My Adventures with TCP/IP Port Security and RACF on z/OS

Joel Tilton
RACF Engineer
Mainframe Evangelist

https://www.linkedin.com/in/joeltilton
RACFEngineer@gmail.com
702-483-RACF (Google Voice)
Long Live the Mainframe
Disclaimers

- All products, trademarks, and information mentioned are the property of the respective vendors.
- Mention of a product does not imply a recommendation.
- Always test new profiles on a non-production system.
- Only you can prevent IPLs!
- The views expressed are his own personal views, and are not endorsed or supported by, and do not necessarily express or reflect, the views, positions or strategies of his employer.
Agenda

- So what is a port anyway?
- Why Port Security with RACF
- EZB.PORTACCESS Profile Syntax
- SAFNAME Design
- Port Reservation Syntax
- Planning & Implementation Strategy
- SERVAUTH Class Activation
- Unreserved Ports – TCP & UDP
- Auditing Port Access
- Required PTFs
- Additional Resources
- Summary
What is a Port?

• An IP address is used to route the message to your computer. Once it arrives there, TCP uses the port number to know which program like ftp or email to hand it to.

• From a SERVAUTH perspective…
  – Any mainframe program binding to
  – and/or listening on a TCP or UDP Port
  – SYS1.TCPIP.PROFILE

• For VTAM binds see VTAMAPPL
  – Stu Henderson & Peter Hager
Why Port Security with RACF?

**Native TCPIP**
- Reservation by Jobname
- Can be spoofed
  - Unless JESJOBS profiles protecting jobnames
- Violations not well logged
- Unreserved ports not easily controlled
- Low Ports *possibly* protected with RESTRICTLOWPORTS
  - PORT JOBNAME reservation takes precedence
  - Did I mention JESJOBS?!

**RACF**
- Reservation by SAFNAME
- Cannot be spoofed
  - RACF profile **FINAL** answer
- Successes or Violations logged to SMF (type 80)
- Unreserved ports easily controlled
- Low Ports **ALWAYS** protected with RESTRICTLOWPORTS
  - EZB.PORTACCESS profiles take precedence
## EZB.PORTACCESS Profile Syntax

### EZB.PORTACCESS.sysname.tcpname.safname

<table>
<thead>
<tr>
<th>Qualifier</th>
<th>Description</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysname</td>
<td>Local SMF ID</td>
<td>• Use * unless need for per system segregation</td>
</tr>
<tr>
<td>tcpname</td>
<td>TCPIP started task jobname</td>
<td>• Use * unless multiple stacks</td>
</tr>
<tr>
<td>safname</td>
<td>Esoteric name coded in port reservation</td>
<td>• Can be generic</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1 – 8 characters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• First Position Not Numeric without PTF UI27609 z/OS V2R1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• First Position Never 0 (zero)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• RFE 75935</td>
</tr>
</tbody>
</table>

### Netaccess Statement SAF Names

APAR PI36695  PTF UI27609 for z/OS V2R1


Applies to NETACCESS, PORT, PORTRANGE, VIPADYNAMIC & VIPARANGE

Complete your session evaluations online at [www.SHARE.org/Orlando-Eval](http://www.SHARE.org/Orlando-Eval)

© 2015, Joel M. Tilton – PORTACCESS @ SHARE Orlando 2015
SAFNAME Design

- Use known protocol name as SAFNAME
  - HTTP, HTTPS, LDAP, LDAPS
  - … if appropriate
- Use generics in profile, as appropriate
  - HTTP*, LDAP*
  - … if appropriate
- Relationship
  - 1 or more port reservations to RACF profile
  - Use Wisely
- Enhance SERVAUTH SAFNAME Netaccess
  - Allow 0 in first position
Port Reservation Syntax – Single

• An example of reserving an individual port with SAF

PORT
;Port Protocol Jobname SAF SAFName
80 TCP * SAF HTTP ;webserver
389 TCP * SAF LDAP

– With SAFNAME, jobname only needed to distinguish between two different port listeners

636 TCP LDAPDIR BIND 192.168.0.8 SAF LDAPD ; LDAPDIR
636 TCP LDAPPKI BIND 192.168.0.9 SAF LDAPPKI ; LDAPPKI
Port Reservation Syntax – Range

• Reserve Port Ranges

PORTRANGE
;Portrange  Length  Protocol  Jobname  SAF  SAFName
1000  51  UDP  *  SAF  OMEGAMON

  – Reserves 1000 through 1050 for Omegamon

• Use Only One
  – Reserve Individual Port
  – Reserve Port Range
Single Port Overrides Range

- Reserve Same Port Collisions
  - Different SAFNAME
  - Only Single Port Reservation Used

- ICH408I
  - Call Security!
  - Update SYS1.TCPIP.PROFILE
  - OBEY Command or IPL
Example: TN3270 – RACF Profile

- **EZB.PORTACCESS.*.*.TN3270**
  - UACC always NONE
  - Permit TN3270 STC user ID with READ
  - AUDIT ALL(READ)
    - Audit all port access attempts; failures and successes
    - Exclude FTP data port from SAF

- **WARNING**
  - Use wisely as an implementation strategy
  - Anything can bind to or listen
Example: TN3270 – Reservation

PORT 23 TCP TN3270

• Non-SAF uses jobname
• Without JESJOBS, submitting a jobname of TN3270 would allow any program to bind to port 23

PORT 23 TCP * SAF TN3270

• With SAF
  – Jobname unnecessary
  – Only use jobname where needed
Port 20 – Why Use SAF?

• All users of unencrypted FTP need access
• z/OS FTP Client may bind & listen on 20 ➔ active FTP connections
• Ideally unencrypted FTP phased out
  – Especially for PCI compliance

• Classic Jobname & RESTRICTLOWPORTS ➔ Less Secure

20 TCP OMVS NOAUTOLOG
  – JOBNAME on PORT statement ➔ RESTRICTLOWPORTS overridden
  – Always bind to port 20 if jobname matches OMVS
  – OMVS jobname associated with any job that uses z/OS UNIX services
  
  – JESJOBS SUBMIT.nodename.jobname.userid
    • Control whom can start a batch job
  – OPERCMDS MVS.START.STC.mbrname.jobname
    • Control whom can start a given job name on the START command
  – FACILITY class BPX.JOBNAME
    • Control which users can use _BPX_JOBNAME environment variable
Port 20 – Making the World A Safer Place

• SERVAUTH does not support RACF-DELEGATED…yet 😊
  – Request For Enhancement (RFE)
    – Please VOTE! Your Vote Matters!

• Redbook Update Coming Soon
  – IBM z/OS V1R13 Communications Server TCP/IP Implementation: Volume 4 Security and Policy-Based Networking
RESTRICTLOWPORTS & UID(0)

Port Authority
• UID(0)
• BPX.SUPERUSER
• SAF

RESTRICTLOWPORTS
• Access Granted
• Access DENIED
• Access Granted
  – Even For Low Ports
  • < 1024
Parse NETSTAT commands

- Write a REXX parsing NETSTAT CONN & PORTLIST output

Use WARNING

- Use WARNING mode
- Consume Type 80s

Con:
- Anything can bind any program to ports

Pro:
- Captures all port access over time

Con:
- Will not see all port usage
- Snapshot in time

Pro:
- No port exposure assuming JESJOBS active
Planning – Gather Information

• Evaluate running STCs and their ports
  – NETSTAT CONN ➔ What is Listening
  – NETSTAT PORTLIST ➔ How it is Reserved
    • REMINDER: SERVAUTH EZB.NETSTAT.**
  – REXX EXEC compare reservations vs. usage
• Create Spreadsheet of Port Listeners & SAFnames
• Partner with Network/VTAM Engineer
  – TCPIP profile changes
  – Weekend IPLs
• Update Software Parms
• Implement one system at a time
  – development, test and then production
• REMEMBER: Only you can prevent IPLs!
Planning – Implementation

• Build EZB.PORTACCESS profiles
  – No effect until SYS1.TCPIP.PROFILE Updated
  – TCP Ports – OMPROUTE READ

• Update port reservations to call SAF

• Activate via IPL or TCPIP OBEY
  – Large number of STCs ➔ IPL
  – OBEY command is dynamic
    • Cycle Started Tasks
    • Excludes FTP
Planning – Intermittent Listeners

• NETSTAT CONN
  – Shows ports in use now
  – Not every port is in constant use by its listener

• Find intermittent port listeners
  – WARNING AUDIT (ALL (READ))
    • EZB.PORTACCESS.*.*.UNRSVTCP
  – Mine SMF records
  – Midnight Logons – Optional, Bring a Friend!

• Update TCPIP profile Port Definitions
  – RDEFINE SERVAUTH EZB.PORTACCESS.*.*.SAFName
  – PERMIT EZB.PORTACCESS.*.*.SAFName class (SERVAUTH)
    access (READ) ID (STC UserID)

• IPL

• Rinse, Recycle, Repeat ...
SERVAUTH Class Activation

• Activate SERVAUTH Class
  – IBM Class Descriptor Table (CDT)
  – `SETR classact(SERVAUTH) audit(SERVAUTH) raclist(SERVAUTH) generic(SERVAUTH)`
  • RC of 4 class but be mindful of SYS1.TCPIP.PROFILE
    – SERVAUTH profiles for DVIPA
    – `EZD1313I -REQUIRED SAF SERVAUTH PROFILE NOT FOUND RACF profile name`

• `RDEFINE RACGLIST SERVAUTH OWNER()`
  – IPL will **not** refresh in-storage RACF profiles
  – Ensure Sysplex Consistency for RACF
  – By Product…Performance Improvement
  – `SETR classact(RACGLIST) audit(RACGLIST)`
  – `SETR RACLIST(...) REFRESH` Builds RACGLIST profiles
Required PTFs – Apply FIRST

• Spurious SAF (RACF) Violations from use of UDP Sockets
  – APAR PI18153  PTF UI8700 for z/OS 1.13
  – APAR PI18151  PTF UI9430 for z/OS 2.1
  – Use of UDP Ephemeral ports causes random security violations

• ABEND S0C4 IN EZBXFUT6
  – APAR PI07541  PTF UI13629 for z/OS 1.13
  – APAR PI08351  PTF UI14006 for z/OS 2.1
  – PORT UNRSV TCP * SAF SAF*name
  – Mapping via SRCIP to a DVIPA with sysplexexports defined
  – Does not have permission to the SAF resource
Unreserved Ports TCPIP PORT Syntax

• **PORT UNRSV TCP * SAF UNRSVTCP**
  – Prevent TCP port listeners ➔ TCP default

• **PORT UNRSV UDP * SAF UNRSVUDP**
  – Prevent UDP port listeners & binds ➔ UDP default

• Stop Unauthorized Port Use
  – Goal: Empty ACLs
  – **AUDIT (ALL (READ)) UACC (NONE)**

• Consideration: Dynamic Ephemeral UDP ports
  – See Required PTFs
Unreserved Ports RACF Profile Syntax

• Build SERVAUTH Profiles
  – `EZB.PORTACCESS.*.*.UNRSVTCP OWNER(...)`  
    UACC(NONE)  WARNING  AUDIT(ALL(READ))
  • Permit all STCs in the Beginning
    – `EZB.PORTACCESS.*.*.UNRSVUDP OWNER(...)`  
    UACC(NONE)  WARNING  AUDIT(ALL(READ))

• Read SMF, Read SMF, Read SMF
  – Logstring Your New BFF
• Reserve Port in SYS1.TCPIP.PROFILE
• Adjust Software Parms, If Necessary
• Obey / IPL
• Cautiously Restrict Access
UDP Unreserved Ports – SOLVED

• Applications Need Dynamic Ephemeral UDP Ports

• STC wants to send E-Mail

• STC opens UDP Ephemeral port on demand
  – SMTP

• Triggers SAF call unless…
  – PTF UI8700 for z/OS 1.13
  – PTF UI9430 for z/OS 2.1
TCP Unreserved Ports – Omegamon

- Binds ports to loopback (127.0.0.1)
  - New PTF under development
  - New Parm KDE_LOOPBACK_POOL
  - Needs 1,000 ports, but wait there’s more!

- Multiples of 4096 + a base number
- $1918 + (n \times 4096) = \text{Agent Port Number}$
  - Where $N = \text{The startup agent number}$.
- 1918 – Base Port, always assigned to hub or remote TEMS (Omegcms)
- 1920 – IBM Tivoli Monitoring Service Console (assigned to first agent to start up, OMEG*)
- 6014 – MVS Agent
- 10110 – CICS Agent
- 14206 – Network Agent
TCP Unreserved Ports – Omegamon

- 6014 TCP * SAF OMEGAMON ; OMEGAMON
- 10110 TCP * SAF OMEGAMON ; OMEGAMON
- 14206 TCP * SAF OMEGAMON ; OMEGAMON
- 18302 TCP * SAF OMEGAMON ; OMEGAMON
- 22398 TCP * SAF OMEGAMON ; OMEGAMON
- 26494 TCP * SAF OMEGAMON ; OMEGAMON
- 30590 TCP * SAF OMEGAMON ; OMEGAMON
- 34686 TCP * SAF OMEGAMON ; OMEGAMON
- 38782 TCP * SAF OMEGAMON ; OMEGAMON
- 42878 TCP * SAF OMEGAMON ; OMEGAMON
- 46974 TCP * SAF OMEGAMON ; OMEGAMON
- 51070 TCP * SAF OMEGAMON ; OMEGAMON
- 55166 TCP * SAF OMEGAMON ; OMEGAMON
- 59262 TCP * SAF OMEGAMON ; OMEGAMON
- 63358 TCP * SAF OMEGAMON ; OMEGAMON
- PORTRANGE 1900 51 TCP * SAF OMEGAMON
- PORTRANGE 1900 51 UDP * SAF OMEGAMON
TCP Unreserved Ports – z/OS FTP Client PTF

• If Passive FTP fails, z/OS FTP Client attempts Active connection by default
  – Active connection Binds & Listens on…
  – Random TCP port! ICH408I!

ICH408I USER() GROUP() NAME()
EZB.PORTACCESS.SYSTEM.TCPIP.UNRSVTCP CL(SERVAUTH)
INSUFFICIENT ACCESS AUTHORITY
FROM EZB.PORTACCESS.*.*.UNRSVTCP (G)
ACCESS INTENT(READ ) ACCESS ALLOWED(NONE )

• APAR PI29994 PTF UI26945 for z/OS 1.13
• APAR PI36683 PTF UI27396 for z/OS 2.1

• New FTP.DATA Parm:
  – PassiveOnly TRUE
• Add Existing Parm too:
  – FWFRIENDLY  TRUE
TCP Unreserved Ports – WAS Port Scans

• PM96838
  – Available for WAS v7.0, v8.0 and v8.5
  – Optionally disable port activity checking when a server is created
  – `com.ibm.ws.management.suppressPortScan=true`
    • JVM argument is added to suppress port check
  – Note that when this is in effect, ports in use by other applications will not be detected and could lead to port conflicts.

• PI40568
  – Still under construction
  – Tentative GA of September 2015
TCP Client Bind Security – WHENBIND

• Purpose: Call SAF for all TCP client binds
  – Client needs to request an ephemeral port by binding to port 0
• Why?
  – Do you really trust a TCP client to be in control of what port it can use?

• PORT UNRSV TCP * SAF UNRSVTCP WHENBIND
  – WARNING RECOMMENDED
  – RDEFINE SERVAUTH EZB.PORTACCESS. SYSNAME.*.UNRSVTCP
    WARNING UACC(NONE) AUDIT(ALL(READ))
    • i.e. One system at a time
  – Read SMF…Parse LOGSTRING for Port
• Plan of Attack: One Software product at a time
TCP WHENBIND – Challenges

• SAS
  – E-mail engine uses ports
  – TCP_EPH_MAP_ENABLED=0 ➔ zero
  – TKMVSENV DD
    • hlq.TKMVSENV(TKMVSENV)

• VPS
  – Remove parm ➔ TCPHOSTS

• FTP with TLS
  – Must reserve data port
Implementation Strategy

• Apply required PTFs
• Activate SERVAUTH class (RACGLIST too!)
• Known TCP & UDP Ports – Phase 1
  – Profiles in WARNING as appropriate
• “Midnight Madness” Port Listeners – Phase 2
• Secure Unreserved UDP ports – Phase 3
• Secure Unreserved TCP ports – Phase 4
  – Goal: Empty ACLs for Unreserved ports
• Secure Unreserved TCP client binds – Phase 5
  – Because we shouldn’t trust TCP software to be in control of ports
Auditing Port Access

22Nov14 12:14:31.11 OMEGC ZOS1 RACF ACCESS success for OMEGC: (READ,READ) on SERVAUTH EZB.PORTACCESS.sysname.TCPIP.OMEGAMON

Jobname + id: OMEGCMS STC12345

Class : SERVAUTH Resource: EZB.PORTACCESS.ZOS1.TCPIP.OMEGAMON

Access used : READ Profile: EZB.PORTACCESS.*.*.OMEGAMON

Log string : TCPIP PORT ACCESS CHECK PORT 01000

- RACF Final Port Authority
- All Port Usage Logged Type 80

LOGSTRING contains port number
  - TCP / UDP Not Specified
  - RFE
To Infinity And Beyond – What’s Next?

• Would it be even better if SERVAUTH would validate the intended use of the port?
  – INSERT AUDIENCE AGREEMENT HERE 😊

• Today
  – If RACF grants access
  – then the Port is Yours

• Tomorrow
  – Validate Port Being Used As Intended
Additional Resources

• Techdocs Library – Using SERVAUTH to Protect TCP Port Usage
  – http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100673

• Techdocs – Undesired PortAccess Violations

• Port Access Control Chapter
  – z/OS Communications Server: IP Configuration Guide

• SERVAUTH Class profiles used by TCP/IP
  – EZB.PORTACCESS syntax
Even more Useful Resources

- **IBM z/OS V1R13 Communications Server TCP/IP Implementation: Volume 4 Security and Policy-Based Networking**

- **RESTRICTLOWPORTS parameter**

- **TCPIP PROFILE Port Assignments**
  - [http://www-01.ibm.com/support/knowledgecenter/SSLTBW_2.1.0/com.ibm.zos.v2r1.halz001/profiletcpipportassignments.htm](http://www-01.ibm.com/support/knowledgecenter/SSLTBW_2.1.0/com.ibm.zos.v2r1.halz001/profiletcpipportassignments.htm)
Some Statistics

![Graph showing statistics for Joel's Hairline, RFE, and PTFs]

- **Joel's Hairline**
  - Closed
  - Open
  - Linear (Open)

- **RFE**
  - Closed
  - Open

- **PTFs**
  - Closed
  - Open
Summary

• Try not. Do…or do not. There is no try!
  – Master Yoda
• How do you tackle any project? One small step at a time…
• Protecting Ports is of Paramount Importance
  – Securing with RACF
    • prevents spoofing
    • logs port usage (success & failures) to SMF
• Requires Proper Planning
• Close partnership with Network Engineer
• Coordinate TCPIP Profile & RACF Changes
• IPL during maintenance windows
• Fix ICH408Is and:
  – Recycle STC or possibly IPL
• Port Security Engaged!

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
© 2015, Joel M. Tilton – PORTACCESS @ SHARE Orlando 2015
My Thanks To…

- Stu Henderson
- Adam Klinger
- Ray Kohring
- Christopher Meyer
- Carolyn Miller
- Howie Odishoo
- Hayim Sokolsky
- Todd Valler
- William Vender
- Bruce Wells

- IBM Omegamon Level 2 & Level 3
- IBM z/OS Comm Server/TCPIP Level 2 & Level 3
- IBM zSecure Level 2 & Level 3
- IBM WAS Level 2 & Level 3

- And the Adventure Continues to Boldly Go Where No Port Has Gone Before …

- DISCLAIMER: No ports were harmed in the making of this presentation…perhaps shaken & stirred but they were not permanently damaged. 😊
Questions?