Monday, March 10, 2014

RuCTF 2014 Quals - TLS (Crypto 300)

Intro

The task consisted of a 19kB pcap file with a single complete TLS conversation between a client and an HTTPS server (using DHE_RSA), and a rather laconic description - **"just break TLS"**. Well, since you asked…

Poking around

Shortly after opening the file in Wireshark intersting details surface. While the server looks absolutely valid, the client seems to have a rather unusal random number generator. The 0x20 byte long random nonce sent in the Client Hello message is:

```
0000000: 4469 4865 2031 3333 3720 3133 3337 2031  DiHe 1337 1337 1
0000010: 3333 3720 3133 3337 2031 3333 3720 3133  337 1337 1337 13
```

Since that looks very non-random, perhaps the client exponent is easy to figure out?

We can get the Diffie-Hellman parameters **p** and **g** from the Server Key Exchange mesage, we also have g^x from the Client Key Exchange:

```
4a771bbd30b56bb87089a665976efc66363448588236d6f61e64e7dfaf54
b187df22337a75930d622b71fc88fb4f5d4af2384e8f0e4a11c967d699f3
05144c369207990053cb2d5e70e596aea4b5b1ac2c274ae08e1eb1bb1d78
eb3b9fd3702d78610b15d39352cbf748919d6930245f4d3e4fc9f48504a1
5e132f08b9c50fb9
```

The first attempt - assuming the exponent to be " 1337", repeated to fill 32 bytes and shifted - was unsuccessful. The second - trying the number **1337** - worked just fine.

So now that the client's private exponent has been recovered, it's time to decrypt the session…

Decryption

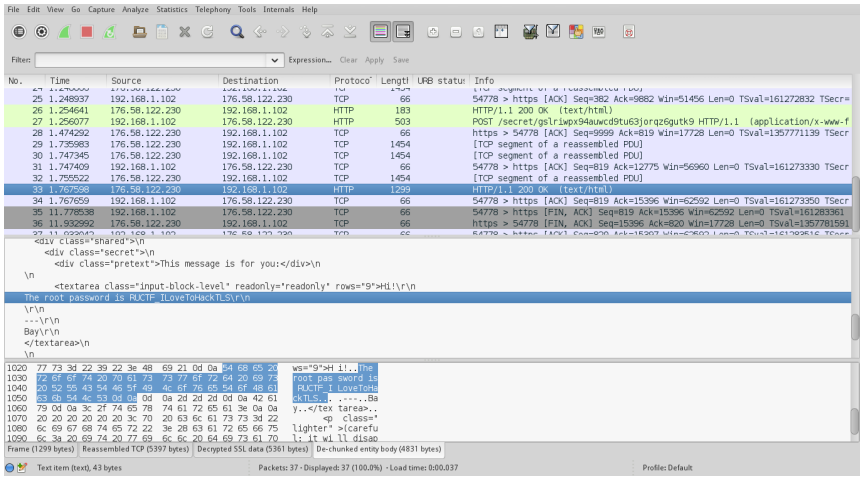
Wireshark comes with a built-in SSL decryption facility. What it needs is a Session ID and a Master Secret. The Session ID is (as this is a new session) sent in the Server Hello message, so we have that. We also have the client secret (1337), the generator (2), the prime and the server public key (g^y). This allows us to compute g^{xy}, the Pre-Master Secret, by simply raising the server public key to the 1337-th power, mod **p**.

The Master Secret is computed - according to TLS specs - as PRF(premaster_secret, "master secret", Client Random + Server Random)[0..47], the "+" being string concatenation, and PRF defined someplace else in the same RFC.

Luckily we have all of those, and there is a compliant implementation of the PRF function in the tlslite python library. Plugging in the appropriate values, the master secret is obtained:

```
4B02C246E50DE1CEA408018AE53F3C78386356A3D4C196E2FC9DE58079F5C57ED4698925E5BE507E315304A81B8AF2AC
```

After creating a master key logfile for Wireshark to consume, the data can be successfully decrypted:



Links

Dragon Sector \
[dragonsecto

Contact

contact@dragons

Archives

► 2017 (6)
► 2016 (4)
► 2015 (4)
▼ 2014 (25)
► December (2)
► October (1)
► June (1)
► May (1)
► April (7)
▼ March (4)
Exploits for recent pwning CTF tas published
Insomni'hack CTF Life is even har (Hardw...
Update: Dragon S TOP1 at Insomn 2014 in...
RuCTF 2014 Qua TLS (Crypto 30
► February (8)
► January (1)
► 2013 (22)

Popular Posts

CONFidence DS Tease registration is open!
Without further ado: http://ctf.dragonsector.p will start on the 26 th of A.M. CEST (GMT+2) - f

0CTF 2017 - EasiestPri
The task, as the name i rather basic (at first glar was a plot twist) format short 32-bit Debian app

EKOPART Malware s 400)
In short, th category 4 challenge was a journey

Since **RUCTF_ILoveToHackTLS** was indeed the correct flag, this concludes the write-up.

Posted by Tomasz Dubrownik at 17:33

3 comments:

VnSpl0it 11 March, 2014 17:21

Hi,

Could you please to upload the challenge file ?

Thanks

Reply

Tomasz Dubrownik 12 March, 2014 13:14

https://drive.google.com/file/d/0Bz8mM2W3uK9nR2VLa1Z6cUdUdkk/edit?usp=sharing ← here you go. Additionally a message from the organizers: “we will publish all our tasks on Github (https://github.com/Hackerdom) after Quals Afterparty.”

Reply

bay 13 March, 2014 17:11

Hi! Nice review!

Here is my solution: https://alexbers.com/ructf2014_quals/crypto300.py

---
Alexander Bersenev, author of this task

Reply

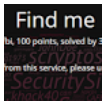
Enter your comment...

Comment as: Select profile...

Publish Preview

Newer Post Home Older Post

negligible x86-64 boiler
leading through a some



EKOPART
FBI 100
The FBI ca
something
personally
seen on a CTF (though
I did have a rather long
FBI ta...

0CTF 201
complicate
177)
Complicat
client-side
task revolving around, v
At the very start you we
way to XSS the adm...

0CTF 2017 - char (shell
The code in the "char" t
rather simple - you get t
2400 bytes of input (usi
"%2400s", so ...

ASIS CTF
Chessboa
175)
This is a ta
ASIS CTF
"Stego" (steganography
and it was solved by 2 t
including ours. This ...

0CTF 2017 - UploadCe
523)
Welcome to another Me
right~ Here you can use
as you wish No more w
begin 1 :) Fill your inforr

UFO CTF
brokoli (fo
This is a ta
CTF 2013
sweet mix
format stegano, forensic
decoding weird alphabe
that'...

Nuit du Ha
2014 - Nib
The recen
CTF Quals
mostly wel
forensics-oriented, with
explicitly categorized as

Contributors

- Adam Iwaniuk
- Gynvael Coldwind
- Jagger
- Krzysztof Katowicz-K
- Lympho Cytus
- Marcin Kalinowski
- Mateusz P
- Michał Kowalczyk
- Tomasz Bukowski
- Tomasz Dubrownik
- j00ru
- q3k
- valis

Mathjax