

Your Friend SERVAUTH or How To Protect Your IP Stack

Hayim Sokolsky
April 2009
NY RACF Users Group
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.

Agenda

- SERVAUTH Info
- z/OS 1.10 SERVAUTH Profile list (44!)
- Details for a select few profiles
 - Stack, Network, Port Access
 - NETSTAT
 - MODDVIPA
 - SOCKOPT
- Questions

SERVAUTH Intro

- TCP/IPish equivalent of the FACILITY class
 - Protects a myriad of things IP
 - Spans IBM products
 - Profile HLQ is IBM component prefix
 - EZA DCAS
 - EZB Most Communication Server components
 - IRR RACF
 - IST VTAM
 - New profile formats added each release
 - 42 at z/OS 1.9, 44 at z/OS 1.10 ...

SERVAUTH Intro

- Common SERVAUTH-isms
 - Many profiles contain qualifiers with
 - sysname* SMFID (SYSID) of the LPAR
 - tcpname* Jobname of the TCP/IP stack
 - ftpdaemonname* Jobname of FTP daemon
 - Standard use of generics
 - Replace above with "*" as appropriate
 - You may have one or more TCP/IP stacks
 - Determined by your network system programmers

SERVAUTH Profiles

- z/OS 1.10 profile list is on the next few slides
 - Profiles in **bold** are covered in this presentation
 - Due to time considerations only a few key profile types are covered
 - Please review the whole profile list prior to implementation to determine which profiles are appropriate or most beneficial to your own environment

z/OS 1.10 SERVAUTH Profiles (1)

EZA. DCAS. *sysname*

EZB. BI NDDVI PARANGE. *sysname. tcpname*

EZB. CI MPROV. *sysname. tcpname*

EZB. FRCAACCESS. *sysname. tcpname*

➔ EZB. FTP. *sysname. ftpdaemonname. ACCESS. HFS*

EZB. FTP. *sysname. ftpdaemonname. PORT xxxxx*

➔ EZB. FTP. *sysname. ftpdaemonname. SI TE. DEBUG*

➔ EZB. FTP. *sysname. ftpdaemonname. SI TE. DUMP*

EZB. I NI TSTACK. *sysname. tcpname*

EZB. I PSECCMD. *sysname. DMD_GLOBAL. command_type*

EZB. I PSECCMD. *sysname. tcpname. command_type*

z/OS 1.10 SERVAUTH Profiles (2)

➔ EZB. MODDVI PA. *sysname. tcpname*

➔ EZB. NETACCESS. *sysname. tcpname. security_zonename*

EZB. NETMGMT. *sysname. clientname. IPSEC. CONTROL*

EZB. NETMGMT. *sysname. clientname. IPSEC. DISPLAY*

EZB. NETMGMT. *sysname. sysname. IKED. DISPLAY*

EZB. NETMGMT. *sysname. tcpname. IPSEC. CONTROL*

EZB. NETMGMT. *sysname. tcpname. IPSEC. DISPLAY*

➔ EZB. NETMGMT. *sysname. tcpname. SYSTCPCN*

➔ EZB. NETMGMT. *sysname. tcpname. SYSTCPDA*

➔ EZB. NETMGMT. *sysname. tcpname. SYSTCPSM*

EZB. NETMGMT. *sysname. sysname. NSS. DISPLAY*

z/OS 1.10 SERVAUTH Profiles (3)

EZB. NSS. *sysname. clientname. IPSEC. CERT*

EZB. NSS. *sysname. clientname. IPSEC. NETMGMT*

EZB. NSS. *sysname. clientname. XMLAPPLIANCE. SAFACCESS*

EZB. NSSCERT. *sysname. mappedlabel name. CERTAUTH*

EZB. NSSCERT. *sysname. mappedlabel name. HOST*

➔ EZB. NETSTAT. *sysname. tcpname. netstat_option*

EZB. PAGENT. *sysname. image. ptype*

➔ EZB. PORTACCESS. *sysname. tcpname. port_safname*

EZB. SNMPAGENT. *sysname. tcpname*

z/OS 1.10 SERVAUTH Profiles (4)

EZB. SOCKOPT. *sysname*. *tcpname*. IPV6_DSTOPTS
EZB. SOCKOPT. *sysname*. *tcpname*. IPV6_HOPLIMIT
EZB. SOCKOPT. *sysname*. *tcpname*. IPV6_HOPOPTS
EZB. SOCKOPT. *sysname*. *tcpname*. IPV6_NEXTHOP
EZB. SOCKOPT. *sysname*. *tcpname*. IPV6_PKTINFO
EZB. SOCKOPT. *sysname*. *tcpname*. IPV6_RTHDR
EZB. SOCKOPT. *sysname*. *tcpname*. IPV6_RTHDRDSTOPTS
EZB. SOCKOPT. *sysname*. *tcpname*. IPV6_TCLASS
➔ EZB. SOCKOPT. *sysname*. *tcpname*. SO_BROADCAST

➔ EZB. STACKACCESS. *sysname*. *tcpname*

EZB. TN3270. *sysname*. *tcpname*. PORTxxxxx
IRR. HOST. *host-name*
IST. NETMGMT. *sysname*. SNAMGMT

Stack Access

EZB. STACKACCESS. *sysname*. *tcpname*

- Controls access to IP stack – open socket
 - Primarily outbound control
 - TCP/IP daemons – FTP, Web servers, etc...
 - FTP end-users (inbound *and* outbound)
 - Sample configurations
 1. UACC(READ) with AUDIT(ALL(READ))
 - Log all use
 2. UACC(NONE) with appropriate permits
 - Restrict use

Network Access

EZB. NETACCESS. *sysname. tcpname. security_zonename*

- Inbound and/or outbound access control
 - Defined in TCP/IP NETACCESS statements
 - Inbound
 - Point Of Entry (POE) control like TERMINAL, etc...
 - Checked for Websphere, FTP, NFS, Distributed DB2...
 - Enables WHEN(SERVAUTH(*profile_name*))
 - SERVAUTH info logged for all generated SMF records
 - Outbound
 - Controls outbound traffic

NETACCESS vs. Firewall

SERVAUTH NETACCESS

- ✓ Controls IP access
- ✓ "Here" to target IP
- ✗ No port control
- ✓ UserID specific
- 👉 Does not replace Firewall

FIREWALL

- ✓ Controls IP access
- ✓ Source IP to target IP
- ✓ Source/target ports
- ✗ Any mainframe user
- 👉 Does not replace NETACCESS

NETACCESS Statement

- Defines security zones for use in SERVAUTH

```
NETACCESS [ INBOUND ] [ OUTBOUND ]
```

```
  [ ip_in_cidr_notation      zonename ... ]
```

```
  [ DEFAULTHOME              zonename ]
```

```
  [ DEFAULT                  zonename ]
```

```
ENDNETACCESS
```

- IP address is IPv4 or IPv6
- DEFAULTHOME – all IPs of this stack
- DEFAULT – any IP not defined above

NETACCESS Sample (1)

```
NETACCESS INBOUND OUTBOUND
 10. 0. 0. 0/8      INTERNAL
 10. 1. 0. 0/16    I SITE NY1
 10. 2. 0. 0/16    I SITE FL2
 172. 3. 0. 0/16    I SRVR1
 172. 4. 0. 0/16    I SRVR2
 192. 168. 1. 0/24  PODUNK
 207. 25. 253. 24/32 I BMPUBS
DEFAULTHOME        LOCAL
DEFAULT            EXTNET
ENDNETACCESS
```

- Assumes 10.*, 172.0.* – 172.31.*, and 192.168.* are used internally in the sample network as standard non-routable (non-internet) IP addresses.

NETACCESS Sample (2)

```
RDEF SERVAUTH EZB.NETACCESS. **          UACC(NONE)
RDEF SERVAUTH EZB.NETACCESS. * . * . I *  UACC(NONE)
RDEF SERVAUTH EZB.NETACCESS. * . * . PODUNK UACC(NONE)
RDEF SERVAUTH EZB.NETACCESS. * . * . I BMPUBS UACC(READ)
RDEF SERVAUTH EZB.NETACCESS. * . * . LOCAL  UACC(READ)
RDEF SERVAUTH EZB.NETACCESS. * . * . EXTNET  UACC(NONE)
```

```
PE EZB.NETACCESS. * . * . I * CLASS(SERVAUTH)          +
  ID(*) ACCESS(READ)
PE EZB.NETACCESS. * . * . PODUNK CLASS(SERVAUTH)       +
  ID(PODFTPGP) ACCESS(READ)
PE EZB.NETACCESS. * . * . EXTNET CLASS(SERVAUTH)       +
  ID(WEBSERV EXTUSERS) ACCESS(READ)
PE 'XMIT.DATASET' ID(PODFTPGP) ACCESS(READ)           +
  WHEN(SERVAUTH(EZB.NETACCESS. * . * . PODUNK))
```


NETACCESS Sample (3)

- Sample NETACCESS & RACF profiles
 - Internal IPs limited to defined (non-RESTRICTED) users
 - Only PODFTP GP users can talk to "PODUNK"
 - WEBSERV & EXTUSERS can use internet addresses
 - PODFTP GP gets read to DATASET only when entering from "PODUNK" IP range.

NETACCESS vs. TERMINAL

- FTP daemon uses
 - NETACCESS for IPv6 only
 - NETACCESS or TERMINAL for IPv4
 - Set in FTP daemon configuration
 - TERMINAL uses 8 character hex for IP address
 - 172.1.42.1 becomes AC012A01
 - Can use generics for whole digit AC012A%%
 - Not as flexible as NETACCESS – 172.1.42.1/25
 - Covers 172. 1. 42. 0 to 172. 1. 42. 127

Port Reservations

EZB. PORTACCESS. *sysname. tcpname. port_safname*

- Reserves ports for services by UserID
 - Non-SAF definitions based upon jobname
 - No violation logging
 - RESERVELOWPORTS (in TCPCONFIG and UDPCONFIG) protects low ports (1 to 1023) to privileged users if not specified in port reservation
 - SAF keyword turns reservation into SERVAUTH resource check.

PORT /PORTRANGE Statement

PORT

[*port# type jobname* [SAF *safname*] ...]

PORTRANGE

[*1st #ports type jobname* [SAF *safname*] ...]

- PORT defines one port at a time
- PORTRANGE defines range (like 2000-2099)
- *type* is TCP or UDP

PORT /PORTRANGE Sample

PORT

21 TCP FTPD

80 TCP * SAF HTTPD

443 TCP * SAF HTTPD

- Port 21 is reserved by jobname
- Ports 80 & 443 are authorized by SERVAUTH
EZB.PORTACCESS.*sysid.tcpname*.HTTPD

```
RDEF SERVAUTH EZB.PORTACCESS.*.*.HTTPD UACC(NONE) +
    AUDIT(FAILURES(READ))
PE EZB.PORTACCESS.*.*.HTTPD CLASS(SERVAUTH) +
    ID(STCHHTTPD) ACCESS(READ)
```

NETSTAT Command

EZB.NETSTAT. *sysname. tcpname. netstat_option*

- Controls use of netstat command and its functions
 - Allows restriction or logging of functions
 - Assuming EZB.NETSTAT.** as profile
 - Restriction via UACC(NONE) + PERMIT
 - Logging via UACC(READ) with AUDIT(SUCCESS(READ))

```
RDEF SERVAUTH EZB.NETSTAT. ** UACC(NONE)
PE EZB.NETSTAT. ** CLASS(SERVAUTH)
ID(SYSPROG NETPROG) ACCESS(READ) +
```

MODDVIPA Command

EZB. MODDVI PA. *sysname. tcpname*

- Controls use of MODDVIPA utility
 - Not intended for general user use
 - Changes VIPA (virtual IP adapter) configuration

```
RDEF SERVAUTH EZB. MODDVI PA. ** UACC(NONE)
PE EZB. MODDVI PA. ** CLASS(SERVAUTH)
  ID(SYSPROG NETPROG) ACCESS(READ)
```

+

Network Management

```
EZB. NETMGMT. sysname. tcpname. SYSTCPCN  
EZB. NETMGMT. sysname. tcpname. SYSTCPDA  
EZB. NETMGMT. sysname. tcpname. SYSTCPSM
```

- Allows use of network management interfaces (TCP info, packet trace, SMF) for vendor products
 - Removes UI D(0) need for some vendor products

```
RDEF SERVAUTH EZB. NETMGMT. *. *. SYSTCP* UACC(NONE)  
PE EZB. NETMGMT. *. *. SYSTCP* CLASS(SERVAUTH) +  
  ID(IPMGMT) ACCESS(READ)
```


FTP

```
EZB. FTP. sysname. ftpdaemonname. ACCESS. HFS
EZB. FTP. sysname. ftpdaemonname. SITE. DEBUG
EZB. FTP. sysname. ftpdaemonname. SITE. DUMP
```

- Controls FTP sensitive functions (SITE) and HFS access

```
RDEF SERVAUTH EZB. FTP. *. *. ACCESS. HFS UACC(NONE)
PE EZB. FTP. *. *. ACCESS. HFS CLASS(SERVAUTH) +
  ID(FTPHFS) ACCESS(READ)
```

```
RDEF SERVAUTH EZB. FTP. *. *. SITE. * UACC(NONE)
PE EZB. FTP. *. *. SITE. * CLASS(SERVAUTH) +
  ID(SYSPROG NETPROG) ACCESS(READ)
```

Sock Options

EZB. SOCKOPT. *sysname. tcpname.* SO_BROADCAST

- Sock Options – SO_BROADCAST controls use of broadcast datagrams

```
RDEF SERVAUTH EZB. SOCKOPT. **. SO_BROADCAST UACC(NONE)
```

```
PE EZB. SOCKOPT. **. SO_BROADCAST CLASS(SERVAUTH) +  
  ID(SOBRCAST) ACCESS(READ)
```

Questions?

- Thanks for coming!