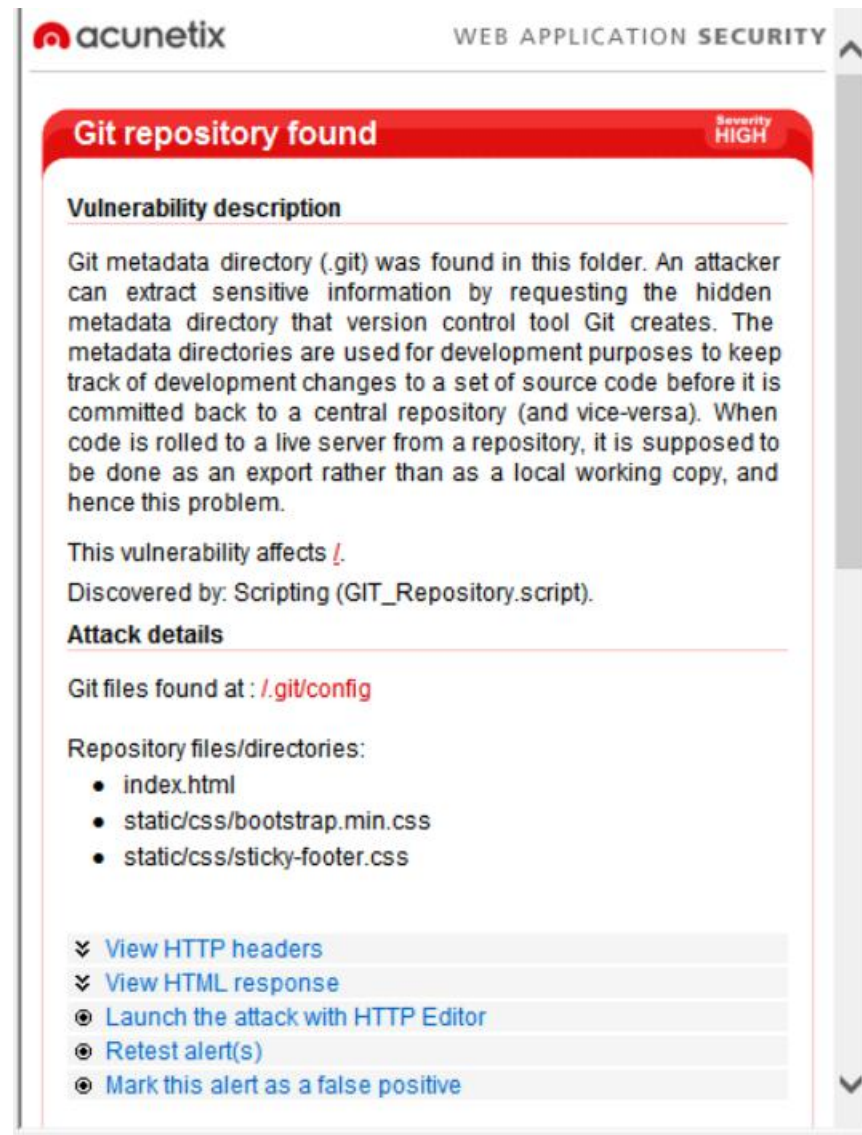


HGAME 2020 WEEK1 WITE UP

一、WEB

1. Cosmos 的博客（50）

看提示知跟 git 有关，扔进 awvs



The image shows a screenshot of the Acunetix Web Application Security interface. At the top, the Acunetix logo and 'WEB APPLICATION SECURITY' are visible. A red alert banner at the top reads 'Git repository found' with a 'Severity HIGH' label. Below this, the 'Vulnerability description' section explains that a Git metadata directory (.git) was found, which can be exploited to extract sensitive information. It also states that the vulnerability affects '!' and was discovered by 'Scripting (GIT_Repository.script)'. The 'Attack details' section lists the files found at '/.git/config' and provides a list of repository files/directories: index.html, static/css/bootstrap.min.css, and static/css/sticky-footer.css. At the bottom, there are several interactive links: 'View HTTP headers', 'View HTML response', 'Launch the attack with HTTP Editor', 'Retest alert(s)', and 'Mark this alert as a false positive'.

acunetix WEB APPLICATION SECURITY

Git repository found Severity HIGH

Vulnerability description

Git metadata directory (.git) was found in this folder. An attacker can extract sensitive information by requesting the hidden metadata directory that version control tool Git creates. The metadata directories are used for development purposes to keep track of development changes to a set of source code before it is committed back to a central repository (and vice-versa). When code is rolled to a live server from a repository, it is supposed to be done as an export rather than as a local working copy, and hence this problem.

This vulnerability affects **!**.

Discovered by: Scripting (GIT_Repository.script).

Attack details

Git files found at: **/.git/config**

Repository files/directories:

- index.html
- static/css/bootstrap.min.css
- static/css/sticky-footer.css

✕ [View HTTP headers](#)

✕ [View HTML response](#)

🔍 [Launch the attack with HTTP Editor](#)

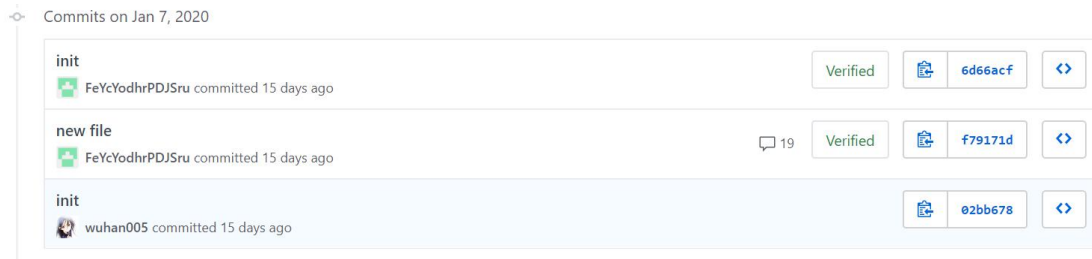
🔍 [Retest alert\(s\)](#)

🔍 [Mark this alert as a false positive](#)

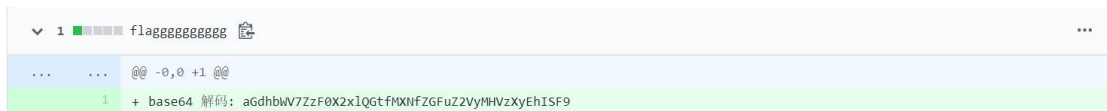
拿到地址，看下有啥东西

```
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
[remote "origin"]
    url = https://github.com/FeYcYodhrPDJSru/8LTUKCL83VLhXbc
    fetch = +refs/heads/*:refs/remotes/origin/*
```

找到 github 地址，过去看看历史记录



一共三个，这个留言那么多指不定什么问题，进去看看



Base64 解码拿到 flag

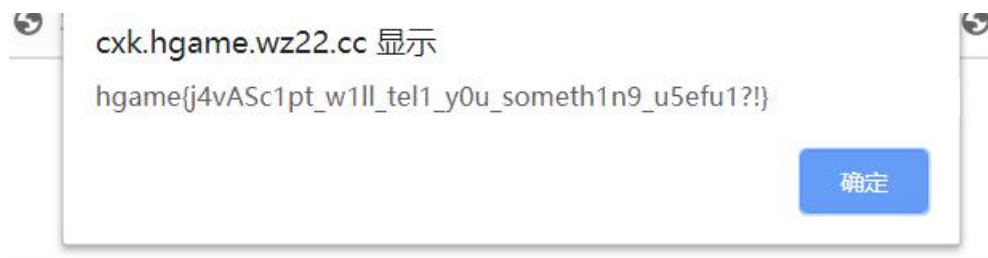
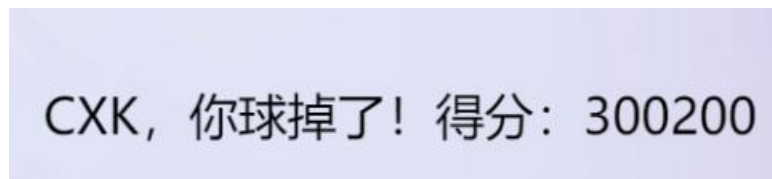
2. 鸡尼泰玫 (100)

小游戏我喜欢

我把这鬼游戏打完了三关

顺手 F12 看源码

```
class Game{storageScore=0;globalScore=0;constructor(main){let g={main:main,actions:{},keydowns:{},state:1,state_START:1,state_
...
class Game{storageScore=0;globalScore=3000000;constructor(main){let g={main:main,actions:{},keydowns:{},state:1,
```



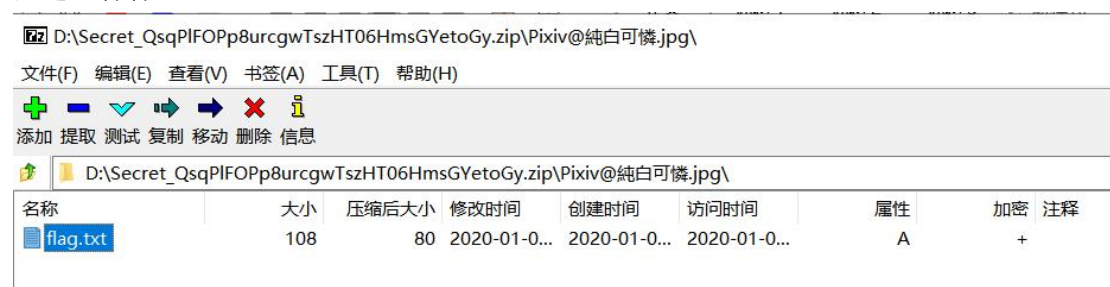
拿到 flag

2. 壁纸

用 7z 开压缩包

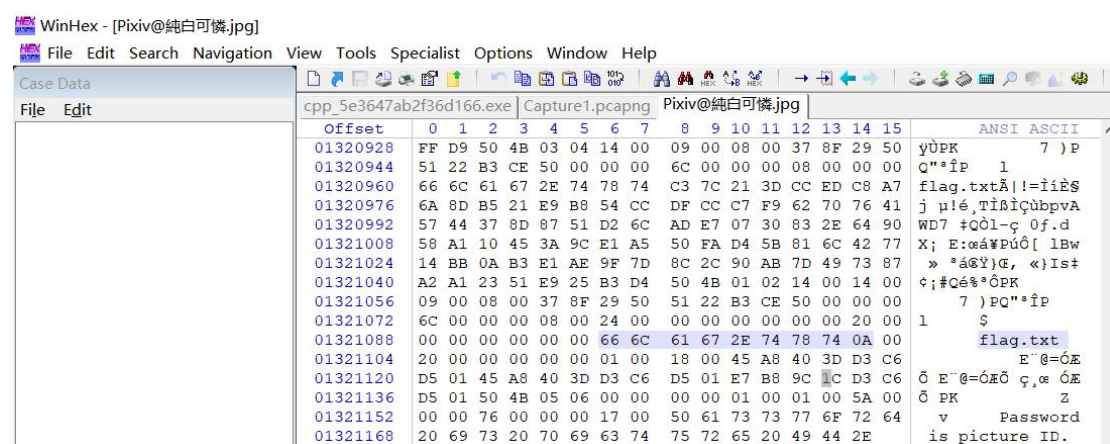


点进去看看



发现一个加密的 flag.txt，这个肯定就是了

Winshark 开图片文件



最后提到“password is picture ID”

图片能有啥 id? 这个时候想起来是 p 站图片，每一个 p 站图都会有自己的 id

去找

Saucesao 启动!

找到 ID，即为 flag 密码，打开 flag

得到一堆/u 啥啥，查了是 unicode 编码，转中文得到 flag

\u68\u67\u61\u6d\u65\u67\u64\u6f\u65\u67\u60\u67\u65\u6b\u6e\u6f\u65\u67\u65\u64\u69\u63\u6f\u6d

中文转UNICODE

UNICODE转中文

生成的相应的字符串

hgame{Do_y0u_KnOW_uNiC0d3?}

4. 签到题 pro

password.txt	312	218	2020-01-1...	2020-01-1...	2020-01-1...	A		1D6618B9	Deflate	NTFS	FAT	20
OK.txt	1 024 698	43 933	2020-01-1...	2020-01-1...	2020-01-1...	A						+

一个加密文件，一个密码

打开看看

Rdjxfwxjfmkn z,ts wntzi xtjrw m xsfjt jm ywt rtntwhf f y h jnsxf qjFjf jnb rg fiyykwtbsnkm tm xa jsdwqjfmkijy wlviHtzqsGsffwyjyynf yssm xfyypnyihjn.

JRFVJYFZVRUAGMAI

* Three fences first, Five Caesar next. English sentence first, zip password next.

上网查询，fences 应该指栅栏加密，caesar 应该指凯撒加密，3、5 分别指的是参数。解密得到密码

EAVMUBAQHQMVPEPDT

即为 ok.txt 的密码。打开

data:text;ook,
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook? Ook. Ook? Ook! Ook. Ook? Ook! Ook. Ook! Ook! Ook! Ook! Ook! Ook!
Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook!
Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook. Ook? Ook!
Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook? Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook! Ook! Ook! Ook!
Ook! Ook! Ook! Ook! Ook? Ook. Ook? Ook! Ook. Ook? Ook! Ook! Ook! Ook! Ook!

什么鬼？一堆 ook？查一下

原来也是能翻译的，翻一下

All the hard work (like actually understanding how those

and his Brainfuck interpreter in PHP

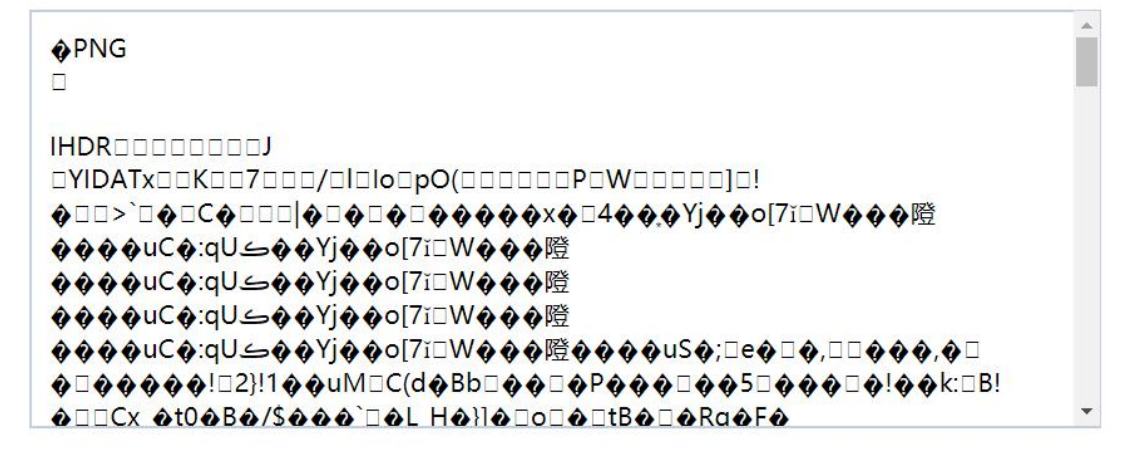
```
data:text;base32,NFLEET2S04YEW3HN5AUCQKBJZJVK2CFKVTUCQKBKFUICQK
BIVCUGQKZIFAUCRCPINCW6S2BIFAU6V2VNRCVCVSSGRXE6MTBKM3DIRK0053UIMZ
PGB3DOV3ZINJV00CHMNCTQ33LMJFXEQKPHBQSW3CBKNLC6MRTIFAUIKZVMM4WIQK
BIRVWOQ2FIF3UCY2NIFIUCK2ZIFTUCOCBIZCECSKBKBDUCSKBMZGUCUKBJ5AUI2D
HIFAUQN2ZJU2GKL3WNIXWINBQM5CTANJZ0ZHG2YLYLJQW6L2KMIYXGM3YJMYFIVR
WK52G25LNMFGYMYKZF5GFUMKRHF3TMY2WLBQXC4DN0VLV04KQPFLTSYSOHBXJXRJ
RMVWHEWTSOBWXCWBSNVIHSMTEKVIIGT30IZLDE4LRLJZGY3DRNI4GY5SXPJTEK4S
SJZMHAYJSME3FU4LMHFIYUODUNZLEIM2E0B4FMZDR0FWWCNK2MFYXS6STCGFZTG6C
LGBKFMNSXORWXX3LBOBTGCWJPJRNDCUJZ043GGVSYMPFYXA3LVK5LXCUDZK44WETR
YKN2EKMLFNZRZFU4TQNVYVQMTNKB4TEZCJ5FU66BRNRXHON20JRAXGVLVOR5HGTC
```

Text to Ook!	Text to short Ook!	Ook! to Text
Text to Brainfuck	Brainfuck to Text	

看起来是 **base32** 的密码，再解一下

```
E8BnDWciU5RTMM4QGGcNZyK7lFMwzhAYZw1nrlrUuZDOEBhndWciU5RTMM4QGGcNZyK7lFMwzhAYZw1nrlrUuZDOEBhndWciU5RTMM4QGGcNZyK7lFMwzhAYZw1nrlrUuWzdkNQLm3ycitqut3aJtaf7Jm3fqvhSGU9anMqa7foW1q/cmadeu/FoZQ1qcy
p7p+h7ap95dr1q3/W/hhCWZ/KnO6Hdqmp1p+swBf+a2ElZX0qcrrd2ibWn+yZt36r4UhlPWPzKmu36tfav3JmnXrxvaGUNanM
qe6foe2qfUna9at/1oYUlmfypzq+h3aptafrfm3/mthCpUG9KnoQq63dom1p/smbd+q+FlDnWk4keeYOMATP55g4wPBoc9YANm
+OekusZO8A6hPcU5l8cwcYAn2OeoQAH+4AQ6DCV9J8swdYAj0Oe4p2S65AwyBPsc9Jckzd4W0Oe4p7H56g4wPBoc95Kqz9w
BhlDUZZLug+6Yp2s3xvfZnTRVWVHfbybgOefSh3Njnd9JUydv9JuM65NGHckuf3UIthVV3mYzrkEcfcjy19didNFVbdZzKuQx59Klf
0Z2O0Vqhj1n8m4Dnn0odzS3fSVGHVfSbjOuTrh3JLn91JU4V95mM65BHH8otfXyNTRVW3WcyrkMefSi39NmdNFVdyZ/JuA55
9KHCOVcSagp3jFYNNr1n+XE+4EQ3iQUApD0A0x37frcvskdYAgPUhfSYBpr1v/XU64EwzhQepCug8w7XXR8sJd4lHPEhSDpCBpr1
u/Xc5U4UwhAepC+k+wLTXrf8u9JwhvAqdSHdB5j2uxfV5YQ7wRAepC6k+wDTXrf+u5xwJxjCg9SFdB9g2uvWf5cT7kRqCLc8pW
ZKunt01ebo85qc3YwdbJ5aXKeskKm2R5/X5Owu7iSn1EjY4EctN6vCZnd2EnPKvMShwL5KjN0eCt1obSLo+EpNVPiWCBBHy4+
r8nZXdgJ7m6ZEscCOWpz9HINzu7CTnhkZQ4FshRm6Ppa3J2F3bCU2qmxLFAjtocfV6T57uwE55SMYWOBXLU5ujzmpzdH3zWl
opcSyQozZHn9fk34SAI4hQwCADXgCAHZ+AEADgA4YAAB8wBAD4gCEAwACMAQA+YAgA8AFADAIAPwAFOhUzrh1mAAAAA
BJRU5ErkJggg==
```

末尾还有==，再来一次 base64



好像是个图片
找工具 **base64** 转图片



扫码得 flag

我好菜