ListLib
13. December 2011

ListLib is a product from GreenHouse.
It prevents a Tandem[1] system against a Denial of Service (DoS) attack coming through the IP network by controlling the activities of LISTNER.

Critical ports LISTNER is listening to are:

- FTPSERV (Port 21)
  A possible attacker can use an FTP client to start as many FTPSERV processes on the Tandem system, until the number of processes, or swap space, or other resources, are exhausted, and no further service from the system is granted.
  This method is the one and only way to check the system for IDs and passwords WITHOUT the need to have an interactive access first!
- ECHOSERV (Port 7)
  The ECHO service is a nice function to test the way to a host, and the way back to the user, by sending a user message to the host, which is echoed back.
  The number of ECHO sessions is NOT restricted, thus there as well is a chance to run a DoS attacks against the Tandem system by creating as many ECHO servers as possible until system resources are exhausted.
- FINGER
  is critical as well because it discloses system status information to a not authenticated user: It should be turned off in PORTCONF!

Function

ListLib is designed as a library. It becomes part of the LISTNER product on the Tandem system and intercepts procedure calls to get the:

- IP address of the calling instance and
- File name of the requested resource, e.g. ECHOSERV, FTPSERV etc.

It as well checks the calling IP address, which is controlled by a

- white list, where it has to be mentioned
- black list, where it is not allowed to be mentioned.

An IP address, which does not pass this check, is rejected right away.

The library does not react proactive in case a DoS happens, but it prevents the Tandem system from running into trouble by that type of attack.

The logging cannot record a user, because there is no known user when the request is recognized, but it records the remote IP address!

[1] To me it still is a Tandem after 33+ years on the platform!

IP address check
When a request is coming into ListLib, it is check is based on two lists:

1. Allowed IP addresses (white list)
2. Denied IP addresses (black list)

To successfully get access to the system, the calling IP address has to be:

- known to the white list, and
- unknown to the black list.

This mechanism enables an easy usage of templates, e.g.:

- IPOK is 100.10.*.*
- IPNOK is 100.10.0.56

In this case, only IP addresses in the range of 100.10.*.* are allowed, except address 100.10.0.56.
This is the first check, and it is independent of the requested service.
The IP addresses can be configured in the EDIT type configuration file LISTCONF.

ECHO services
ListLib allows two ECHO sessions on an IP stack. Any additional ECHO session request is rejected.
This is hard coded.

FTP services
Before the request to start an FTPSERV process is executes, then number of already running FTPSERV processes is checked.
In case the

- number of running FTPSERV processes exceeds the number of configured ones, or
- the number of logged off FTPSERV processes exceeds the configured number

no new FTPSERV process is created, and an error message is optionally sent to the EMS system.
The numbers can be configured in the EDIT type configuration file LISTCONF.

LISTCONF

To make ListLib as flexible as possible, the following attributes can be configured through an EDIT type file named LISTCONF.
LISTCONF has to reside in the same location in which LISTLIB resides.
The following key words are supported:

EMS
The generation of EMS messages can be turned on or off.
Valid values are: ON and OFF.
Default is: ON

EMSCOLLECTOR
All LISTLIB actions are reported to the EMS system. In case the EMS collector is not $0, its name has to be configured here.
In case the entry is missing, $0 is assumed as the default collector process.
Default is: $0

IPOK
A requesting IP address is checked against a list of up to 100 IP addresses. This address has to match one of the entries. In case no match is found, the caller is rejected, and an EMS event is created.
Up to 100 positive IP addresses can be configured.
Wildcards are allowed, e.g. IPOK 192.231.*.23?
*Default is: IPOK *.*.*.* (= all IP addresses are allowed)*

IPNOK
A requesting IP address is checked against a list of up to 100 IP addresses. This address is NOT allowed to match one of the entries. In case a match is found, the caller is rejected, and an EMS event is created.
Up to 100 positive IP addresses can be configured.
Wildcards are allowed, e.g. 192.231.*.23?
Default is: No entry is configured (= NO IP addresses is denied)

LOGGING
ListLib can log its activities into a log file.
The log file resides in the location of the LISTLIB object file, and is named LISTLOG0.
This can be turned ON or OFF.
Default is: OFF

MAXNUMPROCS
Before LISTNER denies the creation of a new process, a defined number of processes already has to exist. The number can be any value between 1 and 1000. In case the number is out of range, it is assumed to be 5.
Default is: 5

MAXNUMLOGGEDOFF
Before LISTNER denies the creation of a new process, a defined number of still logged off processes might exist. The number can be any value between 1 and 1000. In case the number is out of range, it is assumed to be 5.
Default is: 5

PROGRAMFILE
The LISTNER library needs to know program file names to find the number of already running processes, derived from a given object. To make this as flexible as possible, you can define the program file names as templates.
All wildcard characters are supported.
Entries are NOT case sensitive.
The number of program file templates is limited to 20.
*Default is: *.*FTPSERV**

The LISTCONF EDIT type file has to reside in the same location as the LISTLIB file.

Changes in the library are automatically taken into account: Restarting the LISTNER process is NOT required.

The execution of LISTCONF is always reported to the EMS system.

Installation

To add this library to the LISTNER process, perform these steps:

1. Check the file code of LISTLIB on your system:
   `FILEINFO $SYSTEM.SYSnn.LISTNER`

2. Depending on the file code of LISTNER, down load one of these LISTLIB files in BINARY mode to your system:
   - **`LISTLIB.100`**
   - **`LISTLIB.700`**
   - **`LISTLIB.800`**

   and one of these LISTFTP files:
   - **`LISTFTP.100`**
   - **`LISTFTP.700`**
   - **`LISTFTP.800`**

   into an empty subvol.
   It is recommended, to use location $SYSTEM.LISTLIB

3. Download the file LISTTOK.101 in ASCII mode into the same location, and name is LISTTOK.

4. Download the file LISTCONF.101 in ASCII mode into the same location, in which you loaded LISTLIB and LISTFTP.

5. EDIT the LISTCONF configuration file and adjust it to your needs.

6. Stop the LISTNER process(es) in question.
   This does NOT interrupt already running services, already started by LISTNER.
   While the LISTNER is stopped, no new LISTNER controlled sessions can be started.

7. Use the BINDLIB[2] tool to bind the LISTLIB product to the LISTNER:
   `BINDLIB [/OUT <file>/] $SYSTEM.SYSnn.LISTNER WITH $vol.subvol.LISTLIB`

8. Use the SHOWLIB[3] tool to check, that LISTLIB is successfully bind to LISTNER:
   `SHOWLIB [/OUT <file>/] $SYSTEM.SYSnn.LISTNER`

9. Re-start the LISTNER process(es)

---

[2] BINDLIB is a FreeWare tool from Greenhouse. It can be found at www.GreenHouse.de
[3] SHOWLIB is a FreeWare tool from GreenHouse as well and can be found at www.GreenHouse.de (where else!)

De-Installation

1. Stop the LISTNER process(es) in question.
   This does NOT interrupt already running services, started by LISTNER.
   While the LISTNER is stopped, NO new LISTNER controlled sessions can be started.

2. Use the BINDLIB tool to un-bind the LISTLIB product to the LISTNER:
   `BINDLIB [/OUT <file>/] $SYSTEM.SYSnn.LISTNER`

3. Use the SHOWLIB tool to check, that LISTLIB is no longer bound to LISTNER:
   `SHOWLIB [/OUT <file>/] $SYSTEM.SYSnn.LISTNER`

4. Re-start the LISTNER process

LISTLOG

ListLib is able to record all its action into a log file.
This file is automatically created when needed, and named LISTLOG0.
When it runs full, it is renamed to LISTLOG1, and a new LISTLOG0 is created.
Up to 10 log files can exists: LISTLOG0 ..LISTLOG9.
When LISTLOG9 exists, and LISTLOG0 runs ful, LISTLOG9 is purged.

To evaluate the LISTLOGx files, the ENFORM Query LISTLOG is supplied.
A typical output looks like this:

```
Event Time (LCT)   IP Check       Action         IP Address              Resource
-----------------  --------  ---------------  --------------  -------------------------

19Dec2011 11:17:10  OK        Resource started  192.231.36.101   $SYSTEM.ZTCPIP.ECHOSERV
19Dec2011 11:17:32  OK        ECHO rejected     192.231.36.101   $SYSTEM.ZTCPIP.ECHOSERV
19Dec2011 11:17:37  OK        ECHO rejected     192.231.36.101   $SYSTEM.ZTCPIP.ECHOSERV
19Dec2011 11:18:06  OK        ECHO rejected     192.231.36.101   $SYSTEM.ZTCPIP.ECHOSERV
19Dec2011 11:18:16  OK        Resource started  192.231.36.101   $SYSTEM.ZTCPIP.ECHOSERV
19Dec2011 11:27:11  NOK       IP rejected       192.231.36.90    $SYSTEM.ZTCPIP.FTPSERV
19Dec2011 11:27:36  NOK       IP rejected       192.231.36.90    $SYSTEM.ZTCPIP.FTPSERV
19Dec2011 11:28:06  NOK       IP rejected       192.231.36.90    $SYSTEM.ZTCPIP.FTPSERV
19Dec2011 11:28:36  OK        Resource started  192.231.36.101   $SYSTEM.ZTCPIP.FTPSERV
19Dec2011 11:29:32  OK        Resource started  192.231.36.101   $SYSTEM.ZTCPIP.ECHOSERV
```

LISTFTP

LISTLIB keeps track of the FTPSERV processes in a disk file, named LISTDAT. This file is located in the same location as the LISTLIB object file, and created automatically.
LISTFTP can be used to display all active FTPSERV sessions along with some session attributes:

```
$GHS1 LISTNER 128> listftp
  IP  Address        Start Time           Object File Name       Logged On
--------------  -----------------  -------------------------  ---------
192.231.036.081  19 Feb 2004, 13:36  $GHS1.FTP.FTPSERVT             No
$GHS1 LISTNER 129>
```

LISTFTP has to reside in the same location in which LISTLIB resides.