AFLEX TCL CODE INJECTION EXPOSURES

PUBLISHED: SEPTEMBER 9, 2019 | LAST UPDATE: OCTOBER 2, 2019

SUMMARY

Capabilities and selected syntaxes of the Tool Command Language (Tcl) allow or disallow substitutions in various Tcl program statements and commands from the arbitrary data being processed. Tcl coding practices in ACOS aFlex scripts that allow substitutions can expose these scripts to be vulnerable to the unintentional injection of arbitrary Tcl and aFlex commands from the untrusted content of the data stream being processed.

Coding practices that ensure Tcl expressions are enclosed with curly braces '{ }' will disallow substitutions and eliminate potential exposures to command injections from the data stream in aFlex Tcl scripts. An additional benefit of this practice will be improved aFlex performance (reduced overhead) [3].

The developers of Tcl, Tcl Developer Xchange [1], advised the industry of such substitution considerations and related injection attacks on their Tcler's Wiki [2] with their *Brace your expr-essions* (4/2015) [3] , *double substitution* (1/2016) [4], and *Injection Attack* (3/2014) [5] wiki pages. These wiki pages provide insights on the Tcl coding considerations involved and include many clear examples of good and bad coding practices.

Tcl statements in aFlex scripts at potential risk of substitutions and injection exposures include:

`catch`	`history`	`set`	`trace`
`eval`	`if`	`stringmatch`	`while`
`expr`	`list`	`switch`	
`for`	`regexp`	`subst`	
`foreach`	`regsub`	`time`	

Overall, this is not a vulnerability in ACOS or Tcl. Exposures in ACOS systems due to this vulnerability are attributable to at-risk coding practices in the Tcl code used in configured aFlex scripts.

Vulnerabilities arising from such exposures in aFlex Tcl scripts are constrained in ACOS, as several Tcl commands are disabled and not supported. These constraints include limiting aFlex from:

- accessing or modifying internal data (transient or permanent) within the ACOS system and
- creating connections independent of the underlying connection or data stream being processed by the script.

Though these constraints significantly limit the range and scope of malicious exploits of this vulnerability on the ACOS system, it does not eliminate or fully protect against them.

Tcl commands excluded and unavailable in ACOS aFlex include the following:

`after`	`exec`	`interp`	`seek`
`auto_execok`	`exit`	`load`	`socket`
`auto_import`	`fblocked`	`memory`	`source`
`auto_load`	`fconfigure`	`namespace`	`tcl_findLibrary`
`auto_mkindex`	`fcopy`	`open`	`tell`
`auto_mkindex_old`	`file`	`package`	`unknown`
`auto_qualify`	`fileevent`	`pid`	`update`
`auto_reset`	`filename`	`pkg::create`	`uplevel`
`bgerror`	`flush`	`pkg_mkIndex`	`upvar`
`cd`	`gets`	`proc`	`vwait`
`close`	`glob`	`pwd`	
`eof`	`http`	`rename`	

AFFECTED RELEASES

All ACOS releases supporting aFlex are potentially vulnerable to these substitution exposures in Tcl scripts that do not include appropriate bracing considerations in the underlying Tcl code.

This is not a vulnerability in ACOS or Tcl. Exposures in ACOS systems due to this vulnerability are attributable to at-risk coding practices in the Tcl code of configured aFlex scripts.

WORKAROUNDS AND MITIGATIONS

All aFlex Tcl scripts should be reviewed to ensure they are free from exposure to this vulnerability. Where Tcl expressions are found to contain non-braced statement parameters, these instances should be scrutinized to determine if they are non-braced situations not exposed to substitution attacks or unintentional coding constructs that should/can be braced. Substitution exposures identified should then be repaired in the code of aFlex scripts and any related logic in the code adjusted accordingly for deployed ACOS systems.

An open source tool, 'tclscan' [6] can help to identify potentially exposed Tcl statements. 'tclscan' is generally considered to identify ~80% of at-risk Tcl statements. Accordingly, diligence in the code review process and vetting of deployed ACOS systems is important and critical for Tcl aFlex scripts.

Commercial safe coding practice tools, such as Coverity and other alternatives, can also contribute to the review of aFlex Tcl scripts and surface potentially at-risk statements.

SOFTWARE UPDATES

Not applicable for this advisory. This is not a vulnerability in ACOS or its underlying Tcl implementation.

VULNERABILITY DETAILS

See the information available from the *Related Links* section below.

RELATED LINKS

Ref #	General Link
[1]	Tcl Developer Xchange
[2]	Tcler's Wiki
[3]	Tcler's Wiki - Brace your expr-essions
[4]	Tcler's Wiki - double substitution
[5]	Tcler's Wiki - Injection Attack
[6]	tlscan - Scans tcl for command injection

ACKNOWLEDGEMENTS

None

MODIFICATION HISTORY

Revision	Date	Description
1.0	2019-09-09	Initial Publication
1.1	2019-10-02	Grammar and phrasing updates