

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

Joel Tilton
RACF Engineer
Mainframe Evangelist
November 2014
NY & Tampa Bay RACF Users Group

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...
 - Or forest fires!

Agenda

- Why Port Security with RACF
- EZB.PORTACCESS Profile Syntax
- SAFNAME Design
- Port Reservation Syntax
- Planning
- SERVAUTH Class Activation
- Unreserved Ports – TCP & UDP
- Auditing Port Access
- Required PTFs
- Additional Resources
- Summary



Why Port Security with RACF?

NATIVE TCPIP

- Reservation by Stepname
 - Can be spoofed
 - Violations not well logged
- Unreserved ports not easily controlled
- Low Ports protected by RESERVELOWPORTS

RACF

- Reservation by SAFNAME
 - SAFNAME last qualifier of SERVAUTH portaccess profile
 - Cannot be spoofed
 - Successes or Violations logged to SMF (type 8o)
- Unreserved ports easily controlled
- Low Ports protected by RESERVELOWPORTS

EZB.PORTACCESS Profile Syntax

EZB. PORTACCESS. *sysname. tcpname. safname*

Qualifier	Description	Recommendation
sysname	Local SMF ID	<ul style="list-style-type: none">• Use * unless need for per system segregation
tcpname	TCPIP started task jobname	<ul style="list-style-type: none">• Use * unless multiple stacks
safname	Esoteric name coded in port reservation	<ul style="list-style-type: none">• Can be generic• Plan appropriately

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

Port Reservation Syntax – Single

- Port Reservation Syntax – single port

PORT

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

- With SAFNAME, stepname 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 LengthProtocol Stepname SAF SAFName  
1000 51 UDP * SAF OMEGAMON
```

- Reserves 1000 through 1050 for Omegamon

- Use Only One

- Reserve Individual Port
- Reserve Port Range

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
 - FAILURES(READ)
 - Audit only unauthorized attempted use of port
- WARNING
 - Can be used **but** makes port wide open – use carefully

Example: TN3270 – Reservation

PORT 23 TCP TN3270

- Non-SAF uses stepname

PORT 23 TCP * SAF TN3270

- With SAF
 - Stepname unnecessary
 - Only use stepname where needed

Planning – Gather Information

- Evaluate running STCs and their ports
 - REXX EXEC compare reservations vs. usage
- Create list of Port Listeners & *SAFnames*
- Partner with Network/VTAM Engineer
 - TCPIP profile changes
 - Weekend IPLs
- Implement one system at a time
 - development, test and then production

Planning – Implementation

- Build EZB.PORTACCESS profiles
 - SERVAUTH class must be active
 - TCP Ports – OMPROUTE READ
- Update port reservations to call SAF
- Activate via IPL or TCPIP OBEY
 - Simpler to IPL for a large number of STCs
 - OBEY command is dynamic
 - Cycle Started Tasks
 - Excludes FTP

Planning – Intermittent Listeners

- NETSTAT PORTLIST
 - Shows ports in use *now*
 - Not every port is in constant use by its listener
- Find *intermittent* port listeners
 - Setting WARNING AUDIT(ALL(READ))
 - EZB.PORTACCESS.*.*.UNRSVTCP
 - Mine SMF records
 - Midnight Logons – Optional
- 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...
 - SERVAUTH profiles for DVIPA
 - EZD1313I -REQUIRED SAF SERVAUTH PROFILE NOT FOUND RACF *profile name*
- RDEFINE RACGLIST SERVAUTH
 - Performance Improvement
 - SETR classact(RACGLIST) audit(RACGLIST)
 - SETR RACLIST(...) REFRESH Creates

Unreserved Ports Syntax

- `PORT UNRSV TCP * SAF UNRSVTCP`
 - Prevent TCP port listeners, **TCP default**
- `PORT UNRSV TCP * SAF UNRSVTCP WHENBIND`
 - Prevent TCP client port binds, **optional**
- `PORT UNRSV UDP * SAF UNRSVUDP`
 - Prevent UDP port listeners & binds, **UDP default**
- Stop Unauthorized Port Use
- Consideration: Dynamic Ephemeral UDP ports
 - UI8700 – z/OS 1.13
 - UI9430 – z/OS 2.1

Unreserved Ports Profile Syntax

- Build SERVAUTH Profiles
 - Match SAF keyword
EZB.PORTACCESS.*.***UNRSVTCP** OWNER(...)
UACC(NONE) WARNING AUDIT(ALL(READ))
 - EZB.PORTACCESS.*.***UNRSVUDP** OWNER(...)
UACC(NONE) WARNING AUDIT(ALL(READ))
 - Read SMF
 - Cautiously Restrict Access

Unreserved Ports – Challenges

- WAS Admin console scans ports
- z/OS FTP Client
 - If Passive FTP fails, attempt Active
 - Active connection Listens on TCP port
- UDP Ephemeral ports
 - Applications Need Dynamic Ephemeral UDP
 - STC desires to use SMTP to send email
 - STC opens UDP Ephemeral port → SMTP
 - Triggers SAF call unless:
 - UI8700 – z/OS 1.13
 - UI9430 – z/OS 2.1



Auditing Port Access

- LOGSTRING contains the port number
 - TCP / UDP Not Specified

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

Jobname + id: OMEGCMS STC48698

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

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

Log string : TCPIP PORT ACCESS CHECK PORT 01000

Required PTFs

- APAR PI18151 – Spurious SAF (RACF) Violations from use of UDP Sockets
 - UI8700 for z/OS 1.13 and UI9430 for z/OS 2.1
 - Use of UDP Ephemeral ports causes random security violations
 - <http://www-01.ibm.com/support/docview.wss?uid=isg1PI18151>
- APAR PI08351 – ABEND SoC4 IN EZBXFUT6
 - UI13629 for z/OS 1.13 and UI14006 for z/OS 2.1
 - PORT UNRSV TCP * SAF SAFname
 - Mapping via SRCIP to a DVIPA with sysplexports defined
 - Does not have permission to the SAF resource
 - <http://www-01.ibm.com/support/docview.wss?uid=isg1PI08351>

Additional Resources

- Techdocs Library – Using SERVAUTH to Protect TCP Port Usage
 - <http://www-03.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP100673>
- Port Access Control Chapter
 - z/OS Communications Server: IP Configuration Guide
 - <http://pic.dhe.ibm.com/infocenter/zos/v1r13/index.jsp?topic=%2Fcom.ibm.zos.r13.halzoo2%2Fportaccctrl.htm>
- SERVAUTH Class profiles used by TCP/IP
 - EZB.PORTACCESS syntax
 - <http://pic.dhe.ibm.com/infocenter/zos/v1r12/index.jsp?topic=%2Fcom.ibm.zos.r12.halzoo2%2Ff1a1b3a024.htm>

Summary

- Protecting Ports is of Paramount ImPORTance
 - Securing with RACF
 - prevent spoofing
 - log port usage (success & failures) to SMF
- Requires Proper Planning
- Close partnership with Network Engineer
- Coordinate TCPIP Profile & RACF Changes
- IPL during maintenance windows
- Fix ICH408ls and:
 - Recycle STC or possibly IPL
- Port Security Engaged!



Questions?

