REPORTHACKING

@JasonMOliver

COMMAND-LINE FUN WITH NESSUS RESULTS

* * * *

Why?

- * Q: Why parse Nessus data and other tool output data? They have reporting, right? Even custom reporting!?
- * A: IMO Only admins and techs use the tools in a mode that makes these features valuable.
 - * Managers and customers expect analysis.
 - * Also my world has reporting format requirements; a PDF from a scanner is not going to cut it.

Nessus Format Structures

- * Reporting formats available: NBE, Nessus, HTML, XML
- * .nbe Legacy pipe delimited format
 - * Positive: Flat file allows for simple command-line parsing
 - * Negative: Last Field is a blob with a variable format structure. Severity data is lacking.
- * .nessus XML structure output archive format
 - * Positive: Data is segmented correctly and rich content
 - * Negative: .nessus has V1, V2 & a command-line XML version all different. XML takes code to parse.

Scan Types / Analysis Methods

- * Vulnerability Scanning
- * Compliance Scanning
- * Validation Scanning and of Scanning (aka Rescan and Artifacts)

Legacy Approach (NBE)

- * Why is NBE Legacy? To get to NBE from an exported artifact you now need to import the file (one at a time!!! FFS) to a web console, export to .nessus v1 file format, then use the command-line to convert to .nbe
- * nessus -i input-file.nessus -o output-file.nbe
- * Note: This can get more complicated with multi report .nessus files use the help for details.

Command-Line NBE Tricks

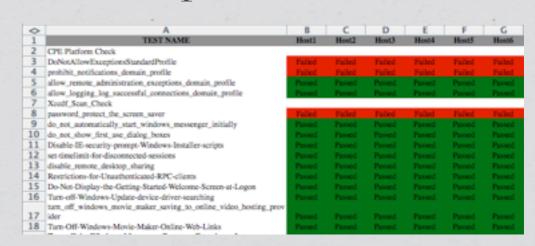
- * Targets Scanned: awk-F'|'(print \$3) filename.nbe | sort-u
- * Live Targets Scanned: grep '| 10180 | 'filename.nbe | awk -F ' | ' '{print \$3}' | sort -u
- * OS Groups & Target Count: grep '| 11936 | 'filename.nbe | awk-F'|' '{print \$3, \$7}' | awk -F'\' '{print \$2}' | sed 's/nRemote operating system : //' | sort | uniq-c
- * Target & OS: grep '|11936|' filename.nbe | awk-F'|''{print \$3, \$7}' | awk-F'\'''{print \$1, \$2}' | sed 's/ nRemote operating system :/,/' | sort-u
- * Scan vs. Inventory Compare (aka Missed Host & Extra Host): grep -x -v -f scan.txt inventory.txt
- * Vuln Counts: grep ' | Security Hole | ' *.nbe | awk -F ' | ' '{print \$5}' | sort -u | wc -l
- * Scrub Plugin Output: sed 's/Plugin output:/#Plugin output:/g' filename.nbe | sed 's/CVE:/#CVE:/g' | sed 's/Other references:/#Other references:/g' | awk-F'#' '{print \$1, \$3, \$4}'

Vulnerability Results

- * XMLTable A .nessus XML command-line parser to build a standard vulnerability table for reports. Supports multiple input files for merged reports.
- * java XMLTable inputFile.nessus > outputFile.[html/xls]
- * Creates a table of unique vulnerabilities; PluginID, RiskFactor, CVSS Score, Synopsis, Detail, Solution, Plugin Publish Date, Exploit Ease, Host List

Baseline Compliance Results

- * XMLCompTable a .nessus XML command-line parser that gives a x (hosts scanned), y (tests conducted) view of the data contained in the input files. When scanning many hosts this allows you to identify trends in baselines much quicker in addition to a simple way of spotting misconfigured hosts.
- * java XMLCompTable input-file.nessus > output-file.[xls/html]



Validation of Artifacts

- * XMLValidate A command-line query of a XML file that validates a .nessus artifact (some scan output) could support validation that a item found in the past was fixed.
- * First Was the pluginID(s) scanned for in the file?
- * Second Was it found on any hosts in the scan output?
- * Third What was scanned?
- * java XMLValidate inputFile.nessus pluginID(s)

Validation Cont. (Output)



java XMLValidate ScanInput.nessus 30218

PluginID: 30218 was located as item 11903 scanned for in the plugin_set.

----> PluginID 30218 was identified on host 10.10.10.1

----> PluginID 30218 was identified on host 10.10.10.2

Scanned Hosts:

10.10.10.1

10.10.10.2

10.10.10.3

10.10.10.4

10.10.10.5

Or in the case the file is clean:

java XMLValidate ScanInput.nessus 30218

PluginID: 30218 was located as item 11903 scanned for in the plugin_set. ----> PluginID 30218 was NOT identified on any scanned host.

Scanned Hosts:

10.10.10.1

10.10.10.2

10.10.10.3

10.10.10.4

10.10.10.5

Whats Next?

- * Statistical Analysis based on CVSS scores with NIST Risk Assessment Output Tables.
- * NIST Compliant Compliance results tables linking the findings to NIST SP 800-53a Control Test findings.
- * Unique vulnerability findings by host over a set of scans / or period of time.
- * Other Ideas?

Source Files

- * XMLTable http://www.blackhat.org/JSN/Blog/Entries/ 2010/12/10_Multiple_Source_Files_files/XMLTable.java
- * XMLCompTable http://www.blackhat.org/JSN/Blog/ Entries/2010/12/10 Multiple Source Files files/ XMLCompTable.java
- * XMLValidate http://www.blackhat.org/JSN/Blog/Entries/2010/11/28_Rescan_Validation_files/XMLValidate.java

Questions

Request: Looking for a mathematician or statistics guru to ping algorithms and formulas off -- if you know one point them my way plz

kthxbye