**25Q2 CSOC User Agent Based Attack**

PREPARED BY:

Mohammed Sharoz

Senior In

REVIEWED BY:

Diwakar Thanikachalam

SIGNED-OFF BY:

Name Middle Lastname

### Introduction:

The applications that are owned by Zalaris are actively for vulnerability using manual Penetration test

# 

Table of Contents

[Introduction: 2](#_Toc24177)

[2025 Q2 CSOC User Agent Based Attack 3](#_Toc566)

### 2025 Q2 CSOC User Agent Based Attack

**Description**

The User-Agent is an elementary part of every HTTP request.

User-Agent strings have many forms, and typically look similar to one of the following examples:

1. Mozilla/5.0 (Windows NT 10.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/34.0.1866.23 Safari/537.36
2. Mozilla/5.0 (iPad; CPU iPhone OS 9\_0\_2 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Mobile/12A405

In practice, anyone can put whatever they want in the User-Agent string, and send it to the web server

Parsing of the User-Agent string — that gets targeted by hackers.

The hackers’ rule of thumb says that when a remote system is processing a user-controlled input and the User-Agent string is, entirely controlled by the user

**Known Attacks**

|  |
| --- |
| * Shellshock Remote Code Execution * Arbitrary Code Execution |

### Test for Shellshock Remote Code Execution

**Description :**

Shell-shock is a critical vulnerability in the Bash shell that allows attackers to execute arbitrary commands via specially crafted environment variables. This test is targeted against a liknux / unix application or server that process the user-agent string and parses the input in the user-agent string

In September 2014, a vulnerability was discovered in bash, a popular shell (command-line interface) in Unix systems. The vulnerability, when exploited, allowed an attacker to take over the affected machine and execute arbitrary commands. This type of attack, named Remote Code Execution (or RCE for short)

Pre-requisites:

* The server has a CGI script (like /cgi-bin/test.sh) that’s interpreted by BashAnd that script gets triggered by the request path
* The malicious command in the User-Agent (or other headers) gets processed by Bash as part of environment variable parsing.

Sample Request and response is given below

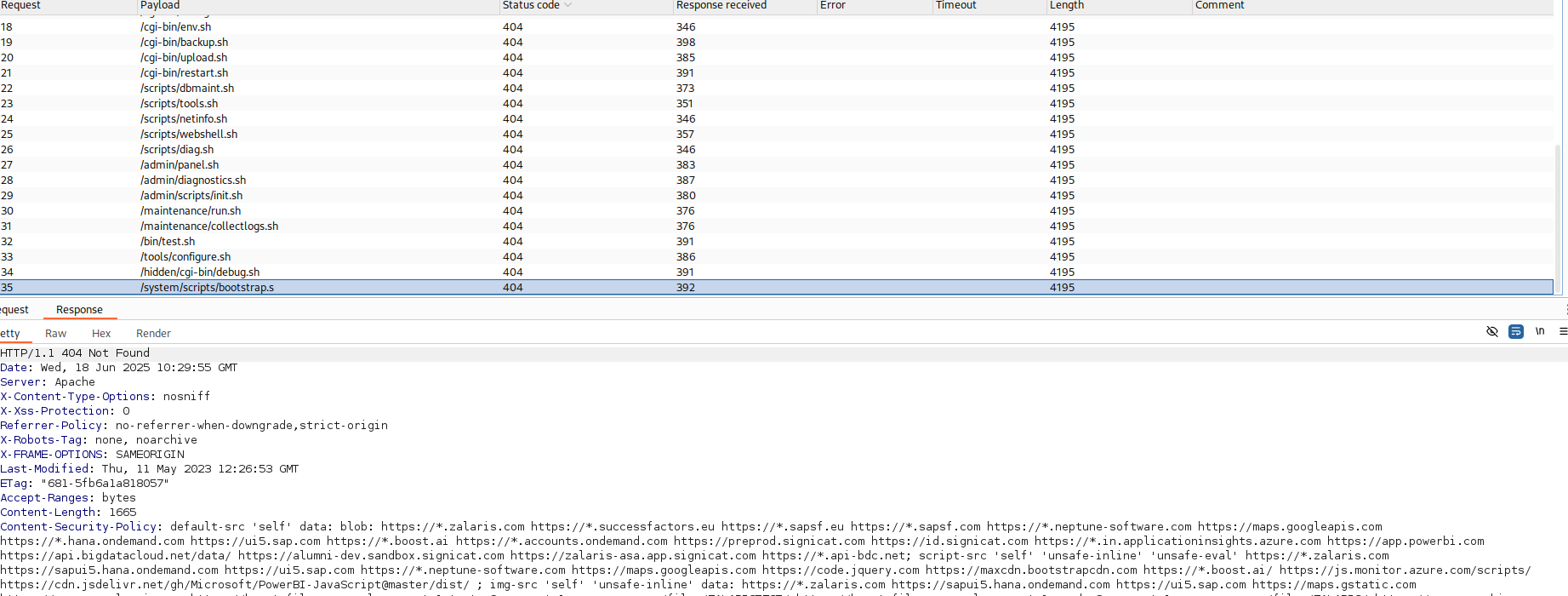
|  |
| --- |
| **Request :**  GET /cgi-bin/test.sh HTTP/1.1  Host: victim.com  User-Agent: () { :; }; echo; /bin/bash -c "echo HACKED"  **Response:**  HTTP/1.1 200 OK  Content-Type: text/plain  HACKED  The above request and response confirms that the application has an attack surface for the external attacker to access the internal bash as a command line user and execute the script |

Here are the list of the paths that can be called by an external user

|  |
| --- |
| /cgi-bin/test.sh  /cgi-bin/status.sh  /cgi-bin/admin.cgi  /cgi-bin/backup.sh  /cgi-bin/env.sh  /cgi-bin/config.sh  /cgi-bin/dbmaint.sh  /cgi-bin/diagnostics.sh  /cgi-bin/upload.sh  /cgi-bin/monitor.sh  /cgi-bin/tools.sh  /scripts/util.sh  /admin/panel.sh  /bin/webshell.sh  /cgi-bin/test.sh  /cgi-bin/status.sh  /cgi-bin/config.sh  /cgi-bin/env.sh  /cgi-bin/backup.sh  /cgi-bin/upload.sh  /cgi-bin/restart.sh  /scripts/dbmaint.sh  /scripts/tools.sh  /scripts/netinfo.sh  /scripts/webshell.sh  /scripts/diag.sh  /admin/panel.sh  /admin/diagnostics.sh  /admin/scripts/init.sh  /maintenance/run.sh  /maintenance/collectlogs.sh  /bin/test.sh  /tools/configure.sh  /hidden/cgi-bin/debug.sh  /system/scripts/bootstrap.s |

**Steps:**

* Capture the request from the target application
* Alter the User agent string with the target payload
* Alter the path according to the above list [list is just for reference any path that invokes a internal command line can be called]
* Forward the request to intruder and alter the user agent to make a ping to the collaborator server 
* The response received is 404 and no ping has been detected in the collaborator - this confirms that for the given paths no bash command has been executed . The screenshot for the same is attached below



**WORKLOG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| When | What | Who | Inference | Action plan |
| 18 June 2025 | Manually Tested for Remote Code Execution | Mohammed Sharoz | The scan done so far did not result in remote code execution [only non harmful code alone is used] | Need further testing using other non harmful scripts |