliquez la vulnérabilité et donnez un exemple d'exploitation.

Q4 – Expliquez comment corriger la vulnérabilité.

Q5 – Fournir la/les lignes de code corrigées contre la vulnérabilité.

Structure de la table [Customer] :

```
CREATE TABLE [Customer]
(
  [CustomerId] INTEGER NOT NULL,
  [FirstName] NVARCHAR(40) NOT NULL,
  [LastName] NVARCHAR(20) NOT NULL,
  [Company] NVARCHAR(80),
  [Address] NVARCHAR(70),
  [City] NVARCHAR(40),
  [State] NVARCHAR(40),
  [Country] NVARCHAR(40),
  [PostalCode] NVARCHAR(10),
  [Phone] NVARCHAR(24),
  [Fax] NVARCHAR(24),
  [Email] NVARCHAR(60) NOT NULL,
  [SupportRepId] INTEGER,
  CONSTRAINT [PK_Customer] PRIMARY KEY ([CustomerId]),
  FOREIGN KEY ([SupportRepId]) REFERENCES [Employee] ([EmployeeId])
  ON DELETE NO ACTION ON UPDATE NO ACTION
)
```

var_dump de \$customerInvoices :

```
var_dump($customerInvoices); // before "printing"

array(7) {
  [0]=>
  array(9) {
    ["InvoiceId"]=>
    string(3) "369"
    ["CustomerId"]=>
    string(2) "52"
    ["InvoiceDate"]=>
    object(DateTime)#3 (3) {
      ["date"]=>
      string(26) "2013-06-11 00:00:00.000000"
      ["timezone_type"]=>
      int(3)
      ["timezone"]=>
      string(13) "Europe/Berlin"
    }
    ["BillingAddress"]=>
    string(17) "202 Hoxton Street"
    ["BillingCity"]=>
    string(6) "London"
    ["BillingState"]=>
    NULL
    ["BillingCountry"]=>
    string(14) "United Kingdom"
    ["BillingPostalCode"]=>
    string(6) "N1 5LH"
    ["Total"]=>
    string(5) "13.86"
  }
  [1]=>
  array(9) {
    ["InvoiceId"]=>
    string(3) "358"
    ["CustomerId"]=>
    string(2) "52"
    ["InvoiceDate"]=>
    object(DateTime)#4 (3) {
      ["date"]=>
      string(26) "2013-05-01 00:00:00.000000"
      ["timezone_type"]=>
      int(3)
      ["timezone"]=>
      string(13) "Europe/Berlin"
    }
  }
  ...
}
```


Code affichant les factures pour un client :

```

1  <?php
2  // dev
3  //ini_set('display_errors', 1);
4  //ini_set('display_startup_errors', 1);
5  //error_reporting(E_ALL);
6  // prod
7  ini_set('display_errors', 0);
8  ?>
9
10 <?php // revue start
11
12 define('POST_DEFAULT_MINIMAL_DATE', '0000-00-00');
13 define('POST_DEFAULT_MINIMAL_TOTAL', '0');
14 define('ORDER_BY_DATE', 'order by InvoiceDate desc');
15 define('ORDER_BY_TOTAL', 'order by Total desc');
16
17 // fake customer session
18 $_SESSION['customer.CustomerId'] = 52;
19 $_SESSION['customer.FirstName'] = 'Emma';
20 $_SESSION['customer.LastName'] = 'Jones';
21
22 $pdoChinook = new PDO(
23     'sqlite:chinook.sqlite',
24     null,
25     null,
26     [
27         PDO::ATTR_EMULATE_PREPARES => false,
28         PDO::ATTR_ERRMODE => PDO::ERRMODE_SILENT, // prod
29         // PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION, // dev
30         PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC
31     ]
32 );
33
34 $sqlMinimalDate = isset($_POST['minimal_date']) ? $_POST['minimal_date'] : POST_DEFAULT_MINIMAL_DATE;
35 $postMinimalTotal = isset($_POST['minimal_total']) ? $_POST['minimal_total'] : POST_DEFAULT_MINIMAL_TOTAL;
36 $sqlMinimalTotal = (int)$postMinimalTotal;
37 $sqlOrderBy = isset($_POST['order_by']) ? $_POST['order_by'] : ORDER_BY_DATE;
38
39 $htmlDateChecked = $sqlOrderBy === ORDER_BY_DATE ? 'checked' : '';
40 $htmlTotalChecked = $sqlOrderBy === ORDER_BY_TOTAL ? 'checked' : '';
41 $htmlMinimalDateValue = $sqlMinimalDate !== POST_DEFAULT_MINIMAL_DATE ? $sqlMinimalDate : '';
42 $htmlMinimalTotalValue = $sqlMinimalTotal !== (int)POST_DEFAULT_MINIMAL_TOTAL ? $sqlMinimalTotal : '';
43
44 $pdoPreparedSelect = $pdoChinook->prepare("
45     select *
46     from invoice
47     where CustomerId = {$_SESSION['customer.CustomerId']}
48     and InvoiceDate >= :minimaldate
49     and Total >= $sqlMinimalTotal
50     $sqlOrderBy
51 ");
52 $pdoPreparedSelect->execute([':minimaldate' => $sqlMinimalDate]);
53 $customerInvoices = $pdoPreparedSelect->fetchAll();
54 foreach($customerInvoices as &$invoice) {
55     $invoice["InvoiceDate"] = date_create($invoice["InvoiceDate"]);
56 }
57 unset($invoice); // because use of &$invoice
58 ?>
59 <h1><?=$_SESSION['customer.FirstName'] ?> <?=$_SESSION['customer.LastName'] ?></h1>
60 <h2>Your Invocies</h2>

```

SUITE ->

```

60 <h2>Your Invocies</h2>
61 <form method="post">
62 <table border="0">
63   <tr>
64     <td>Mnimal Date</td>
65     <td><input type="date" name="minimal_date" value="<?=$htmlMinimalDateValue ?>"></td>
66   </tr>
67   <tr>
68     <td>Minimal Total</td>
69     <td><input type="number" name="minimal_total" value="<?=$htmlMinimalTotalValue ?>"> </td>
70   </tr>
71   <tr>
72     <td>Ordered by </td>
73     <td><input type="radio" name="order_by" value="<?=$ORDER_BY_DATE ?>" <?=$htmlDateChecked ?>>Date
74     <input type="radio" name="order_by" value="<?=$ORDER_BY_TOTAL ?>" <?=$htmlTotalChecked ?>>Total
75   </td>
76 </tr>
77   <tr>
78     <td colspan="2"><input type="submit" value="Apply"></td>
79   </tr>
80 </table>
81 </form>
82 <br>
83 <table border="1" bgcolor="white">
84 <tr><th>Number</th><th>Date</th><th>Total</th></tr>
85 <?php foreach($customerInvoices as $invoice): ?>
86   <tr>
87     <td><?=$invoice["InvoiceId"] ?></td>
88     <td align="center"><?=$date_format($invoice["InvoiceDate"], 'd/m/Y') ?></td>
89     <td align="right"><?=$invoice["Total"] ?></td>
90   </tr>
91 <?php endforeach; ?>
92 </table>
93
94 <style>
95   html {
96     background-color: whitesmoke;
97   }
98   * {
99     font-family: sans-serif;
100     cursor: default;
101   }
102   *:not(h1,h2,h3) {
103     font-size: 1rem;
104   }
105   table {
106     width: 20em;
107   }
108
109   input:not([type="radio"]) {
110     width: 10em;
111     text-align: center;
112   }
113   input[type="submit"] {
114     width: 100%;
115   }
116 </style>

```

2. PLANNING JOURNALIER

Le sujet complet devra rentrer dans une durée de concours de **4 heures 30 maximum**.

C1	DÉBUT	FIN	TÂCHES	TOTAL
	7h30		Arrivée des candidats	
	8h00	9h00	Consignes du jury, étude du premier sujet, et prise en main de l'espace métier	1h00
	9h00	12h00	Module A et B – Test d'intrusion et revue de code	3h00
	12h00	13h30	Déjeuner	1h30
	13h30	15h00	Module C – Forensic	1h30
	15h00	17h00	Correction	2h00
TOTAL ÉPREUVE (h)				4h30

3. MATÉRIAUX ET CONSOMMABLES

A) MIS A DISPOSITION PAR L'ORGANISATION

Liste des matériaux et consommables mis à disposition auprès de chaque compétiteur pour la réalisation de l'épreuve :

INTITULÉ	DESCRIPTION / RÉFÉRENCE	QUANTITÉ	REMARQUES
Poste de travail Windows 10	Destiné à héberger les VM	1 par compétiteur	Pas d'accès à Internet pour le module A et B Logiciels : VMWare Workstation, Putty
VM WS-SRVWIN01	VM du Module A	1 par compétiteur	N/A
VM WS-SRVLIN01	VM du Module A	1 par compétiteur	N/A
VM WS-CLTWIN01	VM du Module A	1 par compétiteur	N/A
VM WS-KALI	VM attaquant du Module A	1 par compétiteur	ID : kali PASS : kali

4. BARÈME DE CORRECTION

Grille avec le détail des critères de notation objectifs et jugements du module A et B.

CYBERSECURITE						
Critère	Sous Critère	Jour	Intitulé du critère de notation	Objectif ou Jugement	Barème	Coef.
			Poste de travail			
A			Critère A :			
			MODULE A : FLAGS A TROUVER			
A	01	1	FLAG 1	O	1	1
A	02	1	FLAG 2	O	1	1
A	03	1	FLAG 3	O	1	1
A	04	1	FLAG 4	O	1	1
A	05	1	FLAG 5	O	1,5	1
A	06	1	FLAG 6	O	1,5	1
A	07	1	FLAG 7	O	3	1
A	08	1	FLAG 8	O	5	1
A	09	1	FLAG 9	O	5	1
TOTAL :					20	
B			Critère B :			
			MODULE B : REVUE DE CODE			
B	01	1	Q1 – REVUE DE CODE 1	J	1	1
B	02	1	Q2 – REVUE DE CODE 1	O	1	1
B	03	1	Q3 – REVUE DE CODE 1	J	2	1
B	04	1	Q4 – REVUE DE CODE 1	J	3	1
B	05	1	Q5 – REVUE DE CODE 1	J	3	1
B	06	1	Q1 – REVUE DE CODE 2	J	1	1
B	07	1	Q2 – REVUE DE CODE 2	O	1	1
B	08	1	Q3 – REVUE DE CODE 2	J	2	1
B	09	1	Q4 – REVUE DE CODE 2	J	3	1
B	10	1	Q5 – REVUE DE CODE 2	J	3	1
TOTAL :					20	
TOTAL					40	