The SSL/TLS protocol

Myrto Arapinis School of Informatics University of Edinburgh

October 27, 2016

イロン イヨン イヨン イヨン 三日

1/27

SSL/TLS protocol

Goals: Confidentiality, Integrity, Non repudiation



SSL/TLS use X.509 certificates and hence asymmetric cryptography to exchange a symmetric key. This session key is then used to encrypt subsequent communication. This allows for **data/message confidentiality**, and message authentication codes for **message integrity** and thus, **message authentication**.

SSL/TLS protocol



Google

One account. All of Google.

Sign in to continue to Gmail

	Myrto	Arapi	nis	
	nyrto.arapi	nis@gr	nail.com	
Passw	ord			
	S	ign in		
Need hel	?			

Manage accounts on this device

One Google Account for everything Google



SSL/TLS protocol





TCP/IP protocol stack



- TCP/IP provides end-to-end connectivity and is organized into four abstraction layers which are used to sort all related protocols according to the scope of networking involved
- The SSL/TLS library operates above the transport layer (uses TCP) but below application protocols



SSL/TLS handshake protocol



💰 W	ireshark Fil	le Edit	View G	o Capt	ture Anal	yze Statis	ics Tel	lephony	Wireless	Tools	Help		S - 3	* 🛜	🔶 😸	52% 💽	Wed 22:47	Q :	Ξ
00								🚺 tist	Smail.pcape	1g									12 ²⁰
	10	-	• • •	- O	-			- 11=	E A	0 6									
		_ 8		- 1		1 1 1	· 👱	-		~ ~	· II								
Apply a	display filter .	<第/>															Expr	ession	+
No.	Time	Source			Destination		Protocol	Length	Info										
- 130	5.946868310	172.	16.76.158		172.217.2	3.45	TCP	74	35638→44	3 [SYN]	Seq=0	Win=2926	00 Len=0 M	ISS=1460	SACK_P	ERM=1 TSv	al=2158060	TSecr	
131	5.977374146	5 172.	217.23.45		172.16.76	. 158	TCP	60	443-3563	B [SYN,	ACK]	Seq=0 Acl	<=1 Win=64	1240 Len:	=0 MSS=	L460			•
132	5.977515105	5 172.	16.76.158		172.217.2	3.45	TCP	54	35638-+44	3 [ACK]	Seq=1	Ack=1 W:	LN=29200 L	.en=0					
133	5.978337850	172.	16.76.158		172.217.2	3.45	TLSv1.	. 261	Client H	ello									
134	5.9/9058/58	5 1/2.	217.23.45		1/2.16.76	.158	TCP	60	443+3563	B [ACK]	Seq=1	ACK=208	Win=64248	J Len=0					
1.125	Cipher Su:	ites Len	gth: 30		177 16 76	15.0	in sur	1913	SAFUAP IL										
	Cipher Su:	ites (15	suites)																
	Cipher	Suite: '	TLS_ECDHE_	ECDSA_W	ITH_AES_12	28_GCM_SHAD	56 (Øxc	02b)											
	Cipher	Suite: '	TLS_ECDHE_	RSA_WIT	H_AES_128	_GCM_SHA256	i (0xc02	f)											
	Cipher	Suite: '	TLS_ECDHE_	ECDSA_W	ITH_CHACHA	20_POLY130	15_SHA25	6 (Øxcca	9)										
	Cipher	Suite: '	TLS_ECDHE_	RSA_WIT	H_CHACHA2	POLY1305_	SHA256	(Øxcca8)											
	Cipher	Suite:	TLS_ECDHE_	_ECDSA_W	ITH_AES_25	56_GCM_SHA3	184 (Øxc	02c)											
	Cipher	Suite:	TLS_ECDHE_	RSA_WIT	H_AES_256	_GCM_SHA384	(0xc03	(8)											
	Cipher	Suite:	TLS_ECDHE_	_ECDSA_W	ITH_AES_2	56_CBC_SHA	(0xc00a)											
	Cipher	Suite:	TLS_ECDHE_	ECDSA_W	ITH_AES_1	28_CBC_SHA	(0xc009	0											
	Cipner	Suite:	ILS_ECOHE_	RSA_WIT	H_AES_128	_CBC_SHA (XC013)												
	Cipner	Suite:	ILS_ECONE_	RSA_WIT	H_AES_200	_CBC_SHA (XC014)												
	Cipher	Suite:	TLS_DHE_RS	M_WITT	AE5_126_C	C_SHA (0x)	(220)												
	Cipher	Suite:	TLC_DCA_WI	N_WIIN_	AE5_200_C		(659)												
	Cipher	Suite:	TIC DCA WI	TH ALS_	256 CBC_5	A (0x0021)													
	Cipher	Suite	TIS RSA WI	TH 3DES	EDE CBC	HA (8y888:													
	Conoressie	on Metho	de Length	· 1			"												
	Compression	on Metho	ds (1 meti	hod)															
	Extension	s Length	: 127																
	Extension:	server	name																
	Extension:	: Extend	ed Master	Secret															
	Extension:	: renego	tiation_i	nfo															
	Extension:	: ellipt	ic_curves																
	Extension:	: ec_poi	nt_format:	s															
	Extension:	: Sessio	nTicket T	LS															
	Extension:	: next_p	rotocol_n	egotiati	ion														
0000 70	Extension:	Annlic	ation Lav	er Prote	ocol Negot	iation													
0030 /2	10 DO 96 08	0 00 16 7 7 7 7 a	5d 98 42	80 53 1	00 00 CO 0	49:~	1 .B	. []											
0050 23	87 d6 28 26	dd 45	fe 94 6c	41 65 2	a 39 0b 4	f #(.)	lAe*	9.0											
0 7	Secure Sockets La	ayer (ssl), 2	07 bytes									8	ackets: 475 -	Displayed:	475 (100.	249 - Load ti	me: 0:0.11 F	rofile: Del	fault

💰 Wireshark File	e Edit View Go C	apture Analyze Stat	stics Telephony	Wireless Too	ls Help	5 · · · · · · · · · · · · · · · · · · ·	> 10 52% 10 Wed 22:	47 Q 📰
00			📑 tl	Gmail.pcapng				2
🧉 📕 🧕 🔘	S 🕺 🗋 🚍	۹ 🗢 🛸 🖀 🤅	F 🗶 🖵 🛛	⊕ ⊖	् 🎞			
Apply a display filter	<\$%/>						📑 * B	xpression +
No. Time	Source	Destination	Protocol Lengti	Info				
132 5.977515105	172.16.76.158	172.217.23.45	TCP 5	1 35638→443 [ACI	<] Seq=1 Ack=1	Win=29200 Len=0		
133 5.978337850	172.16.76.158	172.217.23.45	TLSv1 26	l Client Hello				•
134 5.979058758	172.217.23.45	172.16.76.158	TCP 6	9 443→35638 [AC	<] Seq=1 Ack=20	8 Win=64240 Len=0		
135 6.015031282	1/2.21/.23.45	1/2.16./6.158	TLSV1 281	Server Hello	()	2764 1445 22500 1 15 0		
130 0.015054025	1/2.10./0.158	1/2.21/.23.45	red (22512 bits	i 35038⇒443 [AU) on interface	Seq=208 ACK=	2701 W1n=33580 Len=0		
Ethernet II. Src:	Vmware f0:7d:d2 (00:	50:56:f0:7d:d2). Dst:	Vmware 9e:08:02	(00:0c:29:9e:0	8:07)			
Internet Protocol	Version 4. Src: 172.2	217.23.45. Dst: 172.1	.76.158	. (0010012515010	0.01)			
▶ Transmission Contr	rol Protocol, Src Port	t: 443, Dst Port: 356	88. Seg: 1. Ack	208, Len: 2760				
Secure Sockets Lay	/er							
TLSv1.2 Record	Layer: Handshake Prot	ocol: Server Hello						
Content Type:	: Handshake (22)							
Version: TLS	1.2 (0x0303)							
Length: 76								
Handshake Pro	otocol: Server Hello							
Handshake	Type: Server Hello (2	2)						
Length: /2								
Version: I	LS 1.2 (0X0303)							
F Ranuoli	Longth: 0							
Cinher Sui	te: TIS FOOHE DSA WIT	TH AFS 128 GCM SHA256	(8xc82f)					
Compressio	m Method: pull (8)	1_ALS_110_0C1_51AL50	(UNCOLT)					
Extensions	Length: 32							
▶ Extension:	renegotiation info							
▶ Extension:	server_name							
▶ Extension:	Extended Master Secr	et						
Extension:	SessionTicket TLS							
Extension:	Application Layer Pr	otocol Negotiation						
▶ Extension:	ec_point_formats							
0050 1a r0 C/ 9/ 00 0040 03 58 11 20 c9	d4 fa 52 3c d5 63 f	2 00 00 40 03						
0050 55 6e 3b 4c ea	4c 6f 28 71 0e 38 4	2 1b 8e b6 c7 Un;L.	Lo(q.88					
0060 42 00 c0 2f 00	00 20 ff 01 00 01 0	0 00 00 00 00 B/.						
00/0 00 17 00 00 00 0000 22 00 05 00 02	23 00 00 00 10 00 0	15 00 03 02 68	#h					
0090 00 0c 23 00 04	a5 30 82 04 a1 30 8	12 03 89 a0 03	.00					
🔘 🎢 Secure Sockets La	yer (ssl), 2760 bytes					Packets: 475 - Displayed: 475	(100.0%) - Load time: 0:0.11	Profile: Default

🗋 🗰 Wireshark File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help 🛛 🔂 🛞 🕴 🖘 🕫 🕫	% 💽 Wed 22:48 🔍 🔚
e e e filos	20
◢ ■ ∅ ◎ 🚍 🖹 🛛 🭳 🗢 ⇒ ≌ 💿 🖢 📮 🧮 ۹. ۹. ۹. ۳	
Apply a display filter <%/>	Expression +
No. Time Source Destination Protocol Length Info	
134 5.979658758 172.217.23.45 172.16.76.158 TCP 60 433-35638 [ACK] Seq=1 Ack=208 Win=64240 Len=0	
135 0.0130031202 172.217.23.45 172.10.76.150 172.10.70.138 ILSVI 2014 Server Hello	•
130 0.0130-0025 172.10.00.10 172.27.03.45 172.16.76 158 TI SV1 841 CertificateServer Key Exchange Server Hello Done	
138 6.015126370 172.16.76.158 172.217.23.45 TCP 54 35638-443 [ACK] Seq=208 Ack=3548 Win=36500 Len=0	
serialNumber: 7377627938644829374	
▶ signature (sha256WithRSAEncryption)	
▶ issuer: rdnSequence (0)	
▶ validity	
Subject: ransquerce (0)	
alocitistic (selection)	
subjectPublicKey: 3082010a0282010100aa00c2a0f111bb011132301a5fcdfd	
modulus: 0x00aa00c2a0f111bb011132301a5fcdfdff7762a8fc0fd60a	
publicExponent: 65537	
extensions: 8 items	
V algorithmidentifier (sha25eWithKSAEncryption) Algorithm Id. 1.2 (200 11240 1.1.1) (sha25eWithDSAEncryption)	
0130 aa 00 c2 a0 f1 11 bb 01 11 32 30 1a 5f cd fd ff	
0140 77 62 a8 fc 0f d6 0a 85 67 fe ef cb f7 93 4e 9a wbgN.	
0150 C2 49 DC D5 BC 35 D D/ D/ 47 CD B/ AD DB TB 28 09	
0170 ee 93 af 17 0c b3 51 88 54 f8 86 71 bb 73 df b4Q. Tq.s	
0180 cf 0e 3a c1 ab 72 f9 9e 88 78 26 5b a4 f7 d8 0dr. x&[
01b0 ab 5b 37 e4 97 c0 42 d6 00 48 3d 15 64 0b 76 7c .[7BH=.d.v]	
0100 b3 8b d9 16 41 3b 4b 0d a5 f7 d8 76 d8 7c e7 4bA;Kv. .K	
0100 03 3e 31 38 34 1e 32 35 TO 4C 90 30 01 TE / T 90	
01f0 75 37 2b c3 52 72 7c b6 5e be f5 ed 16 d0 bc 7c u7+.Rrj. ^	
0200 ef 09 27 b5 9e 57 46 e2 f8 2c d3 ed 4a 14 7a 57J.zw	
6220 d 5 4 5 2 9 9 00 00 c 92 36 47 22 0 c c 1 9 bd 66"	
0230 02 03 01 00 01 a3 82 01 67 30 82 01 63 30 1d 06 g0c0	
0240 03 55 10 25 04 10 30 14 00 08 20 06 01 05 05 07	
Frame (841 bytes) Reassembled TCP (31)9 bytes)	
○ 2 INTEGR (pkcs1.modulus), 257 bytes Packets: 475 - Displayed: 475 (100.0%) -	Load time: 0:0.11 Profile: Default

www.gmail.com's certificate

from the	https website accounts.google.com.		
GeoTrust Global CA		1	
Google Interne Google Interne Google Interne	t Authority G2 poole com		
	c		
Common Name	Google Internet Authority G2		
Serial Number	7377627938644829374		
Version	3		
Signature Algorithm	SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)		
Parameters	none		
Not Valid Before	Thursday, 6 October 2016 13:59:57 British Summer Time		
Not Valid After	Thursday, 29 December 2016 12:28:00 Greenwich Mean Time		
Public Key Info			
Algorithm	RSA Encryption (1.2.840.113549.1.1.1)		
Public Key	256 bytes : AA 00 C2 A0 F1 11 BB 01 11 32 30 1A 5F CD FD FF 77 62		
	A8 FC 0F D6 0A 85 67 FE EF CB F7 93 4E 9A CE 49 BC D5 8C 3B 67 B7 4F C6 A7 AB DA F8 E8 04 8A 89 C6 DA 99 EC 3D 42 8C 0E C0 86 0C		
	C4 25 E3 EE 93 AF 17 0C B3 51 88 54 F8 86 71 BB 73 DF B4 CF 0E 3A		
	38 DE BD 15 A5 54 BE 78 C7 AF 1A CD 3B 71 07 AA EC 2E FB 18 EE FE		
	78 16 AB 5B 37 E4 97 C0 42 D6 00 48 3D 15 64 0B 76 7C B3 8B D9		
	4C 9D 30 01 FE 7F 90 61 2A EC A3 D1 28 7D 57 1C 1F A7 E1 48 DF		
	65 02 75 37 28 C3 52 72 7C 86 5E 8E F5 ED 16 D0 8C 7C EF 09 2F 85		
	GA 7E E3 E1 B6 27 D7 B5 44 52 99 0B 0B 0C 92 36 47 22 0C E7 19 BD		
Exponent	65537		
Key Size	2048 bits		

🐞 Wireshark File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help	🛂 🕙 🕴 🤶 🚸 🕮 46% 💽 Wed 23:02 Q, 🔚
⊖ ⊖ ☐ tlsGmail.pcapng	× ⁿ
🔺 🔳 🖉 🐵 🚞 🖹 🖉 🍳 🗢 🔶 🚟 🖉 💆 🚍 🔍 Q. Q. Q. 🎞	
Apply a display filter <\$/>	Expression +
No. Time Source Destination Protocol Length Info	
136 6.015054625 172.16.76.158 172.217.23.45 TCP 54 35638+443 [ACK] Seq=200	3 Ack=2761 Win=33580 Len=0
137 6.015117278 172.217.23.45 172.16.76.158 TLSv1 841 CertificateServer Key B	Exchange, Server Hello Done 🛛 👘
138 6.015126370 172.16.76.158 172.217.23.45 TCP 54 35638-443 [ACK] Seq=208	3 Ack=3548 Win=36500 Len=0
139 6.017904477 172.16.76.158 172.217.23.45 TLSv1 180 Client Key Exchange, Ch	hange Cipher Spec, Hello Request, Hello Request
140 6.019370071 172.16.76.158 172.217.23.14 OCSP 491 Request	
 Secure Sockets Layer Secure Sockets Layer 	
ILSV1.2 Record Layer: Handshake Protocol: Client Key Exchange	
Content Type: Handshake (22)	
Version: (LS 1.2 (0x0305)	
w Handsha Protocol: Client Key Evchanne	
Handshake Twee: Client Key Exchange (16)	
Length: 66	
EC Diffie-Hellman Client Params	
Pubkey Length: 65	
Pubkey: 04580c88228565d38a865aa51d1c08e2d75d731d40c5b5b7	
▶ TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec	
▶ TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages	
0000 00 50 56 10 7d d2 00 0c 29 9e 08 02 08 00 45 00 .PV.}E.	
0010 00 a6 c3 9a 40 00 40 06 ba 02 ac 10 4c 9e ac d9 8.0L	
0030 8/ 20 00 30 01 00 01 0/ 21 49 22 21 09 12 30 10071.1.FF	
0040 04 58 0c 88 22 85 65 d3 8a 86 5a a5 1d 1c 08 e2 .X".eZ	
0050 d7 5d 73 1d 40 c5 b5 b7 04 1c cf 0e 2f d8 77 71 .]s.@/.wq	
0060 de 13 74 92 b7 83 5d 9f 0b 01 2c 00 4b a5 e1 30t], K0	
0070 3C 90 C7 99 30 21 C3 81 14 43 03 04 20 1C 30 35 (07.,	
0090 00 00 00 00 a8 98 79 12 f3 90 21 a7 d6 24 cc ddy!.\$	
00a0 16 37 73 75 62 63 4c 07 10 6c f2 83 c2 34 a3 71 .7subcLl4.q	
00b0 da 81 62 e8b.	
O 🍸 EC Diffie-Hellman client pubkey (ssl.handshake.client_point), 65 bytes	Packets: 475 - Displayed: 475 (100.0%) - Load time: 0:0.11 Profile: Default

🐞 Wireshark File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help	🖪 🕙 🕴 🛜 🔶 🎟 46% 💽 Wed 23:02	Q∷≣
\varTheta 🖯 🕤		× ²
🖌 📕 🖉 🐵 🚞 🖹 🕱 🙆 🔍 🗢 🛸 🖉 🖉 💆 🛄 🔍 Q. Q. 🎹 👘		
Apply a display filter <%/>	Express	ion +
No. Time Source Destination Protocol Length Info		
136 6.015054625 172.16.76.158 172.217.23.45 TCP 54 35638-443 [ACK] Seq=208 Ack	=2761 Win=33580 Len=0	
13/ 5.015117/78 172.15.45 172.16.58 172.21.58 ILSVI 841 CertificateServer Key Excha	nge, Server Hello Done -3549 Win-36500 Len-0	•
136 0.013120370 172.10.70.136 172.217.23.45 TLSU 180 Client Key Exchange Change Change	Cipher Spec. Hello Request. Hello Request	
149 6.019370071 172.16.76.158 172.217.23.14 OCSP 491 Request	expirer spec, necto nequest, necto nequest	
Length: 70		
# Handshake Protocol: Client Key Exchange		
Handshake Type: Client Key Exchange (16)		
Length: bb		
Publicy Lengths 55		
Publey: 04580c88228565d38a865aa51d1c08e2d75d731d40c5b5b7		
▼ TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec		
Content Type: Change Cipher Spec (20)		
Version: TLS 1.2 (0x0303)		
Length: 1		
Change Cipher Spec Ressage		
P 103912 Record Layer, Handshake Protocott, Hattigter Handshake Hessages		
0010 00 a5 c3 9a 40 00 40 05 ba 02 ac 10 4c 9e ac d9		
0020 17 2d 8b 36 01 bb 8f a7 2f 49 2e 21 d9 12 50 186 /I.!P.		
0030 86 94 00 40 00 00 10 03 03 00 40 10 00 00 42 41, M, FBA		
0050 d7 5d 73 1d 40 c5 b5 b7 04 1c cf 0e 2f d8 77 71 .ls.@/.wq		
0060 de 13 74 92 b7 83 5d 9f 0b 01 2c 00 4b a5 e1 30t], K0		
0000 5C 90 C/ 99 30 21 C3 81 T4 43 05 04 20 1C 50 33 (b/t.)		
0090 00 00 00 00 a8 98 79 12 f3 90 21 a7 d6 24 cc ddy!.\$		
00a0 16 37 73 75 62 63 4c 07 10 6c f2 83 c2 34 a3 71 .7subcLl4.q		
O 7 Record Layer (ssl.record), 6 bytes	Packets: 475 - Displayed: 475 (100.0%) - Load time: 0:0.11 Pro	file: Default

miTLS, Triple Handshake, SMACK, FREAK, Logjam, and SLOTH												
i mitls.org	/pages/atta	cks										
SimSec 🔻	CSexam ▼	La cryptograts de	ivoilés Co	nferences 🔻	Res	earchProfiles 🔻	Security-Club =	Teaching 🔻	Tutoring			
		TLS miTLS	Publicat	ions Att	acks	Code	FlexTLS	People				
			Alert	3SHAKE	VHC	SMACK L	ogjam SLOTH					
		Protocol	Cryptographic	Implementation	on	Deployment						

A Zoo of TLS attacks

Attacks on TLS that break the intuitive security property of a virtual recreation of a physically secure channel can be categorized along three dimensions.

- 1. Protocol logic vs. cryptographic design flaw
- 2. Specification/Standard vs. Implementation errors
- 3. TLS vs. Context

Flaws in the protocol logic

Attacks targeting the protocol logic may for instance cause the client and server to negotiate the use of weak algorithms even though they both support strong cryptography.

If the faulty regolation logic conforms to the specification, then the attack is on the specification test (as, e.g., spriatly enabled by the False Start modification), it al implementation can be specificated by the start of the start and enable of the start of th

Another class of protocol logic flaws are state-machine bugs [Early CCS Attack, SMACK Attack].

The attack can also be either an attack on TLS proper, or on its context, e.g. if the attacker can just change the configuration files to deactivate strong cryptography. As the TLS standard does not describe APIs or configuration file formats, context specific attacks are always implementation specific.

The receptiation attack [TLS_Reneg_Attack] is a logical attack on the TLS standard, where one peer believes it is running the first handshake on a connection, while the other peer is running a re-handshake. miTLS prevents the renegotiation attack by implementing the renegotiation extension.

More generally, the TLS specification is vague about how applications should handle data coming from consecutive sessions, e.g., whether it is asket to join them and consider them as a single stream, of it the user should be notified of the change of context. The renegotiation extension partially ites the problem, but it still leaves room for our elert attack, where the attacker can turn any authentic fatal attacts in to a averning atter, which get signatored by default.

Much not seriously, resuming the attacker controlled assess on a a different connection re-analyses the renegotation state. This table is a series of the attacker and the series of the attacker and the series of the series of

Cryptographic design flaws

Attacks exploiting cryptographic design faws may simply result from cryptanahytic progress against the cryptographic building blocks of U.S. Threy can, however, all one result from improper non-blackbox use of otherwise secure cryptographic constructions. An example for this is chosen ciphertext chaining (CBC) mode encrypton. Early versions of U.S. allow using trowledge of the next inflatization vectors (V1) to set up adaptive plaintext attacks, see, e.g. Q. penests. Lurchive for a first mention of the SEAT attack: ≣ •∕০৭ে 14/27 Client and server are allowed to initiate renegotiation of the session encryption in order to:

- Refresh keys
- Increase authentication
- Increase cipher strength
- ▶ ...

Client or server can trigger renegotiation by sending a hello meesage

イロト 不得 とくほと くほとう ほ

15 / 27

SSL/TLS renegotiation weaknesses

- Renegotiation has priority over application data!
- Renegotiation can take place in the middle of an application layer transaction!



(Detailed on the board)

Incorrect implicit assumtion: the client doesn't change through renegotiation

◆□ → < □ → < Ξ → < Ξ → < Ξ → Ξ → ○ Q ○ 17/27

Attacker:

GET /pizza?toppings=pepperoni;address=attacker_str HTTP/1.1
X-Ignore-This:(no carriage return)

Attacker:

```
GET /pizza?toppings=pepperoni;address=attacker_str HTTP/1.1
X-Ignore-This:(no carriage return)
```

Victim: GET /pizza?toppings=sausage;address=victim_str HTTP/1.1 Cookie:victim_cookie

```
Attacker:
GET /pizza?toppings=pepperoni;address=attacker_str HTTP/1.1
X-Ignore-This:(no carriage return)
```

```
Victim:
GET /pizza?toppings=sausage;address=victim_str HTTP/1.1
Cookie:victim_cookie
```

Result: GET /pizza?toppings=pepperoni;address=attacker_str HTTP/1.1 X-Ignore-This:GET /pizza?toppings=sausage;address=victim_str HTTP/1.1 Cookie:victim_cookie

```
Attacker:
GET /pizza?toppings=pepperoni;address=attacker_str HTTP/1.1
X-Ignore-This:(no carriage return)
```

```
Victim:
GET /pizza?toppings=sausage;address=victim_str HTTP/1.1
Cookie:victim_cookie
```

Result: GET /pizza?toppings=pepperoni;address=attacker_str HTTP/1.1 X-Ignore-This:GET /pizza?toppings=sausage;address=victim_str HTTP/1.1 Cookie:victim_cookie

\Rightarrow Server uses victim's account to send a pizza to attacker!

<ロ > < 回 > < 画 > < 直 > < 直 > < 直 > 三 の Q (C) 18/27

Twitter status updates using its API by posting the new status to http://twitter.com/statuses/update.xml, as well as the user name and password

Twitter status updates using its API by posting the new status to http://twitter.com/statuses/update.xml, as well as the user name and password

```
Attacker:

POST /statuses/update.xml HTTP/1.1

Authorization: Basic username:password

User-Agent: curl/7.19.5

Host: twitter.com

Accept:*/*

Content-Length: 140

Content-Type: application/x-www-form-urlencoded

status=
```

Twitter status updates using its API by posting the new status to http://twitter.com/statuses/update.xml, as well as the user name and password

```
Attacker:
POST /statuses/update.xml HTTP/1.1
Authorization: Basic username:password
User-Agent: curl/7.19.5
Host: twitter.com
Accept:*/*
Content-Length: 140
Content-Type: application/x-www-form-urlencoded
status=
Victim:
POST /statuses/update.xml HTTP/1.1
Authorization: Basic username:password...
```

Twitter status updates using its API by posting the new status to http://twitter.com/statuses/update.xml, as well as the user name and password

```
Attacker:
POST /statuses/update.xml HTTP/1.1
Authorization: Basic username:password
User-Agent: curl/7.19.5
Host: twitter.com
Accept:*/*
Content-Length: 140
Content-Type: application/x-www-form-urlencoded
status=
Victim:
POST /statuses/update.xml HTTP/1.1
Authorization: Basic username:password...
```

 \Rightarrow the attacker gets the user name and password of the victim!

The SAML Signle Sign On (SSO) protocol

SAML SSO protocol



SAML SSO protocol

Chrome File Edit View Histor	y Bookmarks	Window Help						P - 9	* ? *) 🌐 100%	í⊂∎i Thu	13 Feb 00:	50 Q.	:=
\varTheta 🔿 🕤 🛛 📾 BBC - Sign in 🛛 🗙														H ₂₁
← → C 🔒 https://ssl.bbc.co.uk/id/si	gnin												Qź	3 =
	BBC O	Sign in Now	s Sport	Weather	IPlayor	TV	More -	Search	c	۹.				
		N IN BBC il	D											
	Don't have a BBC ID? Please register. Email or username Password Final variagement?						Other ways to sign in Vorthat signed its to be BIC to 25 days. If reading the signed of the signed its to be BIC to 25 days. Place only use based space and the signed its to errower. We want the are preset your atkinky to Frankbook or Groups Debut BBC ID							
	Remember r	Remember me Untick if you're using a shared computer. Sign in Cancel						r and easily to o more nformation secu	romment, add rely, and we mission.					
							Spam-free We'll only send BBC ID help	you emails if yo	u ask for them.					
https://sil.bbc.co.uk/id/statecookie/google.com	BBC iWonde	Pack I beco f WW	Up Your me the 1? TV History	Radi	Gareth Malone explains why the song was a success o	, 								

SAML SSO protocol



SAML SSO protocol (OASIS 2005)



Google's SAML-based Single Sign-On for Google Applications deviates from the above protocol for a few, seemingly minor simplifications in the messages exchanged:

- G1. ID and SP are not included in the authentication assertion, i.e. AA = AuthAssert(C; IdP) instead of AuthAssert(ID; C; IdP; SP);
- G2. *ID*, *SP* and *IdP* are not included in the response, *i.e.* $Resp = \text{Response}(\{AA\}_{K_{ldP}^{-1}})$ instead of $\text{Response}(ID; SP; IdP; \{AA\}_{K_{ldP}^{-1}}).$

Attack Google's SSO implementation

[A. Armando, R. Carbone, L. Compagna, J. Cullar, L. Tobarra, "Formal analysis of SAML 2.0 web browser single sign-on: breaking the SAML-based single sign-on for google apps", (FMSE'08)]



SAML SSO protocol (OASIS 2012)



◆ □ → < ≥ → < ≥ → ≥ </p>

9 < ○
26 / 27
</p>

Attack SAML SSO protocol (OASIS 2012)

[A. Armando, R. Carbone, L. Compagna, J. Cullar, G. Pellegrino, A. Sorniotti, "From Multiple Credentials to Browser-Based Single Sign-On: Are We More Secure?", Chapter in Future Challenges in Security and Privacy for Academia and Industry]



 \Rightarrow XSS attack on SAML-base SSO for Google Apps