CORE SECURITY

Intercepting SNC-protected traffic

Martin Gallo

Penetration Testing SME

March 2017

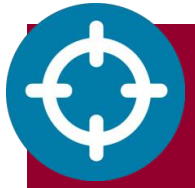
Agenda

Introduction

Problem

SAP Protocols

SAP SNC (Secure Network Connections)



Attack vectors

Vulnerable scenarios

Demo / Tools

Defense

Conclusions

Introduction

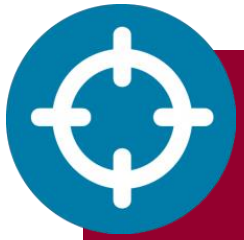
Problem

SAP Protocols

SAP SNC (Secure Network Connections)

Problem: Secure network connections

I need to secure my SAP network connections!



I need to test if my SAP network connections are secure!

Problem: Secure network connections

I need to secure my SAP network connections!

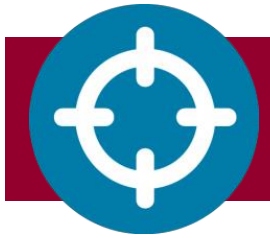
- Critical business processes
- SAP protocols don’t offer strong security by default
- Strong security
  - Authentication
  - Integrity
  - Encryption
- Move passwords out of the game (MFA, SSO, ..)
- Regulations, compliance, etc.
- Encrypt all the things!

Solution: Secure network connections

I need to secure my SAP network connections!

- HTTP, HANA SQL, OData, ..
  - SSL/TLS: Standard, (almost) well understood, supported, ..
- GUI/Diag, RFC, Router, ..
  - SNC (Secure Network Connections)
  - Single Sign-on

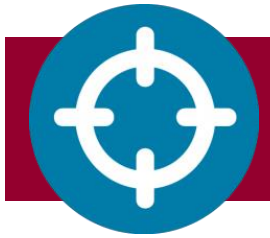
Problem: Secure network connections



I need to test if my SAP network connections are secure!

- Part of the critical attack surface
  - External exposure
  - Security before application-level authentication
- Increased deployment rates
  - Crypto library offered for free
  - Shipping of crypto library by default
- Seen BASIS admins do terrible things

Solution: Secure network connections

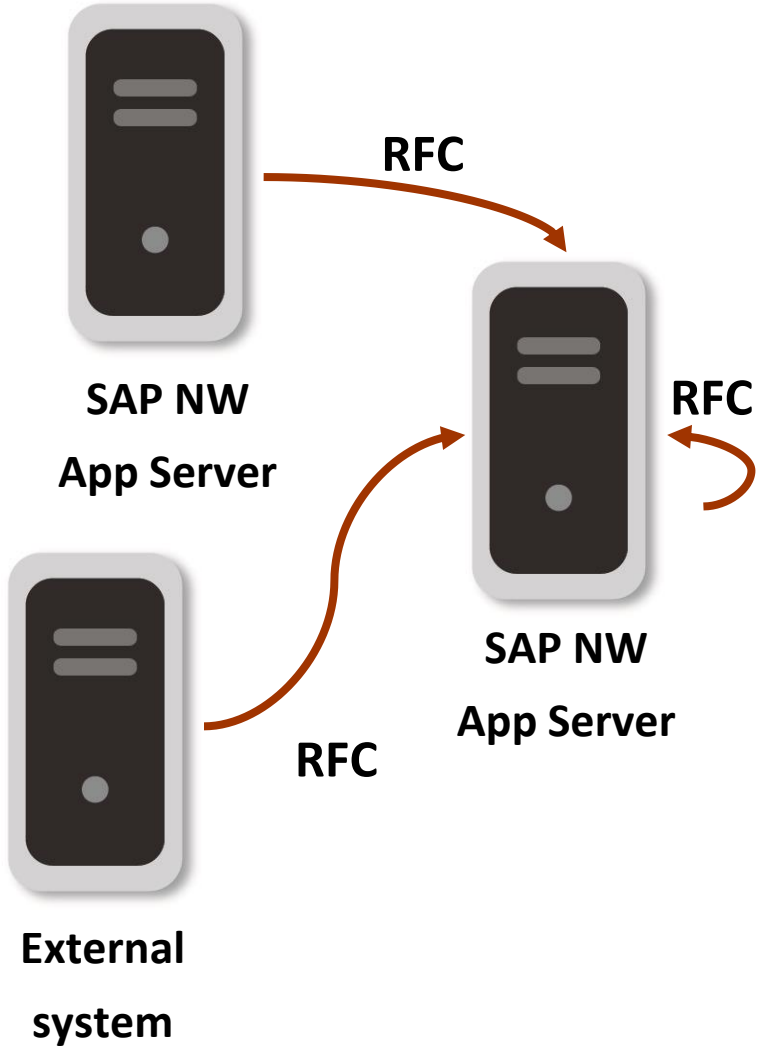


I need to test if my SAP network connections are secure!

- SSL/TLS
  - Tons of tools, guidance and literature, ..
- SNC (Secure Network Connections)
  - No tools available
  - Undocumented protocol
  - ???

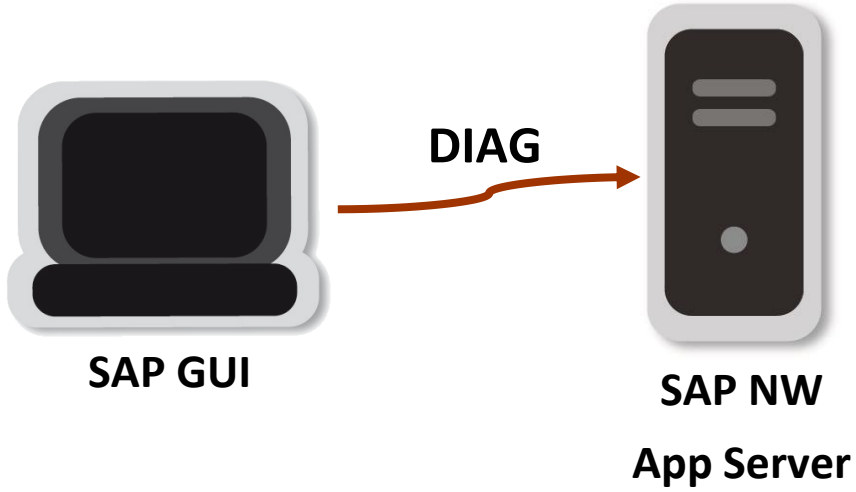
SAP protocols: RFC

- Communications on the application layer
  - Netweaver Application Server
    - Gateway service
  - External/third-party services
  - Other SAP servers
- Interfaces and actions
  - Execution of remote functions
  - Trigger business logic
- Authentication
  - Connection based
  - Username/password based
- Unencrypted by default



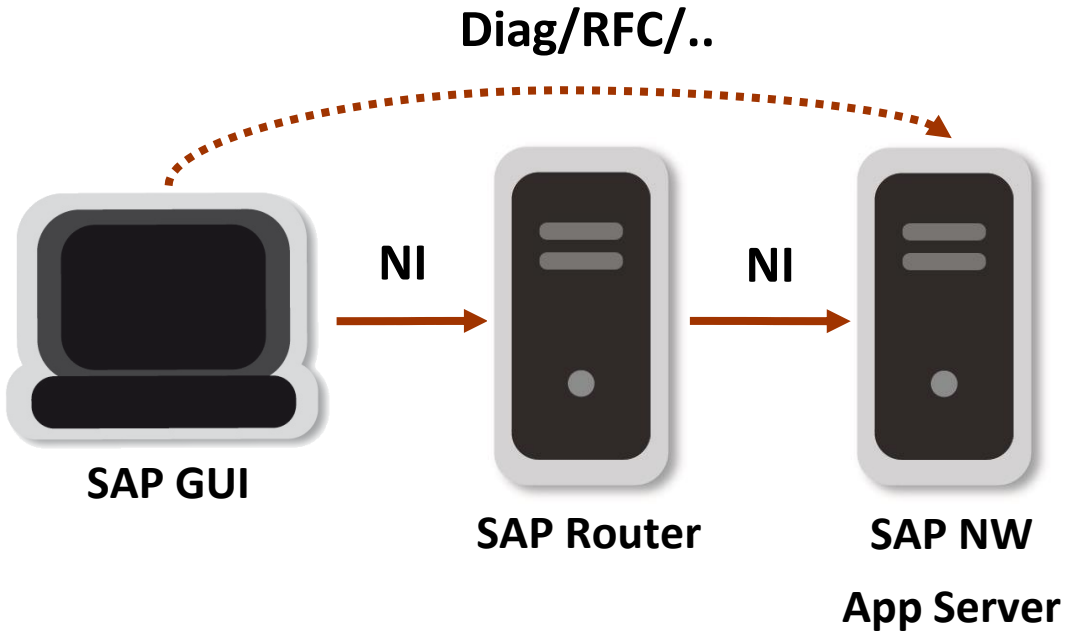
SAP protocols: Diag/GUI

- Communications between presentation and application layer
  - GUI <-> Netweaver Application Server
    - Dispatcher service
- UI components and actions
  - Screens, items, actions, interaction
  - RFC embedded calls
- Authentication
  - Connection based
  - Application-level login
  - Username/password based
- Compressed by default
  - LZC/LZH algorithm
- Unencrypted by default



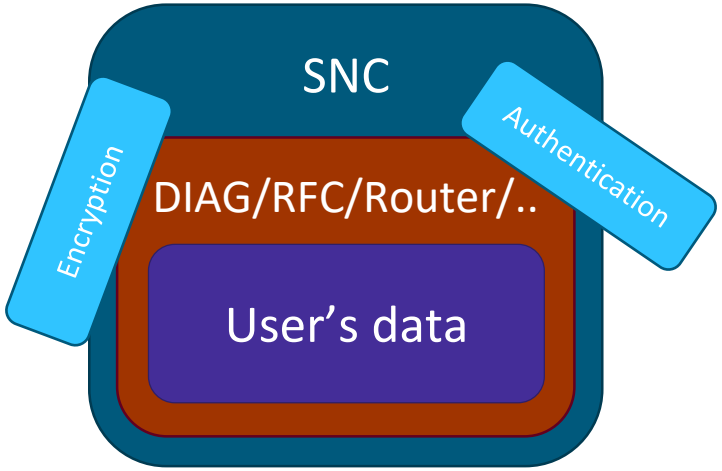
SAP protocols: Router

- Application level proxy
  - GUI
  - Netweaver Application Server
  - External/third-party services
  - Other SAP servers
- Proxy communications to other services
- Unauthenticated by default
  - Supports route passwords
- Unencrypted by default



SNC: Architecture

- Security layer
- Provides security properties
  - Strong authentication
  - Integrity protection
  - Data encryption
- Wraps supported protocols
  - DIAG
  - RFC
  - CPIC
  - SAPlpd
  - SAP Router
  - ..



SNC: Architecture

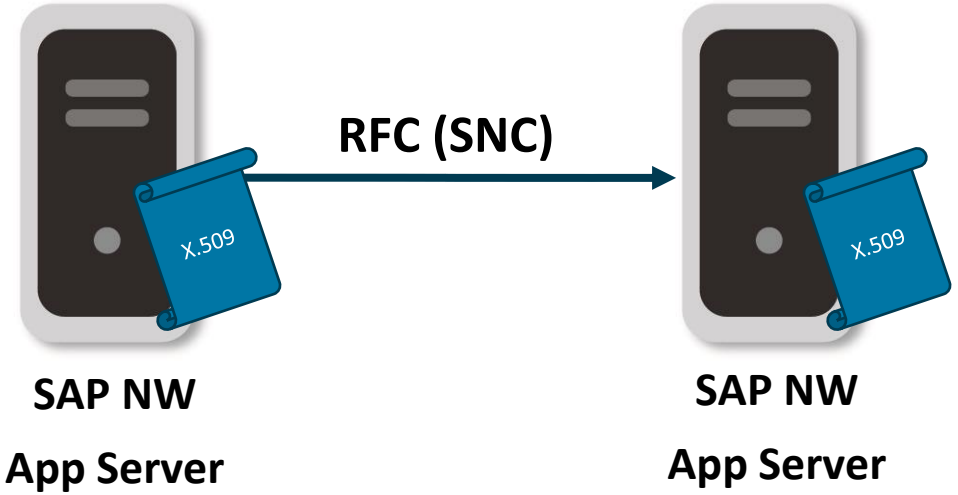
- Implements GSS-API
  - Generic Security Service Application Program Interface version 2
  - https://www.rfc-editor.org/rfc/rfc2743.txt
- Cryptographic primitives implemented via external libraries
  - GSS-API version 2
- SNC Adapters
  - SECUDE 5 gssapi
  - Kerberos 5 GSS-API
  - SAPNTLM gssapi
- Build custom adapters for implementing custom crypto

SNC: Architecture

- CommonCryptoLib
  - Standard SAP's crypto library
  - Replaces SAPCRYPTOLIB, SAPSECULIB
  - Implementation using GSS-API version 2
- Features
  - SNC for server-to-server and client-to-server
  - X.509 certificates and Kerberos
  - FIPS 140-2 compatible
  - Intel AES-NI feature set
  - ..
- Central Note for CommonCryptoLib 8 (replacing SAPCRYPTOLIB) - 1848999

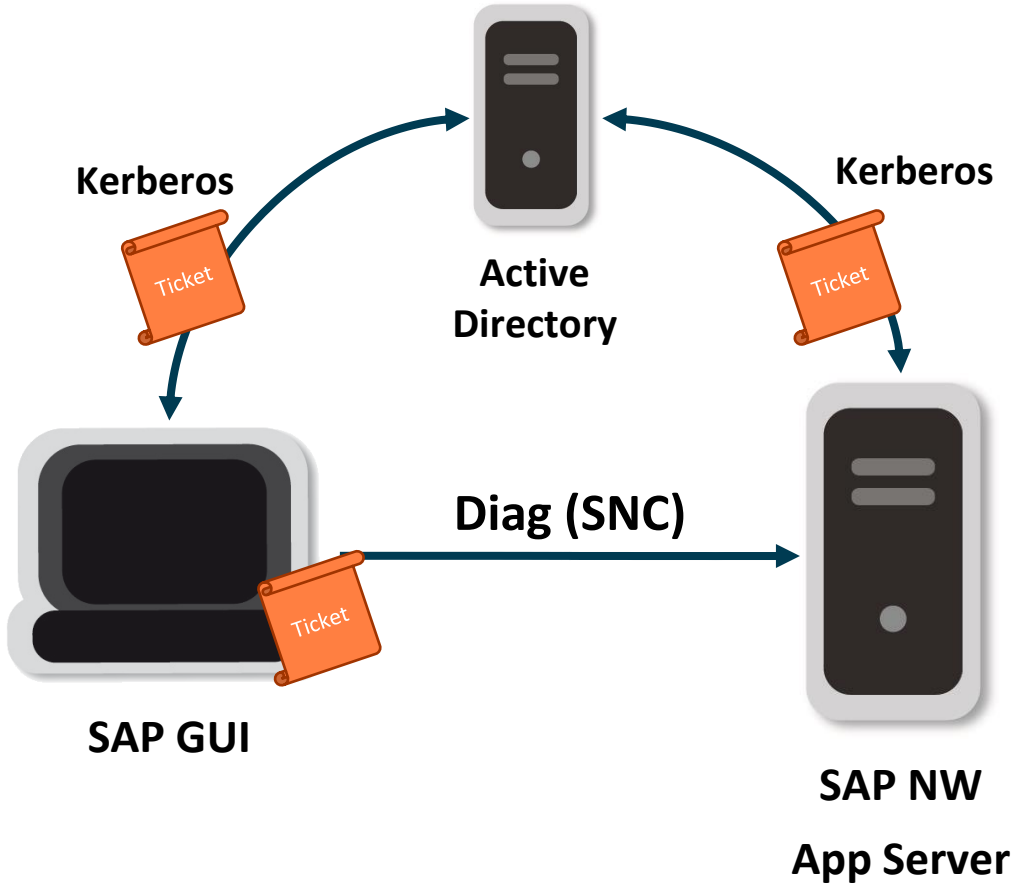
SNC: Implementation scenarios

- Server-to-server SNC with X.509 certificates
  - Each app server has its X.509 certificate
  - Mutual authentication



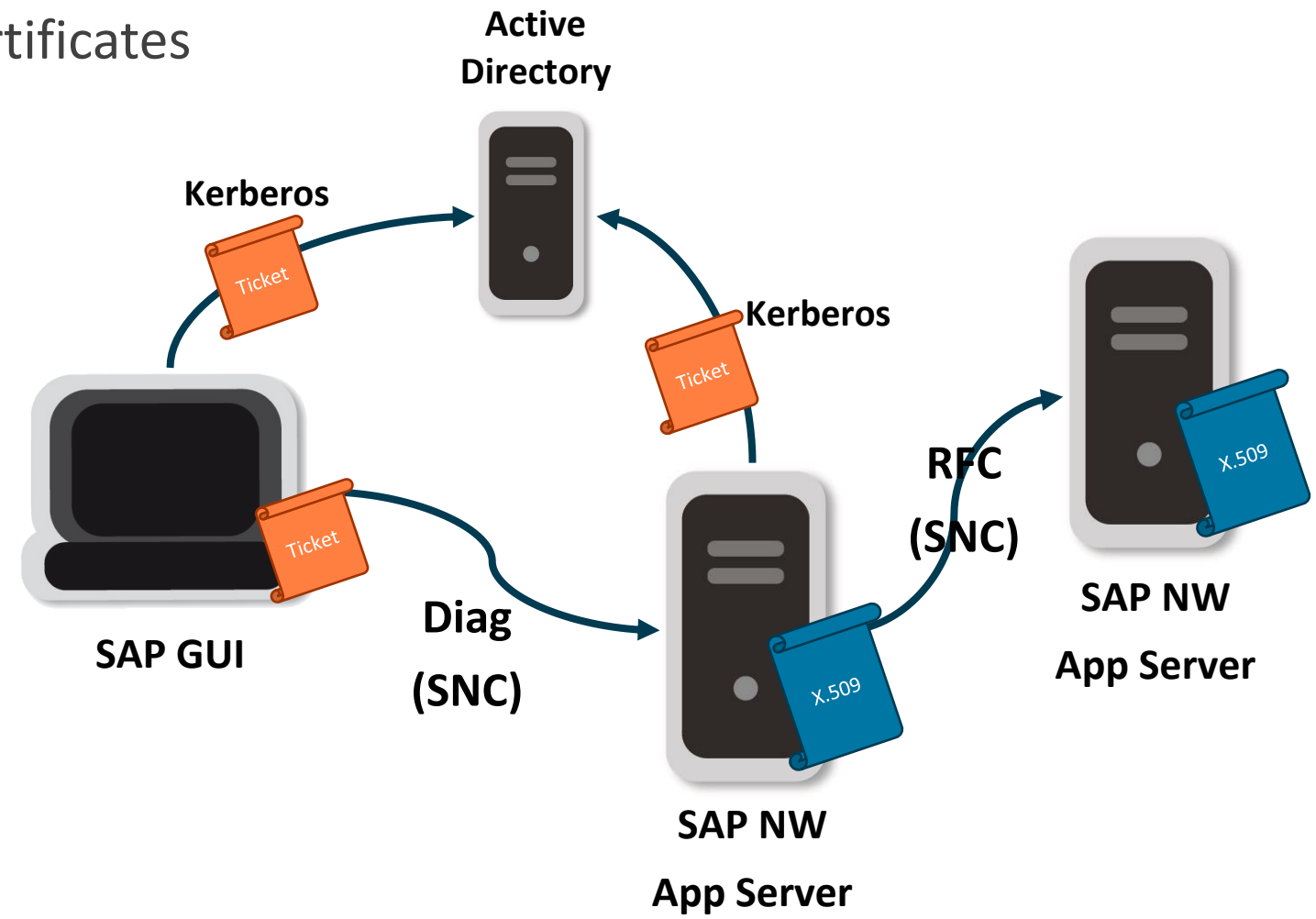
SNC: Implementation scenarios

- Client-to-server SNC with Kerberos
  - User mapping between SAP-AD
  - Single-sign-on



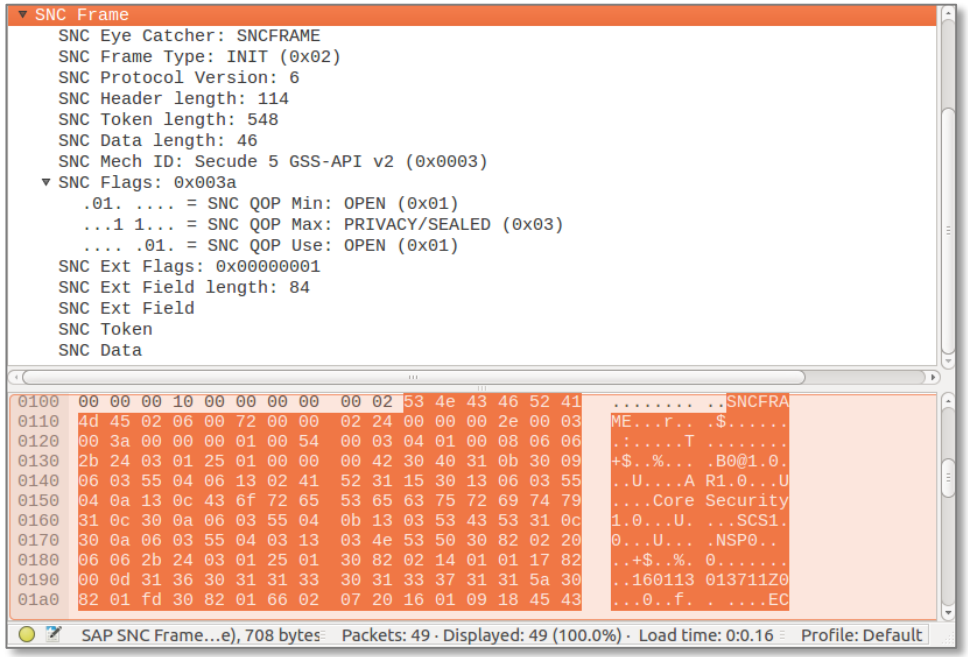
SNC: Implementation scenarios

- Hybrid implementation
  - Server-to-server using X.509 certificates
  - Client-to-server using Kerberos



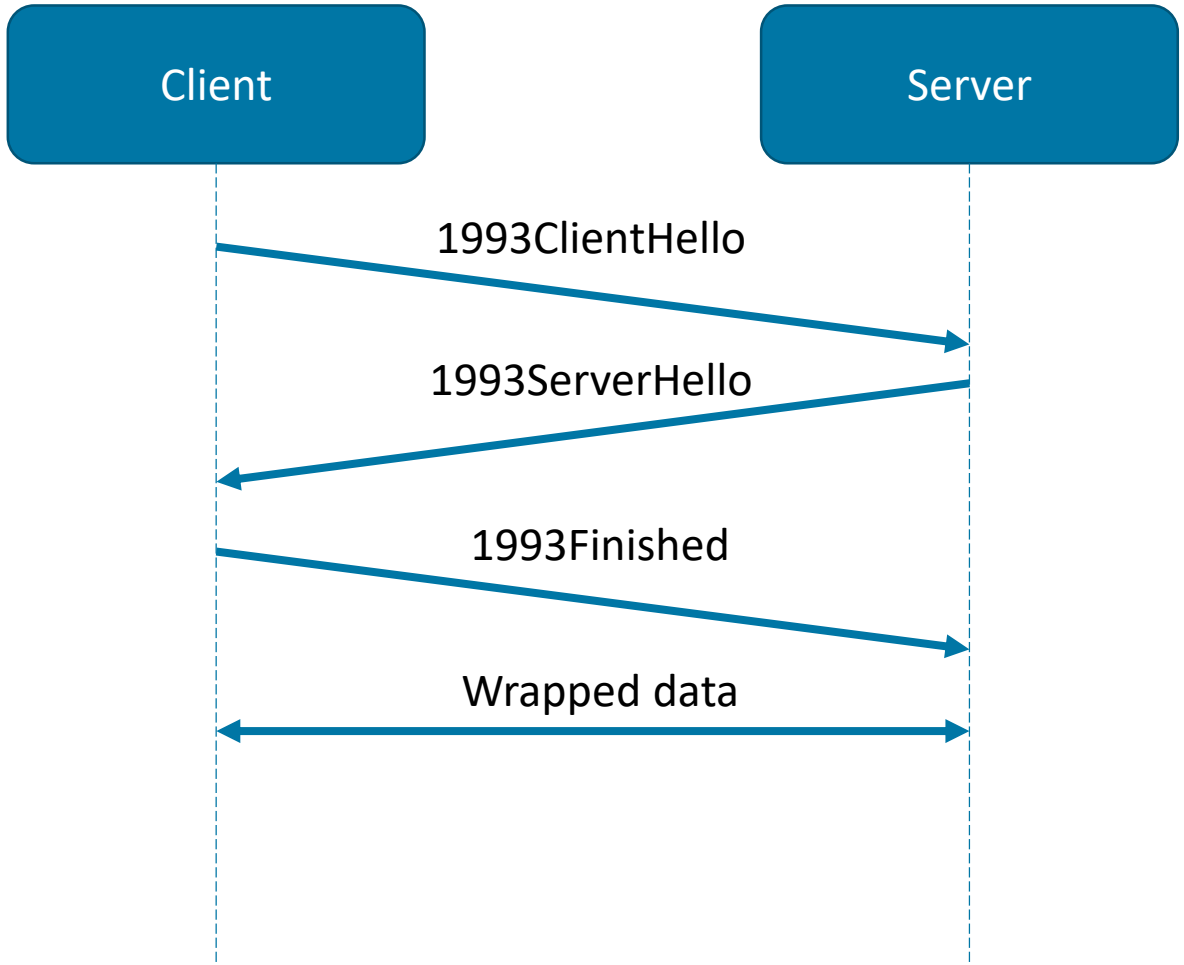
SNC: Protocol overview

- Versions
  - 1993 SAPCRYPTOLIB, CommonCryptoLib
  - 2010_1_0 CommonCryptoLib
  - 2010_1_1 CommonCryptoLib >= 8.5
- Messages structure
  - SNC Frame
    - Main header
      - Frame Type, protocol version, lengths, QoP, etc.
    - Extensions fields
      - Variable, according to the type of message
    - Token
      - Variable, according to the type of message
    - Data



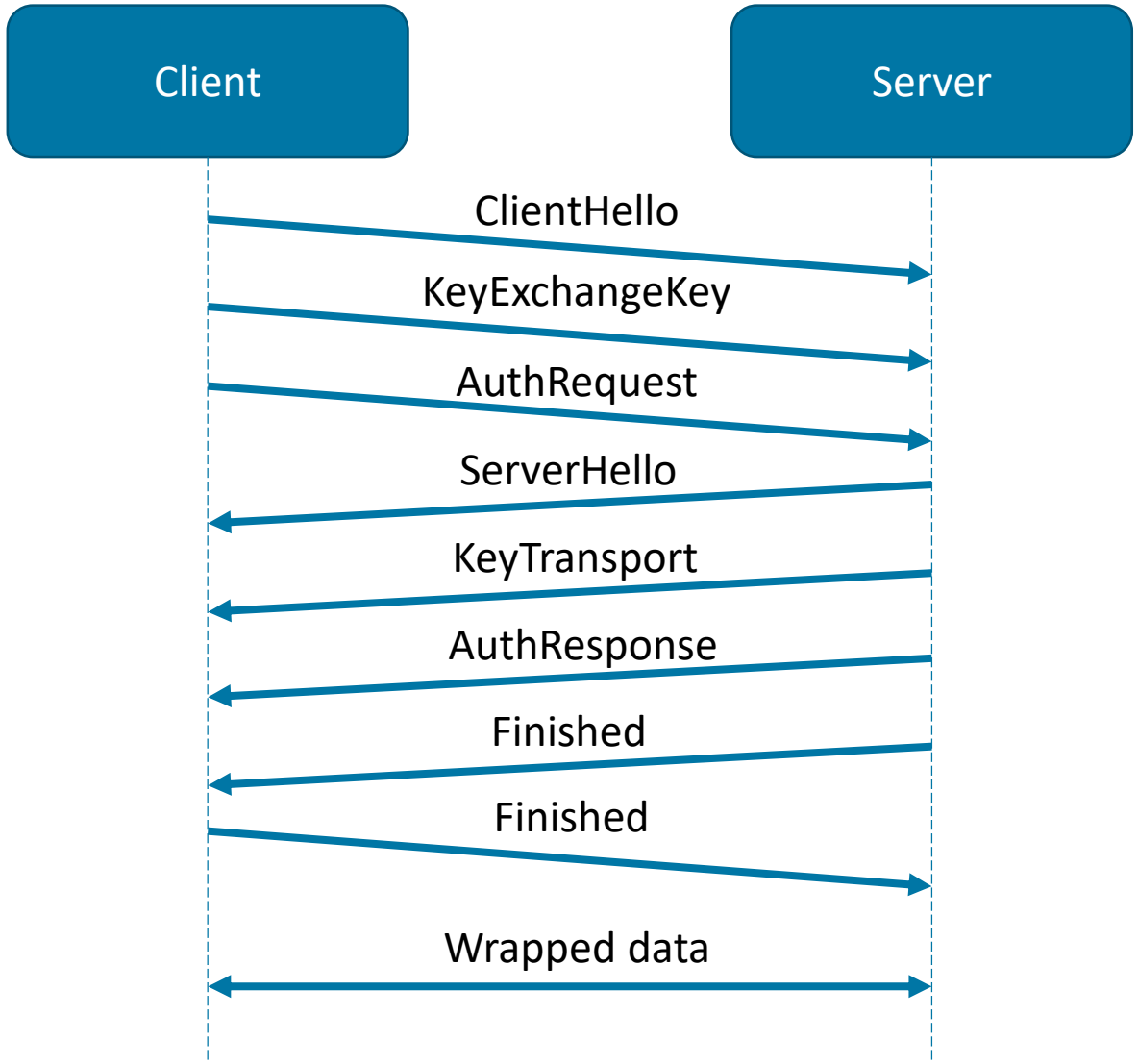
SNC: Protocol overview

- SNC Frame types
  - REVERSE_REQ, INIT_REQ, **INIT**, **INIT_ACK**, ACCEPT, ACCEPT_ACK, ACCEPT_FAILED, **DATA_OPEN**, **DATA_MIC/DATA_SIGNED**, **DATA_WRAP/DATA_SEALED**, SHUTDOWN, SHUTDOWN_MSG, REJECTED, ERROR, UNKNOWN
- Handshake protocol version 1993
  - Messages
    - 1993ClientHello
    - 1993ServerHello
    - 1993Finished
    - Wrapped data
  - Key exchange
    - RSA (default)
    - Kerberos



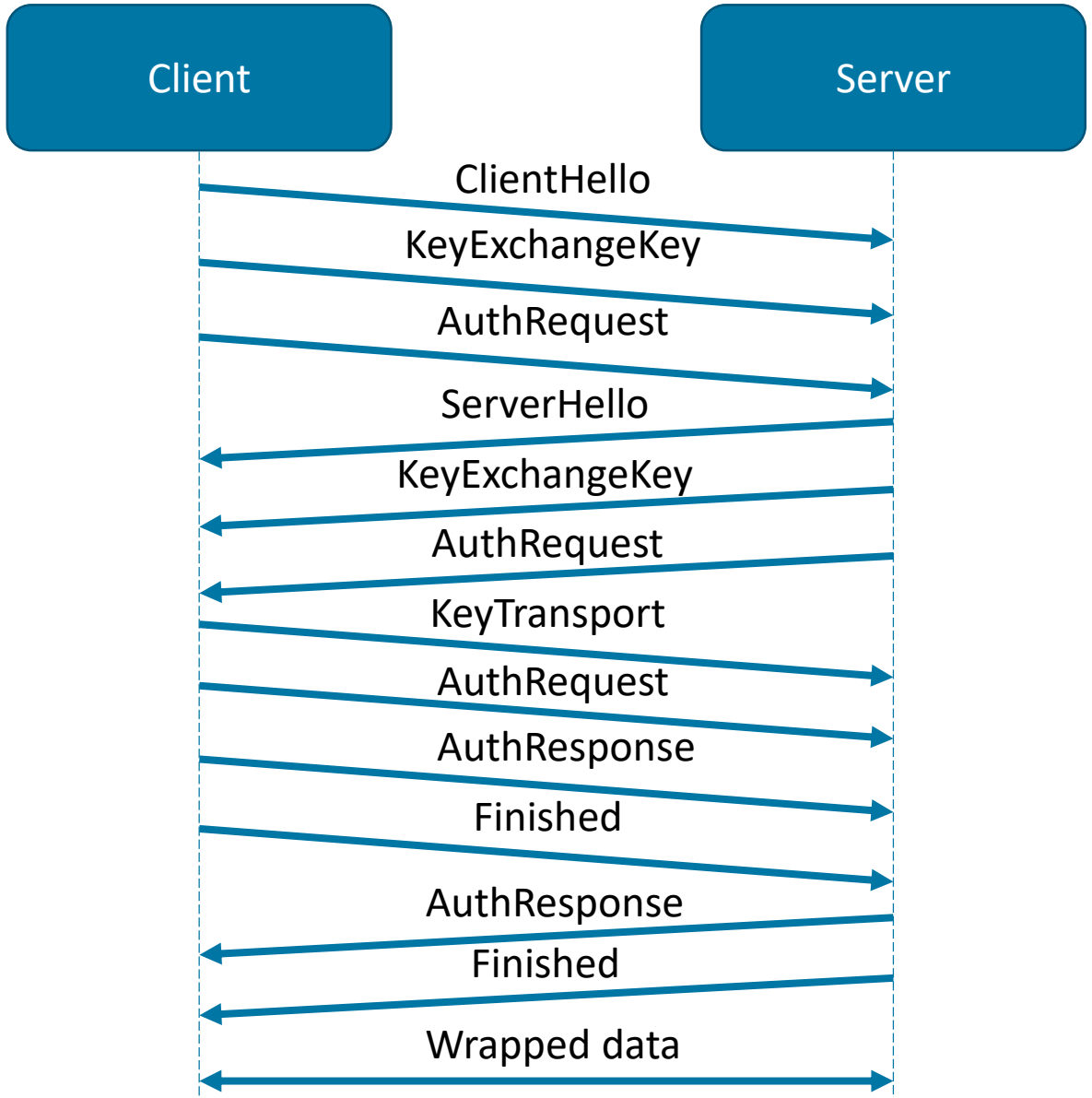
SNC: Protocol overview

- Handshake protocol version 2010_1_0
  - Messages
    - ClientHello
    - KeyExchangeKey
    - AuthRequest
    - ServerHello
    - KeyTransport
    - AuthResponse
    - Finished
    - Wrapped data
  - Key exchange
    - RSA (default)
    - Kerberos



SNC: Protocol overview

- Handshake protocol version 2010_1_1
  - Messages
    - ClientHello
    - KeyExchangeKey
    - AuthRequest
    - ServerHello
    - KeyTransport
    - AuthResponse
    - Finished
    - Wrapped data
  - Key exchange
    - RSA
    - Kerberos
    - ECDHE (default)



SNC: Protocol overview

Cipher suite	Strength	Protocol 1993	Protocol 2010_1_0	Protocol 2010_1_1
SNC_CL_RSA_AES128_RIPEMD160	MEDIUM	X		
SNC_CL_RSA_AES128_SHA1	MEDIUM	X		
SNC_CL_RSA_DES3_RIPEMD160	MEDIUM	X		
SNC_CL_RSA_DES3_SHA1	MEDIUM	X		
SNC_CL_RSA_AES256_SHA256	HIGH	X	X	X
SNC_CL_RSA_AES128_SHA256	HIGH		X	X
SNC_KERBEROS_AES256_SHA256	HIGH		X	X
SNC_KERBEROS_AES128_SHA256	HIGH		X	X
SNC_SR_RSA_AES256_SHA256	HIGH			X
SNC_SR_RSA_AES128_SHA256	HIGH			X
SNC_ECDHE_P256_AES256_SHA256	HIGH			X
SNC_ECDHE_P384_AES256_SHA512	HIGH			X
SNC_ECDHE_P521_AES256_SHA512	HIGH			X

SNC: Quality of Protection

- GSS-API Quality of protection
  - GSS-API RFC introduces QOP concept
  - Let implementations decide protection levels
  - Parties should be able to negotiate

- SNC QoP levels
  - Authentication only (level 1)
  - Integrity Protection (level 2)
  - Privacy Protection (level 3)

```
1.2.4:  Quality of Protection

   Some mech_types provide their users with fine granularity control
   over the means used to provide per-message protection, allowing
   callers to trade off security processing overhead dynamically against
   the protection requirements of particular messages. A per-message
   quality-of-protection parameter (analogous to quality-of-service, or
   QOS) selects among different QOP options supported by that mechanism.
   On context establishment for a multi-QOP mech_type, context-level
   data provides the prerequisite data for a range of protection
   qualities.
```

SNC: Quality of Protection

- Application Server specifies QOP on different parameters
  - `snc/data_protection/min`
    - Minimum requirement on protection level
    - Values from 1 to 3
  - `snc/data_protection/max`
    - Maximum protection level for connections **initiated** by the SAP system
    - Values from 1 to 3
  - `snc/data_protection/use`
    - Recommended level of protection
    - Values from 1 to 9 (>= *min*, <= *max*)
- In order to negotiate, other values were implemented
  - Default protection (level 8): “Use the value from *`snc/data_protection/use`*.”
  - Maximum protection (level 9): “Use the value from *`snc/data_protection/max`*.”

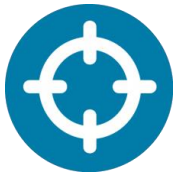
Attack vectors

Vulnerable scenarios

Demo

Tools

Vulnerable scenario: Eavesdropping / sniffing



- Passive attacker
- Able to sniff on the traffic
  - Wireless network
  - Controls one hop in the network
- Attacker only needs to unwrap traffic

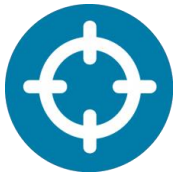
- Setup
  - SAP AS ABAP Server configured in default SNC mode
  - SAP GUI client using *QOP = 2* (integrity only)
- Demo time!

Vulnerable scenario: Interception and tampering

- Active attacker
- Able to perform a man-in-the-middle attack
  - Spoofing
    - ARP, DNS
  - Wireless network
  - Controls one hop in the network
- Attacker needs to unwrap, tamper, rewrap data
  - Unwrap data, tamper application level protocol, update modified lengths and then wrap data again

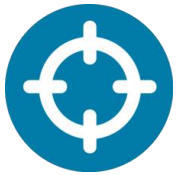
- Setup
  - SAP AS ABAP Server configured in default SNC mode
  - SAP GUI client using *QOP = 1* (authentication only)

Attack vectors: SNC/CommonCryptoLib



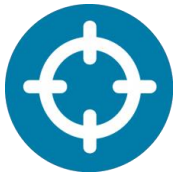
- Big attack surface
  - Memory leaks
    - SPNego in AS ABAP / security note 2253695
  - Timing attacks
  - Padding oracles
  - Memory corruption
    - lack of RELRO / security note 2427966
  - Downgrade/disabling attacks
  - Certificate/signature validation
    - XML security / security note 2434136

Attack vectors: SNC/CommonCryptoLib



- Big attack surface
  - Off-line password/key attacks
    - usage of SHA-1 as default in X.509 signature / security note 2275390
  - XML, ASN.1 and general parsing
    - certificate parsing / security note 2376742
  - Cryptographic flaws
  - NTLM/Kerberos attacks
  - ..

Tools



- pysap
  - Python library to craft/parse network packets
  - Added SNC Support
  - Open Source (GPL)
    - https://github.com/CoreSecurity/pysap
- SAP Wireshark plug-in
  - Plug-in for Wireshark that dissects SAP's protocols
  - Added SNC Support
    - Diag and Router protocols
  - Open Source (GPL)
    - https://github.com/CoreSecurity/SAP-Dissection-plug-in-for-Wireshark
- New releases soon!

Defense Conclusions

Defense

- Patch, patch, patch
  - Kernel & CommonCryptoLib up-to-date
  - Reduce attack window and exposure
- Review, test
  - Architecture review
  - Penetration testing
- Prepare for the worst scenario
  - Patch procedures
  - Forensic capabilities

Defense

- Use CommonCryptoLib > 8.5
  - ECDHE, PFS when working with Single Sign-on 3.0
- Secure *ALL* paths
  - Client-to-server
  - Server-to-server
  - Internal connections
  - When protocol not supported, use SAP Router with SNC routes
- Disallow unencrypted connections
  - SAP Note 1690662 - Option: Blocking unencrypted SAPGUI/RFC connections

Defense

- Set all QOP levels to maximum level
  - Parameters `snc/data_protection/(min/max/use)`
    - Value `3` = privacy protection

- Use the latest protocol available in the environment
  - Parameters `ccl/snc/(client/server)_protocol`
    - ~~`1993` (supported by SAPCRYPTOLIB and CommonCryptoLib)~~
    - `2010_1_0` (supported by CommonCryptoLib)
    - `2010_1_1` (supported by CommonCryptoLib 8.5 and newer)

Configuration parameters: SAP Note 2338952 - CommonCryptoLib 8.5: Configuration Profile Parameters

Defense

- Set good cipher suites
  - Parameters `ccl/snc/(client/server)_cipher_suites`
    - Client: value `HIGH`
    - Server: value `HIGH` or select particular cipher suites

- Enable certificate revocation
  - Parameters `ccl/pkik/*` and `ccl/(snc/ssf/ssl)/pkix_revocation_check`
  - So far only CRL supported
    - Planned Support for OSCP in CCL/Single Sign-On

Configuration parameters: SAP Note 2338952 - CommonCryptoLib 8.5: Configuration Profile Parameters

Conclusions

- Crypto is hard
  - Good crypto is harder
- Attacks against SNC are practical
  - They’re already among us!
- Bad implementation not necessarily better than nothing
  - Lax controls (because it’s protected!)
  - False sense of security
  - Lack of accountability
- Opportunistic encryption might not be enough in most cases
  - Assess risk

Thank you!

thanks to the Troopers crew!
and thanks Joris, Euge!

Contact:

mgallo@coresecurity.com

@martingalloar

TROOPERS

CORE SECURITY