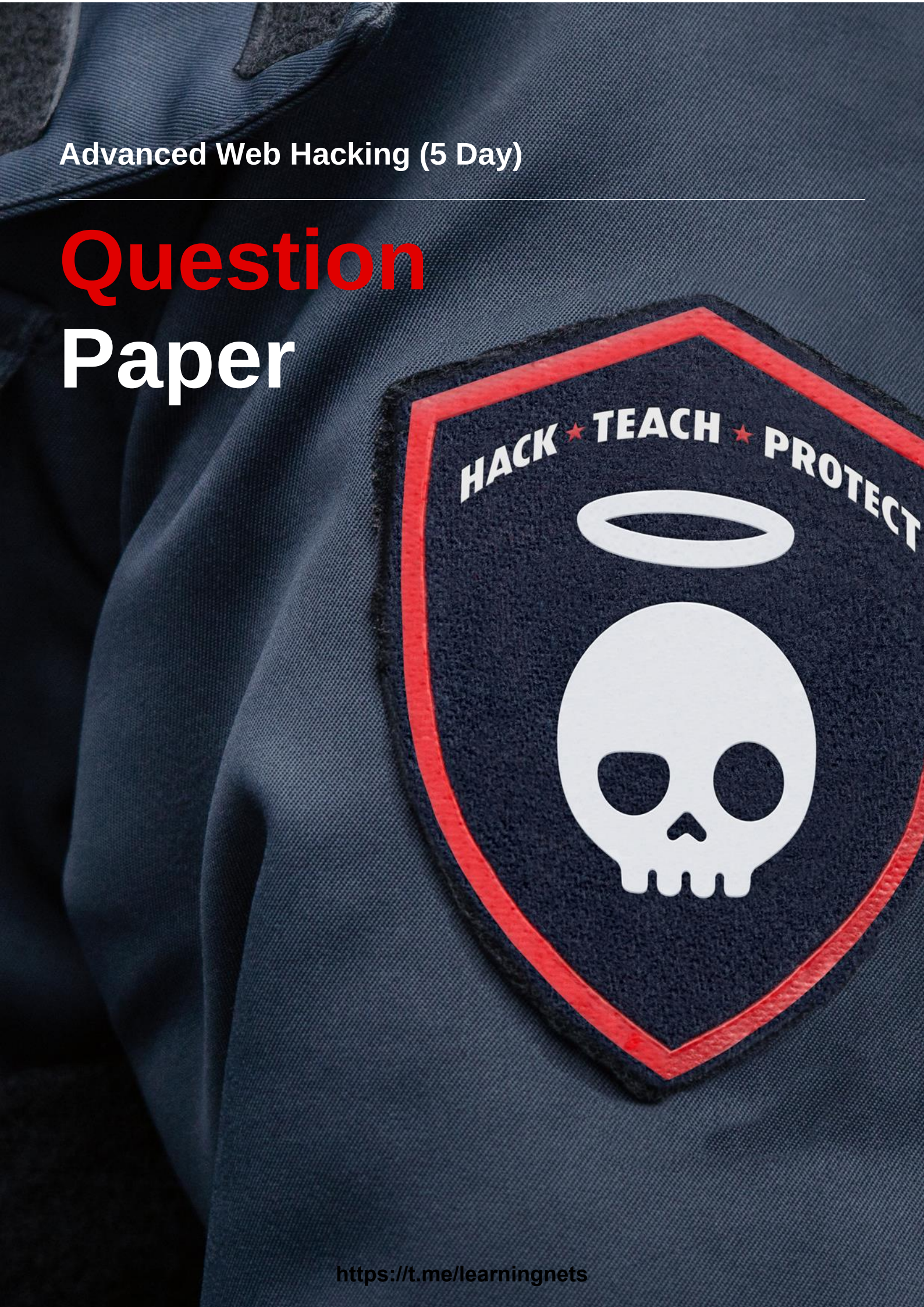
Advanced Web Hacking (5 Day)

Question Paper



Contents

Module: Attacking Authentication and SSO

Boundary Condition

Challenge URL: http://shop.webhacklab.com/login.php

- **Bypass the login security feature to login as user "bcuserX@webhacklab.com".**

JWT Brute Force Attack

Challenge URL: http://topup.webhacklab.com/Account/Login

- **Login to the "topup" application using your registered account to generate the access token.**
- **Brute-force the secret key for the JWT.**
- **Generate a valid token for user "jwtuserX@webhacklab.com" and access all the order details.**

SAML Authorization Bypass

Challenge URL: http://topup.webhacklab.com/saml/SAML.aspx

- **Login as user "not-a-john@webhacklab.com".**
- **Decode the SAML data into XML format.**
- **Exploit SAML XML to login as user "john@webhacklab.com".**

Module: Password Reset Attacks

Cookie Swap

Challenge URL: http://topup.webhacklab.com/Account/ForgotPassword

- **Change the password of the user "csuserX@webhacklab.com" through forgot password functionality.**

Host Header Validation Bypass

Challenge URL: http://topup.webhacklab.com/Account/ForgotPassword

- **Bypass host header validation to perform header poisoning for your account.**
- **Capture the password reset token.**
- **Change the password of the account using the captured token.**

Module: Business Logic and Authz Flaws

Mass Assignment

Challenge URL: http://topup.webhacklab.com/api/user

- **Escalate privilege from a "bronze" user to a "gold" user through profile update to avail additional discount.**

Invite/Promo Code Bypass

Challenge URL: http://topup.webhacklab.com/Shop/Topup

- **Identify the promo code generation mechanism for O2 Mobile.**

- **Brute-force and identify valid secret promo codes to get maximum discount on recharge (greater than 50%).**

API Authorization Bypass

Challenge URL: http://topup.webhacklab.com/api/user

- **Identify the password question of "aabuserX@webhacklab.com" user.**
- **Update the phone number of the user "aabuserX@webhacklab.com.com".**

HTTP Parameter Pollution (HPP)

Challenge URL: http://misc.webhacklab.com:5984/_utils/

- **Create a new user (userX) with "admin" role in the CouchDB instance.**

Module: XML External Entity (XXE) Attacks

XML External Entity (XXE)

Challenge URL: http://hc.webhacklab.com/

- **Identify and exploit XXE to extract the contents of the file "/etc/passwd".**

Advanced XXE Exploitation over OOB

Challenge URL: http://hc.webhacklab.com/

- **Identify and exploit blind XXE over OOB channels on the API v2 to extract the contents of the file "/etc/passwd" from the host.**

XXE through SAML

Challenge URL: http://topup.webhacklab.com/saml/SAML.aspx

- **Exploit SAML XML to perform XXE attack and extract the contents of the file "c:/windows/win.ini" from the host.**

XXE in File Parsing

Challenge URL: http://shop.webhacklab.com/career.php

- **Upload a file having "docx" type to perform an XXE attack and extract the contents of the file "/etc/passwd" from the host.**

Module: Breaking Crypto

Known Plaintext Attack

Challenge URL: http://topup.webhacklab.com/Account/ForgotPassword

- **Reset the password of the user "johnwebhacklab@gmail.com" by generating a valid password reset link**

Padding Oracle Attack

Challenge URL: http://topup.webhacklab.com/download.aspx?invoice={ciphertext_invoice}

Identify a padding oracle vulnerability to:

- **Decrypt the ciphertext for the invoice parameter.**
- **Encrypt the payload to download the content of the "web.config" file from the server**

Exploiting padding oracles with fixed IVs

Challenge URL: http://reimbursement.webhacklab.com/Support/LoadSupportTicketFile

- **Access the file where id=0 which can only be accessible by an admin.**

Hash length extension Attack

Challenge URL: http://topup.webhacklab.com/Shop/Topup [Payment]

- **Buy a topup at less than total payable amount using your registered account.**

Auth Bypass using pre-shared MachineKey

Challenge URL: http://admin.webhacklab.com/

- **Identify a pre-shared Machine Key used in the application using "Blacklist3r"**
- **Create a new auth token for "admin" user and gain access to the administrative console**
- **Use http://utility.webhacklab.com/ to generate payloads**

Module: Remote Code Execution (RCE)

PHP Object Injection

Challenge URL: http://shop.webhacklab.com/help.php

- **Exploit a PHP object injection instance to access "/etc/passwd" file from the server.**

PHP Deserialization Attack

Challenge URL: http://slim.webhacklab.com:8081

- **Identify and exploit the PHP Deserialization vulnerability.**
- **Get a reverse shell and extract the system information such as username, OS type from the server.**

Java Deserialization Attack - Binary

Challenge URL: http://mblog.webhacklab.com/login

- **Identify and inject a payload into the serialised data to make the host send DNS requests to an external host.**
- **Get a reverse shell and extract the system information such as usernames, OS type from the server and also read "/etc/passwd" file.**

Bonus: Tricky Java Deserialization Attack - Binary

Challenge URL: http://mblog.webhacklab.com/login

- **Identify and inject a payload into the serialised data to make the host send DNS requests to an external host.**
- **Get a reverse shell and extract the system information such as usernames, OS type from the server and also read "/etc/passwd" file.**

Java Deserialization Attack - XML

Challenge URL: http://mblog.webhacklab.com/api/add/microblog

- **Identify the request to inject XML serialised data and inject a payload into it to make the host send ping requests to an external host.**

- **Get a reverse shell and extract the system information such as username, OS type from the server and also read "/etc/passwd" file.**

Jackson JSON Deserialization Attack

Challenge URL: http://mblog.webhacklab.com/mblog/api/add/microblog

- **Get a reverse shell and extract the system information such as username, OS type from the server and also read "/etc/passwd" file.**

.NET Serialization Attack

Challenge URL: http://admin.webhacklab.com

- **Identify and exploit the .Net Deserialization vulnerability to make the host send HTTP requests to an external host.**
- **Get a reverse shell and extract the system information such as username, OS type from the server and read "win.ini" file.**

Python Serialization Attack

Challenge URL: http://reimbursement.webhacklab.com/Support/AddTicket

- **Identify and exploit the Python Deserialization vulnerability to make the host send DNS requests to an external host.**
- **Get a reverse shell and extract the system information such as username, OS type from the server and read "/etc/passwd" file.**

Bonus: Plex Python Deserialization

Challenge URL: http://plex.webhacklab.com:32400

- **Perform RCE using python deserialization vulnerability.**

Ruby/ERB Template Injection

Challenge URL: http://shop.webhacklab.com/referral.php

- **Identify the template engine and exploit it to extract the content of the file "/etc/passwd"**

Module: SQL Injection Masterclass

Second Order SQL Injection

Challenge URL: http://topup.webhacklab.com/Account/SecurityQuestion

- **Identify a Second order injection using your account.**
- **Exploit the injection to extract the name of the user running the service.**

SQLi Through Crypto - OOB

Challenge URL: http://topup.webhacklab.com/Shop/Order

- **Identify a data encryption endpoint using your registered account.**
- **Utilize the knowledge of encryption endpoint to confirm SQL injection using an OOB channel.**

SQL Injection to Reverse Shell

Challenge URL: http://topup.webhacklab.com/api/voucher

- **Continue with previous exercise to obtain a reverse shell on the DB host using Metasploit and native Windows tools (powershell, certutil, cscript etc.).**

Second-order SQL Injection on Joomla

Challenge URL: http://cms.webhacklab.com:81/administrator/index.php

- **Identify and exploit second order SQL Injection point in Joomla Instance**
- **Fetch the databases from database server**

Advance SQLMAP Usage with eval option

Challenge URL: http://topup.webhacklab.com/api/Product/GetProduct?pid=&sig=

- **Identify SQL Injection point**
- **Fetch the databases from the database server**

Data Exfiltration over DNS via SQLi

Challenge URL: http://topup.webhacklab.com/Account/SecurityQuestion

- **Exploit the injection vulnerability to exfiltrate the output of command "ipconfig" over DNS channel.**

GraphQL Exploitation

Challenge URL: http://expense.webhacklab.com:3000/viewexpense

- **Exploit SQL injection in one of the GraphQL endpoints and retrieve admin credentials.**
- **Use Introspection to extract the PII (Salary) of the 'userX@webhacklab.com'.**
- **Using GraphQL mutation, view expenses of all the users.**

Module: Tricky file uploads

Bypassing File Validations #1

Challenge URL: http://topup.webhacklab.com/Account/Profile

- **Identify the upload functionality and abuse it to upload a web shell.**

Bypassing File Validations #2

Challenge URL: http://shop.webhacklab.com/feedback.php

- **Bypass the file validation checks to upload a web shell (user**X**.fileextension) and execute commands on the host.**

SQLi via File Metadata

Challenge URL: http://reimbursement.webhacklab.com/Expense/Add

- **Identify and Exploit SQL Injection via File Metadata properties to retrieve current database user and database name.**

Module: Server Side Request Forgery (SSRF)

SSRF To Check Open Ports and Fetch File

Challenge URL: http://shop.webhacklab.com/products.php

- **Utilizing SSRF extract the contents of the internal file "/etc/passwd".**
- **Identify the ports open on the host "http://192.168.200.10/".**

SSRF via PDF Generation

Challenge URL: http://topup.webhacklab.com/Account/Profile

- **Utilise PDF export injection to confirm SSRF using OOB channel.**
- **Retrieve the content of the internal file "win.ini":**

Module: Cloud Pentesting

AWS - SSRF Exploitation - Elastic Beanstalk

Challenge URL: http://cloud.webhacklab.com/view_pospdocument.php?doc= {}

- **Identify and exploit SSRF vulnerability to gain access to S3 buckets and download the source of the application hosted on AWS cloud.**
- **Upload a webshell via Continuous Deployment (CD) pipeline.**

AWS Serverless Exploitation

Challenge URL: https://8nfjm12vx0.execute-api.us-east-2.amazonaws.com/default/awh-lambda-demo?query='notsosecure'

- **Identify and exploit Remote Code Execution vulnerability in the Lambda function**
- **Obtain Secret Tokens**
- **Gain access to S3 bucket**
- **Connect to EC2 instance**

Leaked Storage Account

Challenge URL: N/A

- **Extract the source code and achieve Remote Code Execution for the function from the storage account of “notsosporty" using the techniques learned in this module.**

Exploiting AWS Cognito Misconfigurations

Challenge URL: http://cognito.webhacklab.com/

- **Identify AWS cognito misconfiguration and read the secrets from the secret manager.**

Module: CMS Pentesting

Pentesting Hardened CMS

Challenge URL: http://cms.webhacklab.com/wordpress/

- **Identify and exploit Vulnerabilities in WordPress instance**
- **Fetch the databases from the database server**

Module: Web Cache Attacks

Web Cache Deception

Challenge URL: http://webcache.webhacklab.com:8080/login.php

- **Identify Web Cache Deception vulnerability to access sensitive content without authentication, which would otherwise be only accessible to an authenticated User.**

Web Cache Poisoning

Challenge URL: http://webcache.webhacklab.com/

- **Identify whether there are any unkeyed inputs used by the application and if the server caches the output for the same. Edit those unkeyed inputs with malicious payloads to do the following to random user when poisoned cache is requested.**

  a) Perform Cross-Site Scripting

  b) Execute malicious script from remote location controlled by us

  c) Steal Credentials through Form submission to remote location controlled by us.

Note: TTL of cache is set to 20 sec.

Module: Miscellaneous Vulnerabilities

Unicode Normalization Attack

Challenge URL: http://reimbursement.webhacklab.com/Account/ResetPassword

- **Identify and exploit the forgot password functionality to login as userX**

Second-order IDOR

Challenge URL: http://reimbursement.webhacklab.com/Expense/LoadExpenseFile?id=

- **Exploit Second-order IDOR to view reimbursement details of another user on the application who owns id = 1, 2, 3**

Leverage Git misconfiguration to ViewState RCE

Challenge URL: http://books.webhacklab.com/.git

- **Leverage Git misconfiguration to extract the Machine Key.**
- **Exploit ViewState to perform Remote Code Execution(RCE)**

HTTP Desync Attacks

Challenge URL: http://covid19.webhacklab.com:5000

- **Discover the Cross-Site Scripting vulnerability.**
- **Perform HTTP Desync Attack to get the Cross-Site Script executed when a new user visits.**