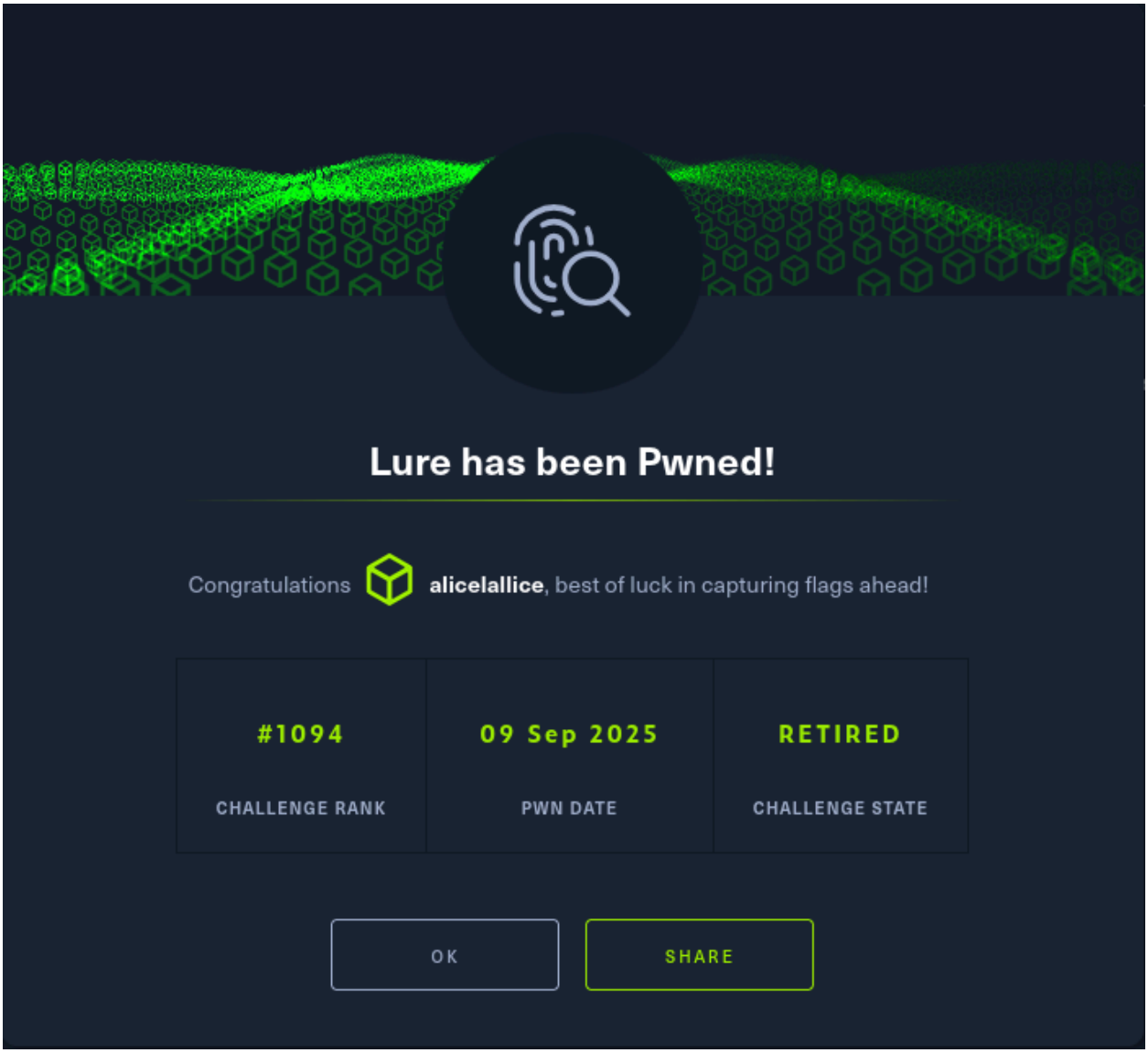
Lure

Types	forensic
CTF	HTB



1. Initial Analysis

- Received file: `UrgentPayment.doc`

- Checked with `strings`:

```
strings UrgentPayment.doc
```

Observations:

- Prompts: “Enable Editing”, “Enable Content” → **social engineering**
- Protected by Microsoft Office Protected View
- Embedded OLE streams, typical of Word documents with macros
- References to `Document_Open` and PowerShell

2. Detect Macros with `olevba`

```
olevba UrgentPayment.doc
```

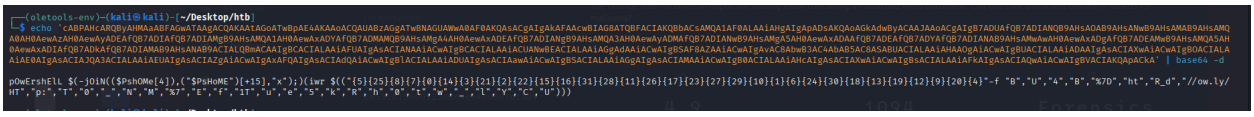
Findings:

- Auto-executing macro: `Document_Open`
- Reads environment variable: `Environ$("UserDomain")`
- Runs a **PowerShell encoded command** (`Shell("pOweRshElL -ec ...")`)
- **Suspicious keywords:** `Environ`, `Shell`, `vbNormalFocus`, `pOweRshElL`, **Hex Strings**

> These are classic indicators of malicious Office macros.

Decode the PowerShell Command

The PowerShell command is **base64 encoded in UTF-16LE**. To safely decode:



Deobfuscate PowerShell (Python)

Malware used **character-by-character obfuscation**. Decoding safely with Python:

```
template = "{5}{25}{8}{7}{0}{14}{3}{21}{2}{22}{15}{16}{31}{28}{11}{26}{17}{23}{27}{29}{10}{1}{6}{24}{30}{18}{13}{19}{12}{9}{20}{4}"
chars = [
    "B","U","4","B","%7D","ht","R_d","//ow.ly/HT","p:","T","0","_","N","M","%7","E","f","1T",
    "u","e","5","k","R","h","0","t","w","_","l","Y","C","U"
]

import re
indexes = [int(i) for i in re.findall(r"\{(\d+)\}", template)]
url = ''.join([chars[i] for i in indexes])
print("Decoded URL / Flag:", url)
```

. Decode URL-Encoded Flag

- `%7B` → `{`
- `%7D` → `}`

Final flag:

```
HTB{k4REfUl_w1Th_Y0UR_d0CuMeNT5}
```

✅ Flag successfully retrieved **without executing any malicious code**.

Tools & Commands Used

Tool	Purpose
`strings`	Quickly check the document for suspicious text or macros
`olevba`	Detect and analyze VBA macros in Office documents
Python	Decode base64 and UTF-16LE encoded PowerShell, deobfuscate strings

Tool	Purpose
`base64`	Decode base64 content