The strongSwan IPsec Solution with TNC Support

TCG Members Meeting June 2011 Munich

Prof. Dr. Andreas Steffen
Institute for Internet Technologies and Applications
HSR University of Applied Sciences Rapperswil
andreas.steffen@hsr.ch
Where the heck is Rapperswil?
HSR - Hochschule für Technik Rapperswil

- University of Applied Sciences with about 1500 students
- Faculty of Information Technology (300-400 students)
- Bachelor Course (3 years), Master Course (+1.5 years)
The strongSwan IPsec Solution with TNC Support

TCG Members Meeting June 2011 Munich

IKEv2 Open Source Implementation
strongSwan Usage Scenarios

- strongSwan is an **Internet Key Exchange Daemon** responsible for automatically setting up IPsec-based VPN connections.
The FreeS/WAN Genealogy

1999
FreeS/WAN 1.x

2000
X.509 1.x Patch
Super FreeS/WAN

2003
Openswan 1.x

2004
Openswan 2.x
FreeS/WAN 2.x
X.509 2.x Patch

2005
Openswan 2.x
strongSwan 2.x
IKEv2 RFC 4306

2006
IKEv1 & partial IKEv2

2007
IKEv1 & IKEv2

ITA IKEv2 Project
New architecture, same config.
IKEv2 Interoperability Workshops

Spring 2007 in Orlando, Florida
Spring 2008 in San Antonio, Texas

- **strongSwan** successfully interoperated with IKEv2 products from Alcatel-Lucent, Certicom, CheckPoint, Cisco, Furukawa, IBM, Ixia, Juniper, Microsoft, Nokia, SafeNet, Secure Computing, SonicWall, and the IPv6 TAHI Project.
strongSwan Key Customers

• Alcatel-Lucent, Clavister, Ericsson, Nokia Siemens Networks, Ubiquisys
  • Femtocells/Security Gateways for GSM/UMTS/LTE Mobile Networks

• Astaro
  • Astaro Security Gateway

• Secunet
  • SINA Box for High Security Applications (German Federal Government)

• U.S. Government
  • Open Source IKEv2/IPsec Reference and Test System for Suite B Elliptic Curve Cryptography
strongSwan Business Model

- **Paid development of customer-specific add-ons**
  - Features of general interest are released back into the main strongSwan distribution under the GPLv2 open source license

- **Commercial licensing of the HSR-owned IKEv2 source code**
  - Licensee is **not** obliged to disclose any proprietary modifications and add-ons to the IKEv2 strongSwan source code.
Supported Platforms

• **Operating Systems**
  • Linux
  • Android
  • FreeBSD
  • Mac OS X

• **Hardware Platforms (32/64 bit)**
  • Intel, Via, AMD
  • ARM, MIPS (e.g. Freescale, Marvell, 16-core Cavium Octeon)
  • PowerPC

• **Networking Stack**
  • IPv4
  • IPv6 (SuSE Linux Enterprise with strongSwan certified by DoD in 2008)
  • Mobile IPv4/IPv6

• **Portable Source Code**
  • 100% written in C but with an **object-oriented** modular approach
  • Performance scalability through extensive use of **multi-threading**
What about Windows?
Windows 7 VPN with Machine Certificates

- Microsoft tested IKEv2 interoperability using strongSwan right up to the final Windows 7 release.
Windows 7 VPN with EAP Authentication

- UsingIKEv2 EAP-MSCHAPv2 or EAP-TLS with smartcards
strongSwan Applet for the Linux Desktop

- D-Bus based communication.
strongSwan in a Mixed VPN Environment

Windows Active Directory Server

Linux FreeRadius Server

Corporate Network

High-Availability strongSwan VPN Gateway

Internet

Windows 7 Agile VPN Client

strongSwan Linux Client
IKEv2 Authentication Methods

• Based on Public Keys
  • X.509 certificates with RSA or ECDSA keys
  • PKCS#11 smartcard interface
  • CRLs via HTTP/LDAP, OCSP

• Based on Pre-Shared Keys (PSK)
  • Arbitrary PSK length, beware of weak secrets!

• Based on the Extended Authentication Protocol (EAP)
  • EAP-MD5, EAP-MSCHAPv2, EAP-GTC
  • EAP-SIM, EAP-AKA (GSM/UMTS/CDMA2000)
  • EAP-TLS, EAP-TTLS, EAP-PEAPv0

• Interface to AAA Server
  • EAP-RADIUS

• EAP and TNC Methods implemented as Plugins
  • strongSwan IKEv2 daemon loads plugins at run-time
The strongSwan IPsec Solution with TNC Support

TCG Members Meeting June 2011 Munich

Trusted Network Connect Capabilities
strongSwan as a TNC client and PEP

Network Access Requestor
Supplicant/VPN Client, etc.

Policy Enforcement Point
Switch/Firewall/VPN Gateway

TNC@FHH

EAP-TNC in EAP-TTLS
IF-TNCCS-1.1 Protocol on the TNC Client Side

13[TNC] sending TNCCS Batch (633 bytes) for Connection ID 1

13[TNC] <xml version="1.0"/>
13[TNC] <TNCCS-Batch BatchId="1" Recipient="TNCS"/>
13[TNC] <TNCC-TNCS-Message>
13[TNC] <Type>00000003</Type>
13[TNC] <XML>
13[TNC] <TNCCS-PreferredLanguage>en</TNCCS-PreferredLanguage>
13[TNC] </XML>
13[TNC] </TNCC-TNCS-Message>
13[TNC] <IMC-IMV-Message>
13[TNC] <Type>0080ab31</Type>
13[TNC] <Base64>RHVtbX1JTUMgbWVzcg4gPSBhGxvd==</Base64>
13[TNC] </IMC-IMV-Message>
13[TNC] </TNCCS-Batch>

13[IKE] sending tunneled EAP-TTLS AVP [EAP/RES/TNC]
13[ENC] generating IKE_AUTH request 7 [ EAP/RES/TTLS ]
13[NET] sending packet: from 192.168.0.100[4500] to 192.168.0.1[4500]

15[NET] received packet: from 192.168.0.1[4500] to 192.168.0.100[4500]
15[ENC] parsed IKE_AUTH response 7 [ EAP/REQ/TTLS ]
15[IKE] received tunneled EAP-TTLS AVP [EAP/REQ/TNC]
15[TNC] received TNCCS Batch (473 bytes) for Connection ID 1

15[TNC] <xml version="1.0"/>
15[TNC] <TNCCS-Batch BatchId="2" Recipient="TNCC"/>
15[TNC] <IMC-IMV-Message>
15[TNC] <Type>0080ab31</Type>
15[TNC] <Base64>RHVtbX1JTUMgbWVzcg4gPSBhGxvd==</Base64>
15[TNC] </IMC-IMV-Message>
15[TNC] </TNCCS-Batch>
TNC Policy Enforcement

Host with TNC Client

Policy Enforcement Point

Remediation Server

Isolation Network

TNC Server

Policy Manager

Corporate Network

Source: Jörg Vieweg, FH Hannover
strongSwan Configuration on the PEP side

conn rw-allow
 rightgroups=allow
 leftsubnet=10.1.0.0/16
 also=rw-eap
 auto=add

conn rw-isolate
 rightgroups=isolate
 leftsubnet=10.2.0.0/16
 also=rw-eap
 auto=add

conn rw-eap
 left=192.168.0.1
 leftcert=moonCert.pem
 leftid=@moon.strongswan.org
 leftauth=eap-ttls
 leftfirewall=yes
 rightauth=eap-radius
 rightid=*@strongswan.org
 rightsendcert=never
 right=%any
IF-PEP Protocol on the strongSwan PEP

05[CFG] received RADIUS Access-Accept from server '10.1.0.10'
05[IKE] received RADIUS attribute Tunnel-Type: tag = 0, value = 9
05[IKE] received RADIUS attribute Filter-Id: 'allow'
05[IKE] RADIUS authentication of 'carol@strongswan.org' successful
05[IKE] EAP method EAP_TTLS succeeded, MSK established
05[ENC] generating IKE_AUTH response 11 [ EAP/SUCC ]
05[NET] sending packet: from 192.168.0.1[4500] to 192.168.0.100[4500]
04[NET] received packet: from 192.168.0.100[4500] to 192.168.0.1[4500]
04[ENC] parsed IKE_AUTH request 12 [ AUTH ]
04[IKE] authentication of 'carol@strongswan.org' with EAP successful
04[IKE] authentication of 'moon.strongswan.org' (myself) with EAP
04[IKE] IKE_SA rw-allow[1] established between 192.168.0.1[moon.strongswan.org]...192.168.0.100[carol@strongswan.org]
02[CFG] received RADIUS Access-Accept from server '10.1.0.10'
02[IKE] received RADIUS attribute Tunnel-Type: tag = 0, value = 9
02[IKE] received RADIUS attribute Filter-Id: 'isolate'
02[IKE] RADIUS authentication of 'dave@strongswan.org' successful
02[IKE] EAP method EAP_TTLS succeeded, MSK established
02[ENC] generating IKE_AUTH response 11 [ EAP/SUCC ]
02[NET] sending packet: from 192.168.0.1[4500] to 192.168.0.200[4500]
01[NET] received packet: from 192.168.0.200[4500] to 192.168.0.1[4500]
01[ENC] parsed IKE_AUTH request 12 [ AUTH ]
01[IKE] authentication of 'dave@strongswan.org' with EAP successful
01[CFG] constraint check failed: group membership required
01[CFG] selected peer config 'rw-allow' unacceptable
01[CFG] switching to peer config 'rw-isolate,'
01[IKE] authentication of 'moon.strongswan.org' (myself) with EAP
01[IKE] IKE_SA rw-isolate[2] established between 192.168.0.1[moon.strongswan.org]...192.168.0.200[dave@strongswan.org]
Network Endpoint Assessment (RFC 5209)

- Posture Collectors (1 .. N)
- Posture Broker Client
- Posture Transport Clients (1 .. K)
  - PA (RFC 5792)
  - PA-TNC
  - PB (RFC 5793)
  - PB-TNC
- Posture Validators (1 .. N)
- Posture Broker Server
- Posture Transport Servers (1 .. K)
  - PT
  - Internet Drafts

NEA Client

NEA Server
strongSwan as a TNC client and TNC server
TNCCS-2.0 Protocol on the TNC Client Side

13[TNC] creating PB-PA message type 'ITA-HSR' 0x00902a/0x01
13[TNC] adding PB-PA message
13[TNC] PB-TNC state transition from 'Init' to 'Server Working'
13[TNC] sending PB-TNC CDATA batch (88 bytes) for Connection ID 1
13[TNC] => 88 bytes @ 0x8081044
13[TNC] 0: 02 00 00 01 00 00 00 58 00 00 00 00 00 00 00 06 ........X........
13[TNC] 16: 00 00 00 1F 41 63 63 65 70 74 - 4C 61 6E 67 75 ....Accept-Langu
13[TNC] 32: 61 67 65 3A 20 65 6E 80 00 00 00 00 00 00 01 00 age: en.........
13[TNC] 48: 00 00 31 00 00 90 2A 00 00 00 01 00 FF FF 01 ............. J
13[TNC] 64: 00 00 00 C1 2D 6F 80 00 90 2A 00 00 00 01 00 ......../.....
13[TNC] 80: 00 00 11 61 6C 6F 77 ...allow

13[IKE] sending tunneled EAP-TTLS AVP [EAP/RES/TNC]
13[ENC] generating IKE_AUTH request 7 [ EAP/RES/TTLS ]
13[NET] sending packet: from 192.168.0.100[4500] to 192.168.0.1[4500]

14[NET] received packet: from 192.168.0.1[4500] to 192.168.0.100[4500]
14[ENC] parsed IKE_AUTH response 7 [ EAP/REQ/TTLS ]
14[IKE] received tunneled EAP-TTLS AVP [EAP/REQ/TNC]
14[TNC] received TNCCS batch (58 bytes) for Connection ID 1
14[TNC] => 58 bytes @ 0x8080fee
14[TNC] 0: 02 80 00 02 00 00 00 3A 80 00 00 00 00 00 00 01 ........:.........
14[TNC] 16: 00 00 00 32 00 00 90 2A 00 00 00 01 FF FF 00 01 ........ J
14[TNC] 32: 01 00 00 00 2C 40 A0 6C 00 00 90 2A 00 00 00 01 .....@.1.....
14[TNC] 48: 00 00 00 12 72 65 70 65 61 74 ....repeat

14[TNC] PB-TNC state transition from 'Server Working' to 'Client Working'
14[TNC] processing PB-TNC SDATA batch
14[TNC] processing PB-PA message (50 bytes)
14[TNC] handling PB-PA message type 'ITA-HSR' 0x00902a/0x01
Current Work

- **TCG Certification of IF-IMC, IF-IMV, and IF-PEP Interfaces**
  - Participation at the TNC 2011 Spring PlugFest in Chantilly, VA
  - Passed IF-IMC and IF-IMV compliance test suites
  - IF-PEP layer 2 VLAN test suite must first be adapted for layer 3 VPN

- **IMC/IMV Test Pair with IF-M (RFC 5792 PA-TNC) Interface**
  - Available now as strongSwan developers release
  - Stable strongSwan 4.5.3 release expected in July 2011.

```
13[TNC] creating PA-TNC message with ID 0xc12ed62f
13[TNC] creating PA-TNC attribute type 'ITA-HSR' 0x00902a/0x00000001
13[TNC] => 5 bytes @ 0x808123c
13[TNC] 0: 61 6C 6C 6F 77 allow
13[TNC] creating PB-PA message type 'ITA-HSR' 0x00902a/0x01

14[TNC] handling PB-PA message type 'ITA-HSR' 0x00902a/0x01
14[TNC] processing PA-TNC message with ID 0x2c40a06c
14[TNC] processing PA-TNC attribute type 'ITA-HSR' 0x00902a/0x00000001
14[TNC] => 6 bytes @ 0x8080568
14[TNC] 0: 72 65 70 65 61 74 repeat
```
Future Work

- **Implementation of PTS protocol binding to IF-M**
  - HSR student Sansar Choinyambuu, implementor of the strongSwan IF-TNCCS 2.0 interface who is now working on TPM-based remote attestation is going to tackle the Platform Trust Service (PTS) protocol binding to IF-M as part of her Master Thesis.

- **Ultimate Goal: Full support of PTS attestation**
  - Stable strongSwan release with PTS attestation support expected in Q1 2012.
Thank you for your attention!

Questions?

www.strongswan.org/tnc/