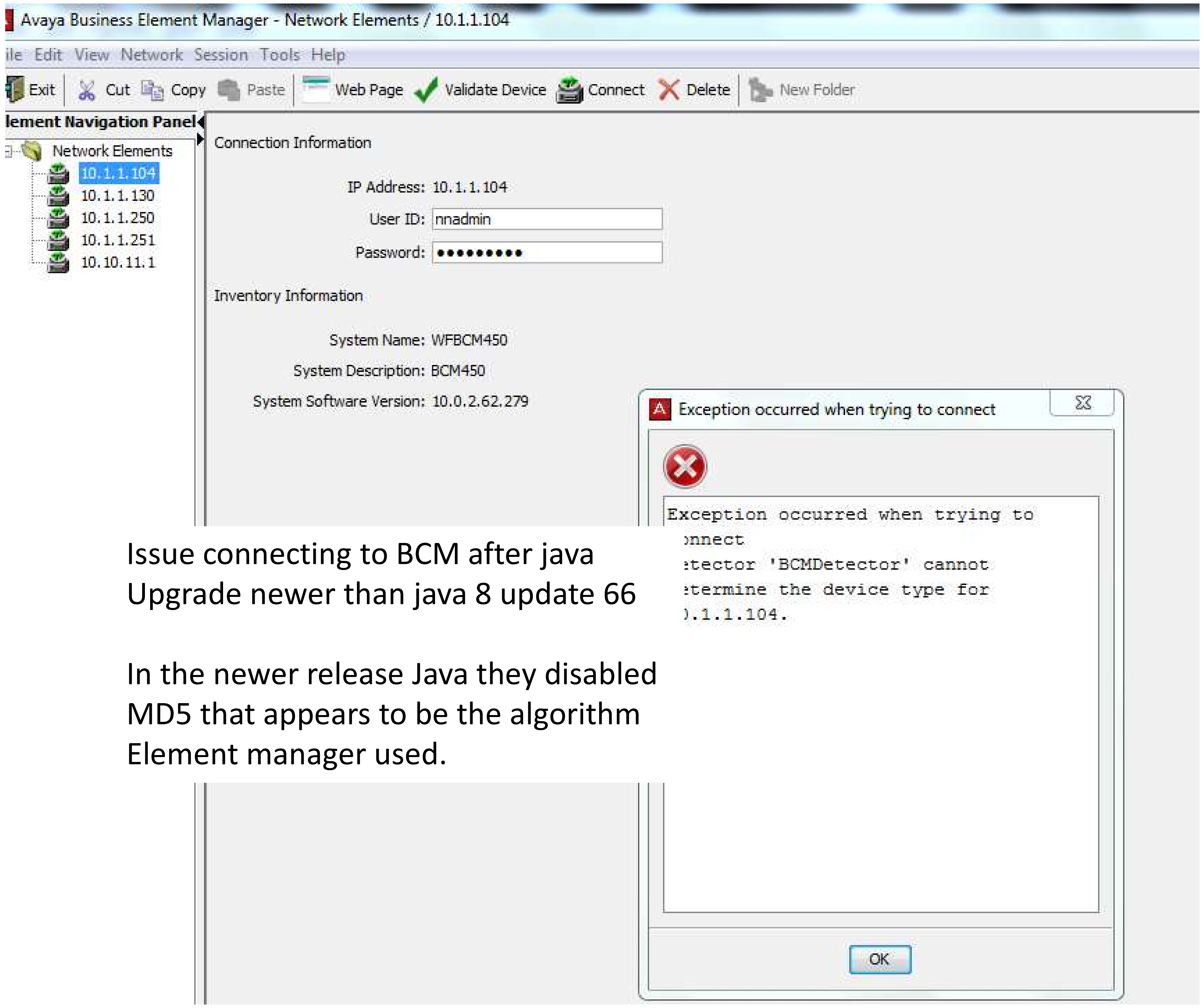

Avaya Business Element Manager - Network Elements / 10.1.1.104
ile Edit View Network Session Tools Help
Exit Cut Copy Paste Web Page Validate Device Connect Delete New Folder
lement Navigation Panel
Network Elements
10.1.1.104
10.1.1.130
10.1.1.250
10.1.1.251
10.10.11.1
OK

Connection Information

IP Address: 10.1.1.104

User ID: nnadmin

Password: ••••••••••

Inventory Information

System Name: WFBCM450

System Description: BCM450

System Software Version: 10.0.2.62.279

Exception occurred when trying to connect

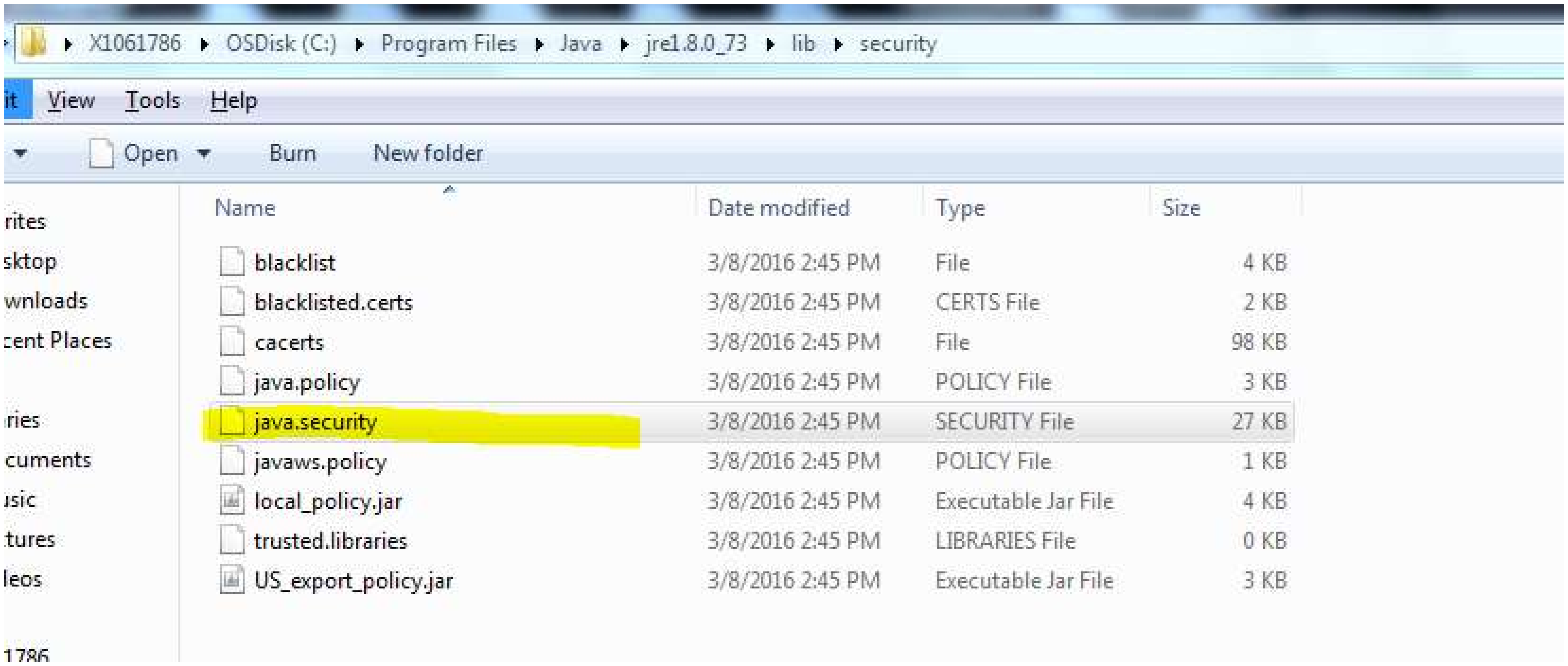
```
Exception occurred when trying to
onnect
etector 'BCMDetector' cannot
etermine the device type for
).1.1.104.
```

Issue connecting to BCM after java Upgrade newer than java 8 update 66

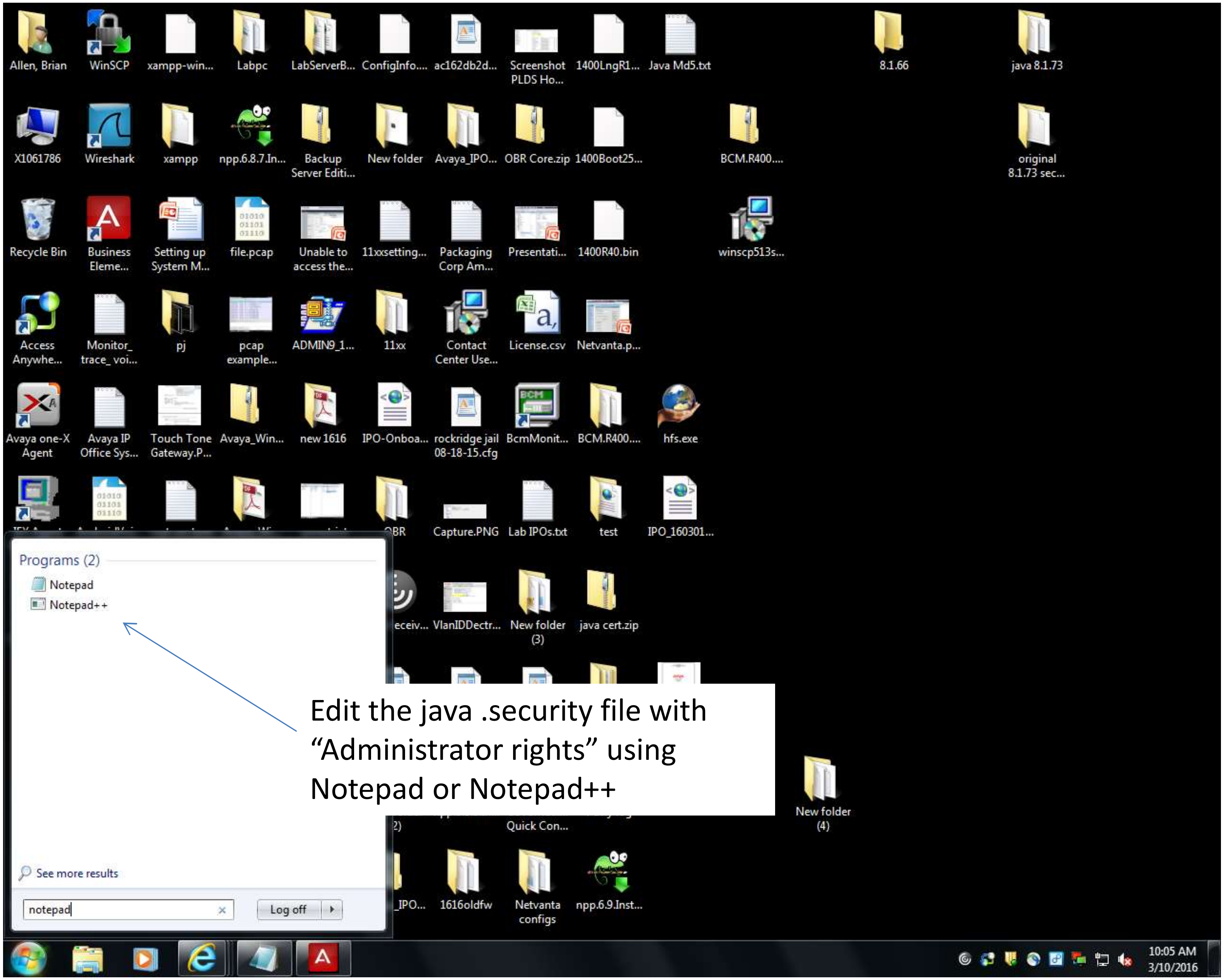
In the newer release Java they disabled MD5 that appears to be the algorithm Element manager used.

Before you begin, make a copy the java.security file someplace safe to revert Back if needed.

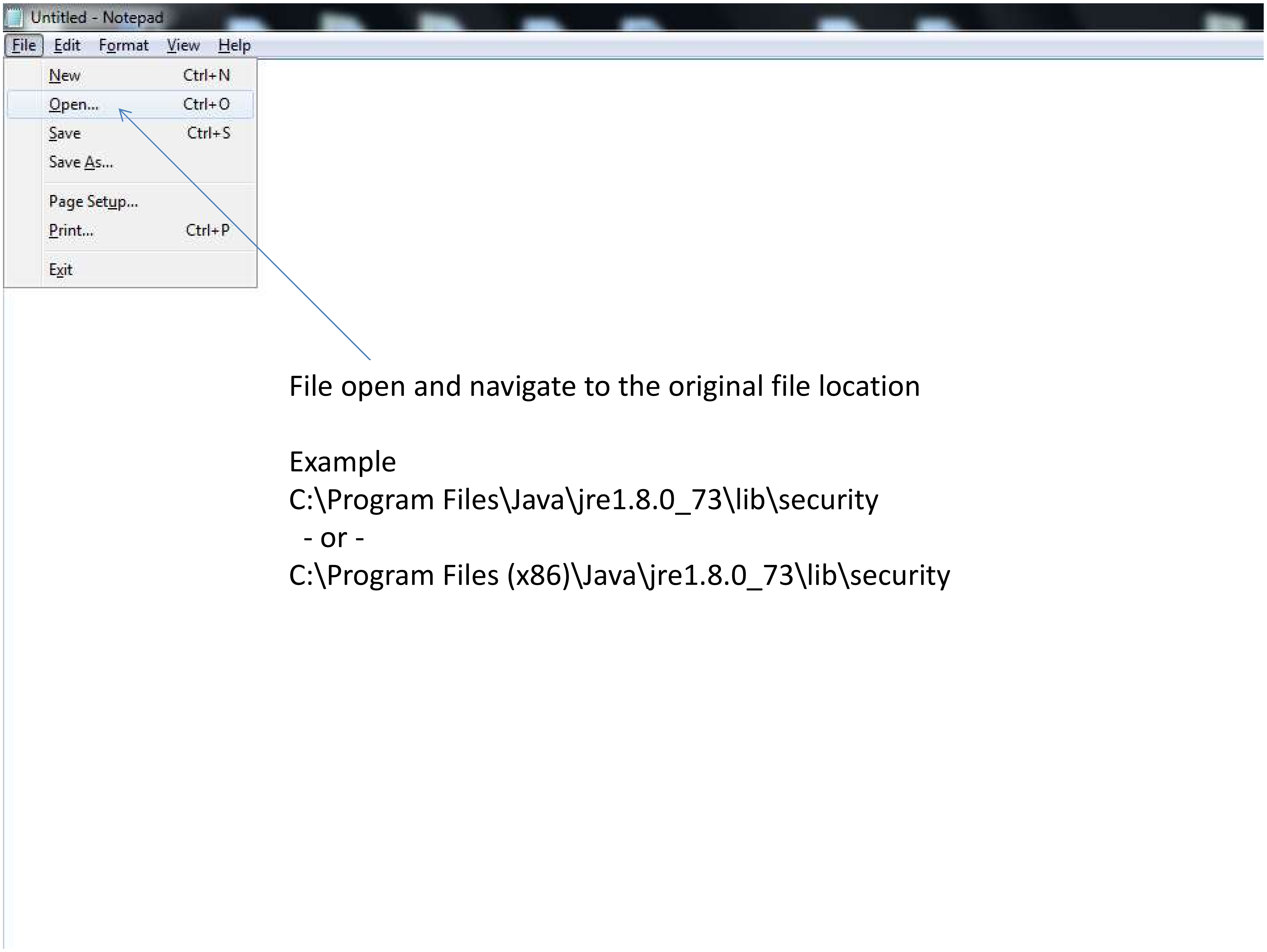
Example location:
C:\Program Files\Java\jre1.8.0_73\lib\security
 - Or -
C:\Program Files (x86)\Java\jre1.8.0_73\lib\security



Name	Date modified	Type	Size
blacklist	3/8/2016 2:45 PM	File	4 KB
blacklisted.certs	3/8/2016 2:45 PM	CERTS File	2 KB
cacerts	3/8/2016 2:45 PM	File	98 KB
java.policy	3/8/2016 2:45 PM	POLICY File	3 KB
java.security	3/8/2016 2:45 PM	SECURITY File	27 KB
javaws.policy	3/8/2016 2:45 PM	POLICY File	1 KB
local_policy.jar	3/8/2016 2:45 PM	Executable Jar File	4 KB
trusted.libraries	3/8/2016 2:45 PM	LIBRARIES File	0 KB
US_export_policy.jar	3/8/2016 2:45 PM	Executable Jar File	3 KB

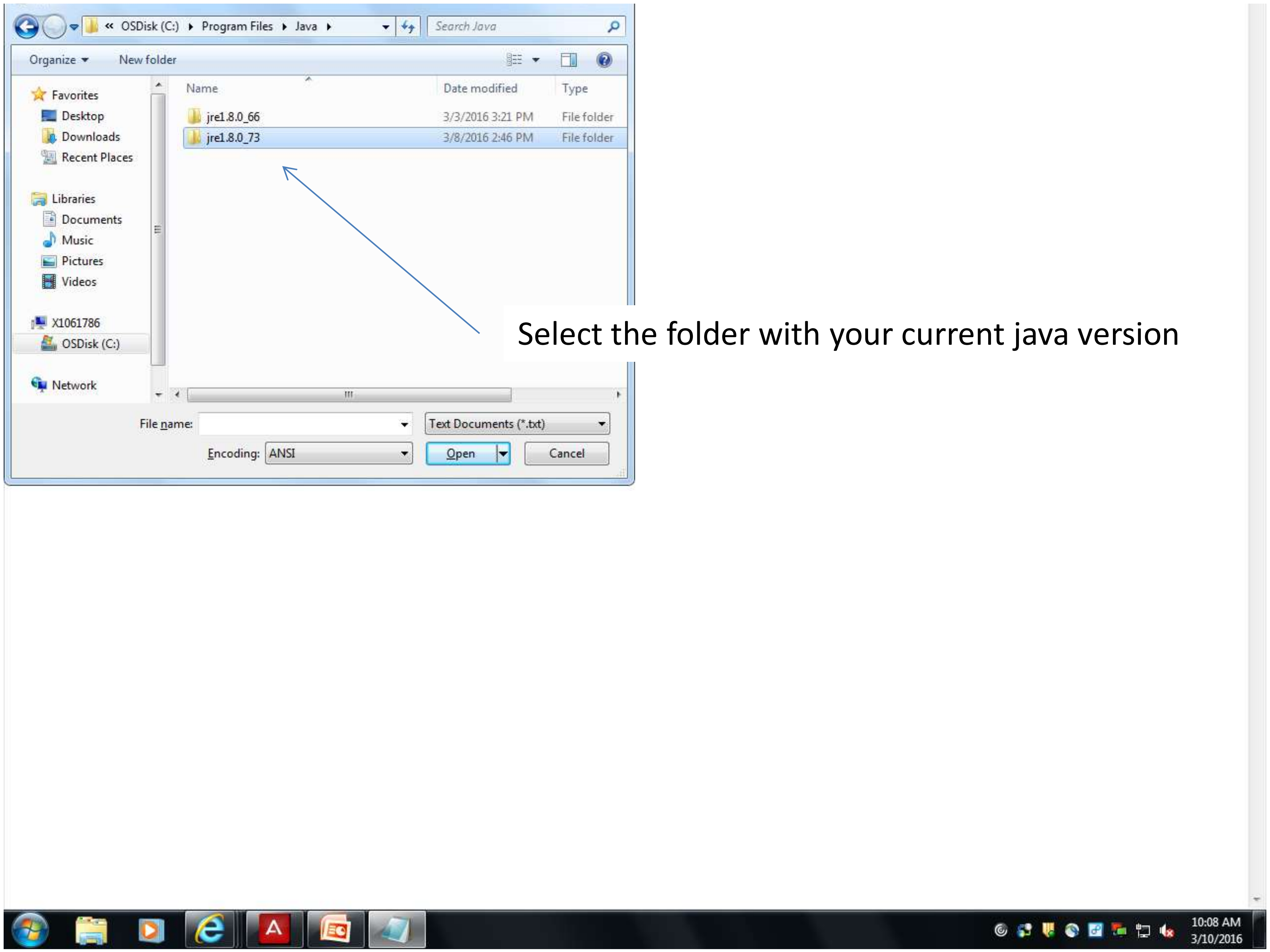


Edit the java .security file with “Administrator rights” using Notepad or Notepad++



File open and navigate to the original file location

Example
C:\Program Files\Java\jre1.8.0_73\lib\security
 - or -
C:\Program Files (x86)\Java\jre1.8.0_73\lib\security



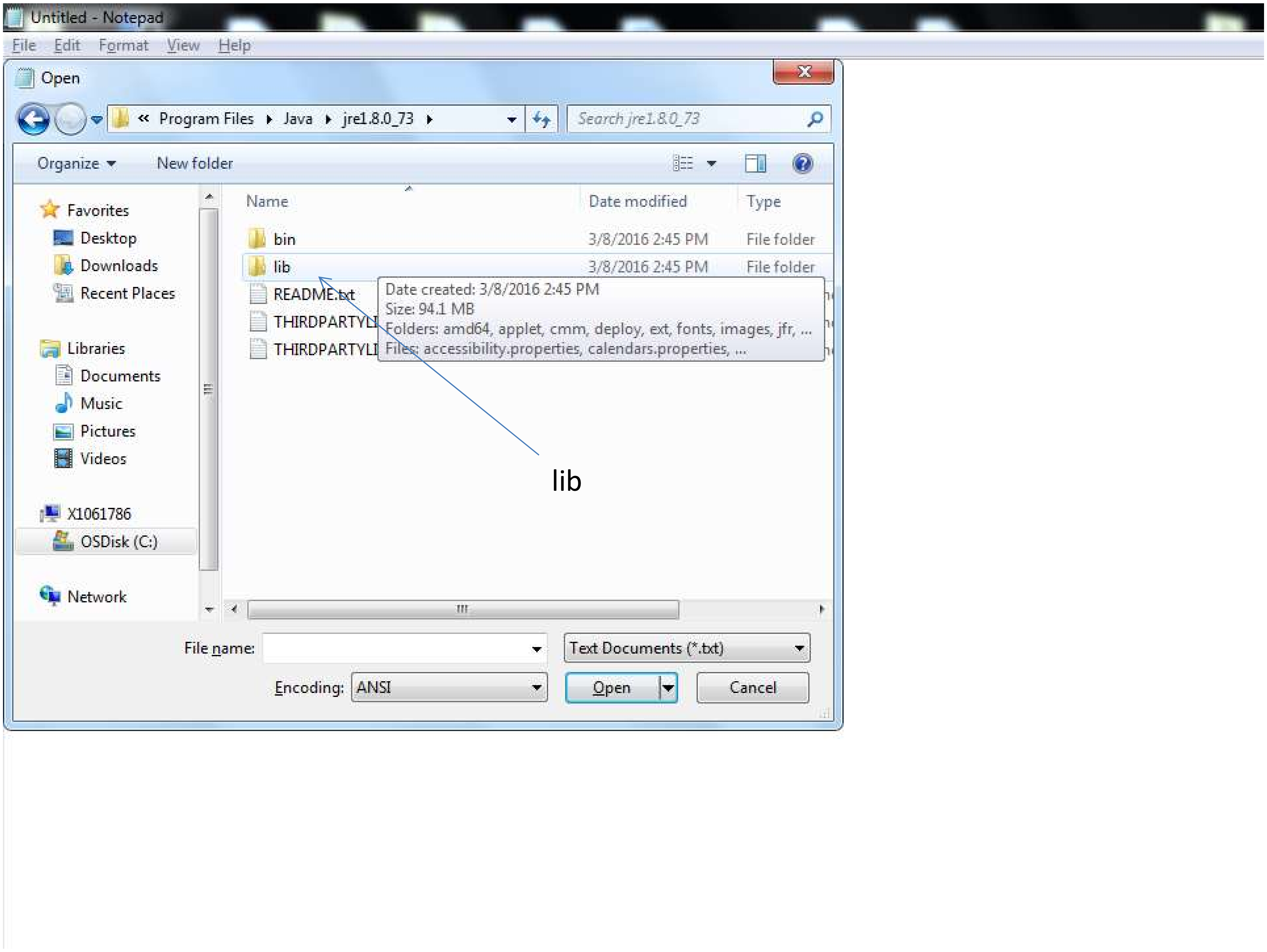
Name	Date modified	Type
jre1.8.0_66	3/3/2016 3:21 PM	File folder
jre1.8.0_73	3/8/2016 2:46 PM	File folder

File name:

Text Documents (*.txt)

Encoding: ANSI

Select the folder with your current java version



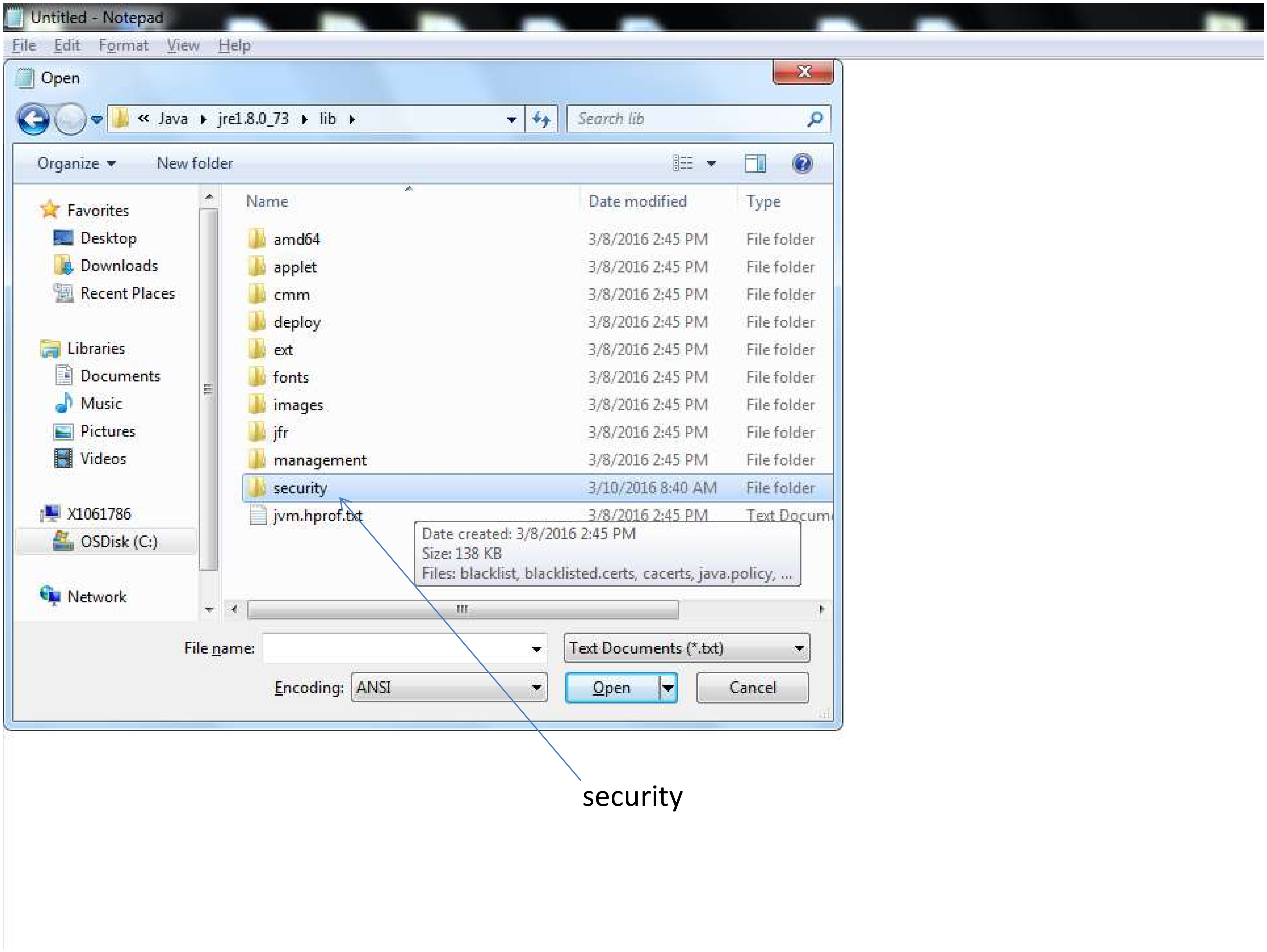
« Program Files ▸ Java ▸ jre1.8.0_73 ▸

Search jre1.8.0_73

Name	Date modified	Type
bin	3/8/2016 2:45 PM	File folder
lib	3/8/2016 2:45 PM	File folder
README.txt		
THIRDPARTYLI		
THIRDPARTYLI		

Date created: 3/8/2016 2:45 PM
Size: 94.1 MB
Folders: amd64, applet, cmm, deploy, ext, fonts, images, jfr, ...
Files: accessibility.properties, calendars.properties, ...

File name:
Text Documents (*.txt)
Encoding: ANSI

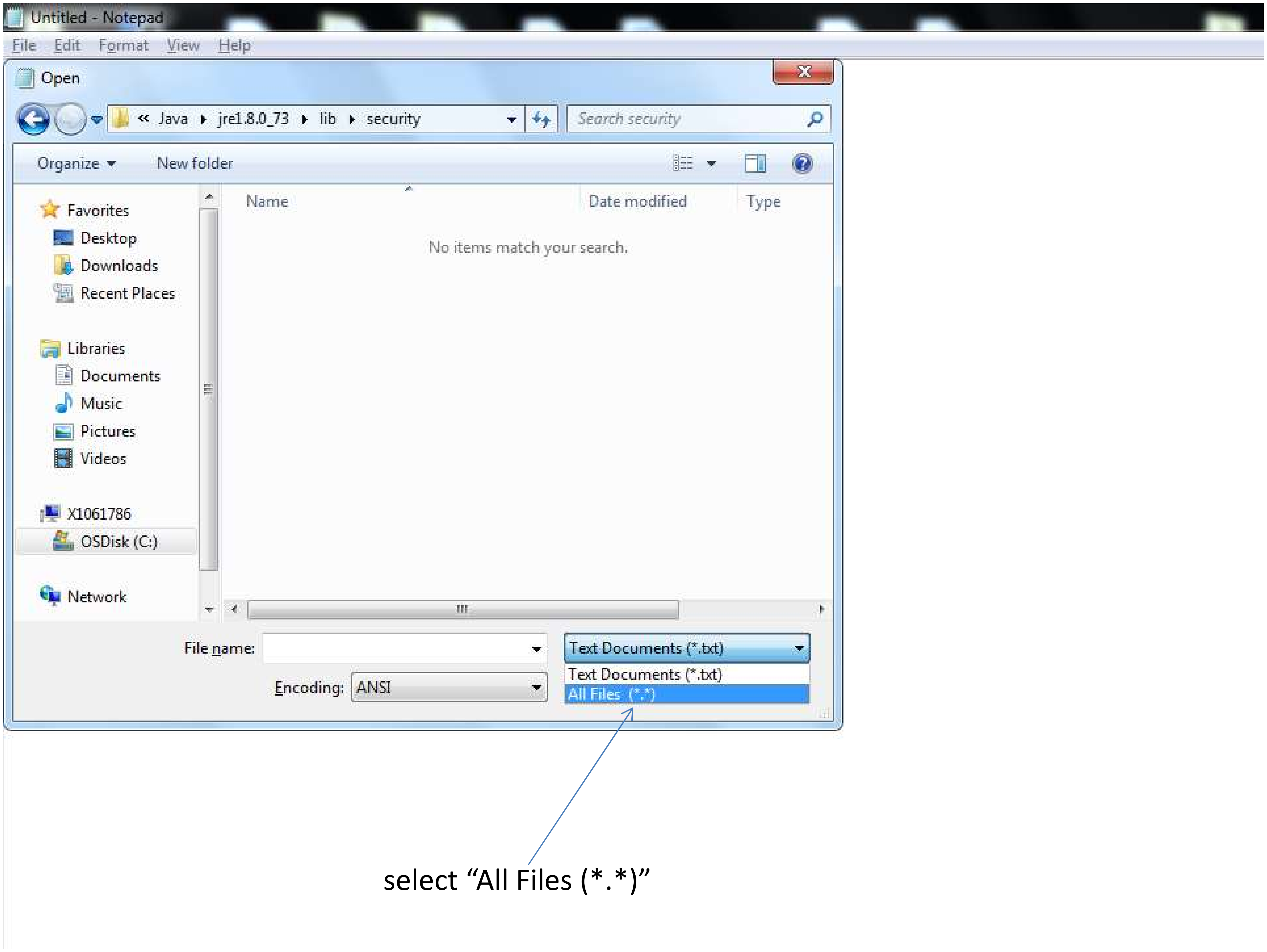


Name	Date modified	Type
amd64	3/8/2016 2:45 PM	File folder
applet	3/8/2016 2:45 PM	File folder
cmm	3/8/2016 2:45 PM	File folder
deploy	3/8/2016 2:45 PM	File folder
ext	3/8/2016 2:45 PM	File folder
fonts	3/8/2016 2:45 PM	File folder
images	3/8/2016 2:45 PM	File folder
jfr	3/8/2016 2:45 PM	File folder
management	3/8/2016 2:45 PM	File folder
security	3/10/2016 8:40 AM	File folder
jvm.hprof.txt	3/8/2016 2:45 PM	Text Docum

Date created: 3/8/2016 2:45 PM
Size: 138 KB
Files: blacklist, blacklisted.certs, cacerts, java.policy, ...

File name:
Text Documents (*.txt)
Encoding: ANSI

security



No items match your search.

File name:

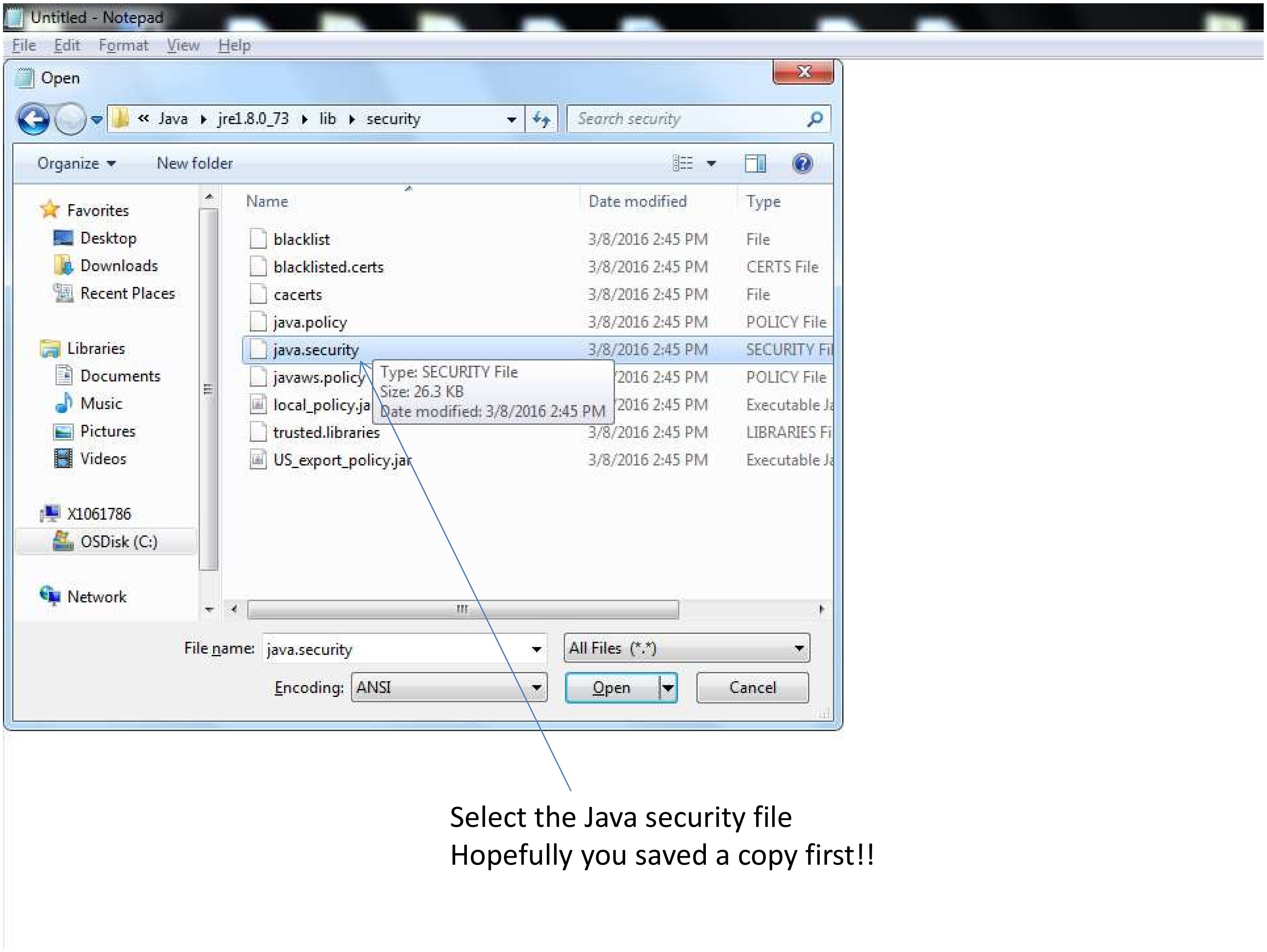
Encoding: ANSI

Text Documents (*.txt)

Text Documents (*.txt)

All Files (*.*)

select “All Files (*.*)”



Untitled - Notepad

Open

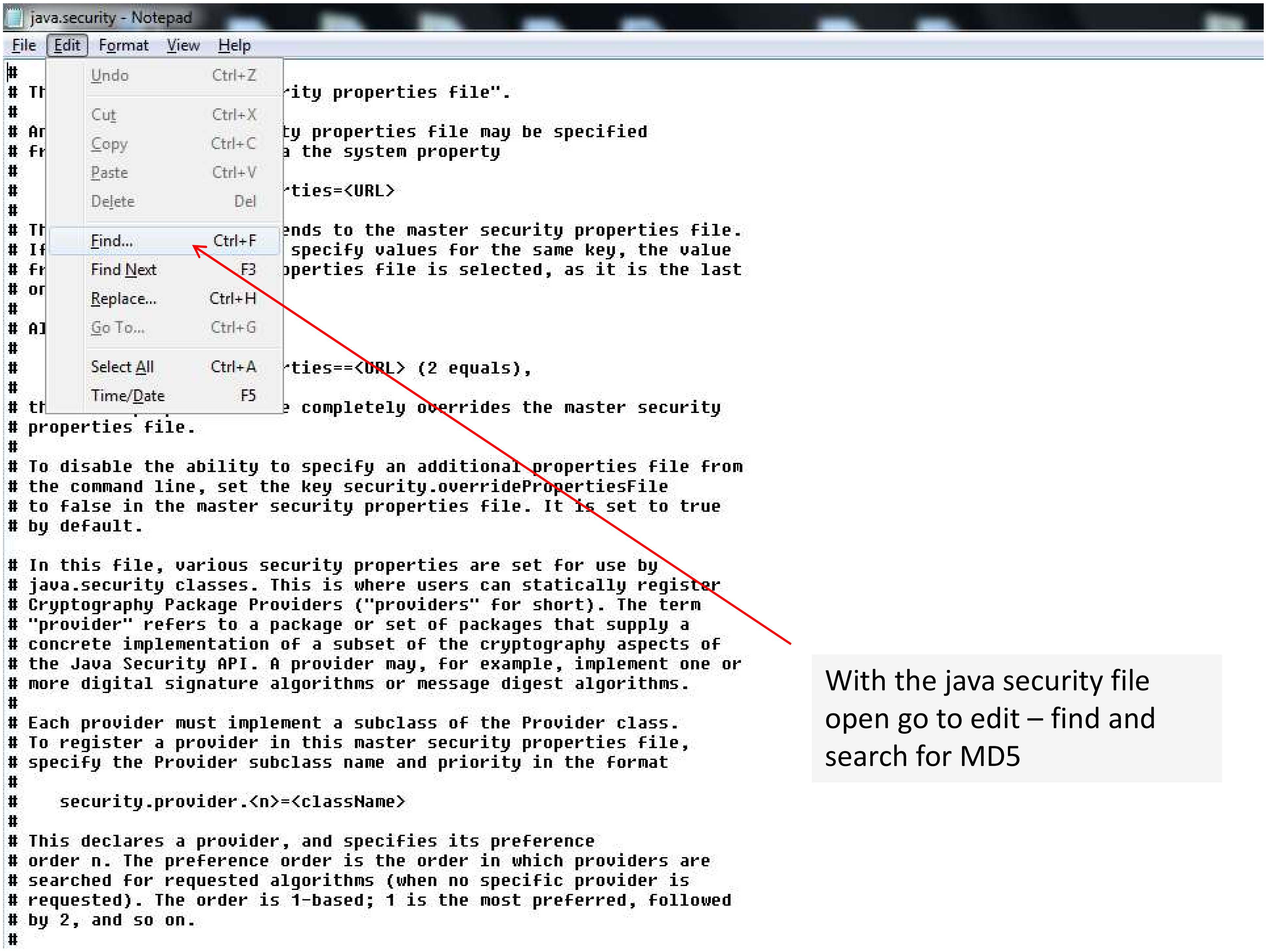
« Java ▸ jre1.8.0_73 ▸ lib ▸ security

Name	Date modified	Type
blacklist	3/8/2016 2:45 PM	File
blacklisted.certs	3/8/2016 2:45 PM	CERTS File
cacerts	3/8/2016 2:45 PM	File
java.policy	3/8/2016 2:45 PM	POLICY File
java.security	3/8/2016 2:45 PM	SECURITY Fi
javaws.policy	/2016 2:45 PM	POLICY File
local_policy.ja	/2016 2:45 PM	Executable Ja
trusted.libraries	3/8/2016 2:45 PM	LIBRARIES Fi
US_export_policy.jar	3/8/2016 2:45 PM	Executable Ja

Type: SECURITY File
Size: 26.3 KB
Date modified: 3/8/2016 2:45 PM

File name: java.security
All Files (*.*)
Encoding: ANSI

Select the Java security file
Hopefully you saved a copy first!!



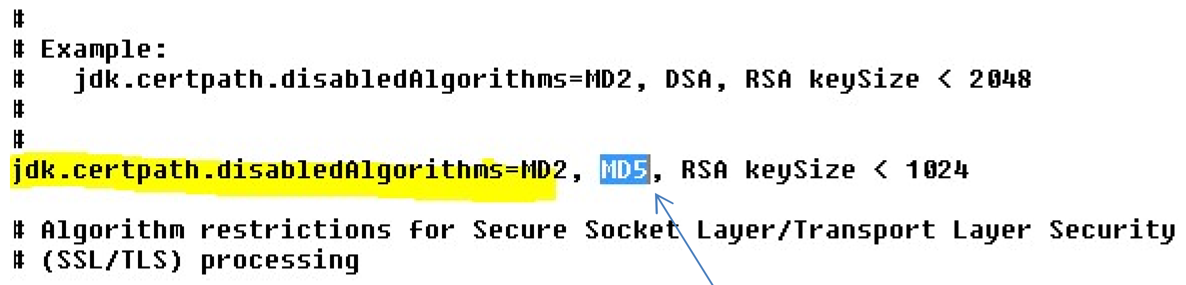
```
#
# Th                    rity properties file".
#
# An                    ty properties file may be specified
# fr                    a the system property
#
#                       rties=<URL>
#
# Th                    ends to the master security properties file.
# If                     specify values for the same key, the value
# fr                    operties file is selected, as it is the last
# or
#
# Al
#
#                       rties==<URL> (2 equals),
#
# th                    e completely overrides the master security
# properties file.
#
# To disable the ability to specify an additional properties file from
# the command line, set the key security.overridePropertiesFile
# to false in the master security properties file. It is set to true
# by default.

# In this file, various security properties are set for use by
# java.security classes. This is where users can statically register
# Cryptography Package Providers ("providers" for short). The term
# "provider" refers to a package or set of packages that supply a
# concrete implementation of a subset of the cryptography aspects of
# the Java Security API. A provider may, for example, implement one or
# more digital signature algorithms or message digest algorithms.
#
# Each provider must implement a subclass of the Provider class.
# To register a provider in this master security properties file,
# specify the Provider subclass name and priority in the format
#
#    security.provider.<n>=<className>
#
# This declares a provider, and specifies its preference
# order n. The preference order is the order in which providers are
# searched for requested algorithms (when no specific provider is
# requested). The order is 1-based; 1 is the most preferred, followed
# by 2, and so on.
#
```

With the java security file open go to edit – find and search for MD5

Search will show the first MD5 option and remove MD5

You only need to remove MD5 from lines that don't have a # in front of Them and there are two

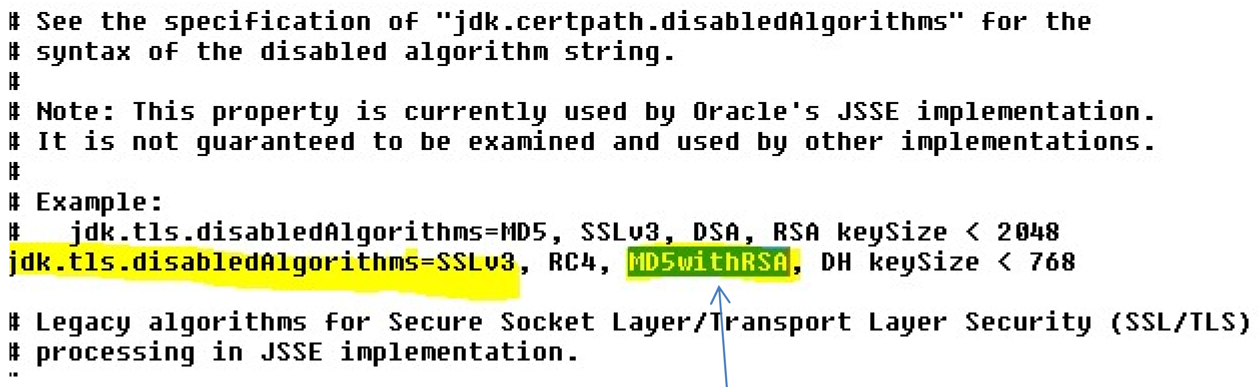


```
#
# Example:
#   jdk.certpath.disabledAlgorithms=MD2, DSA, RSA keySize < 2048
#
#
jdk.certpath.disabledAlgorithms=MD2, MD5, RSA keySize < 1024

# Algorithm restrictions for Secure Socket Layer/Transport Layer Security
# (SSL/TLS) processing
```

Delete: Md5,

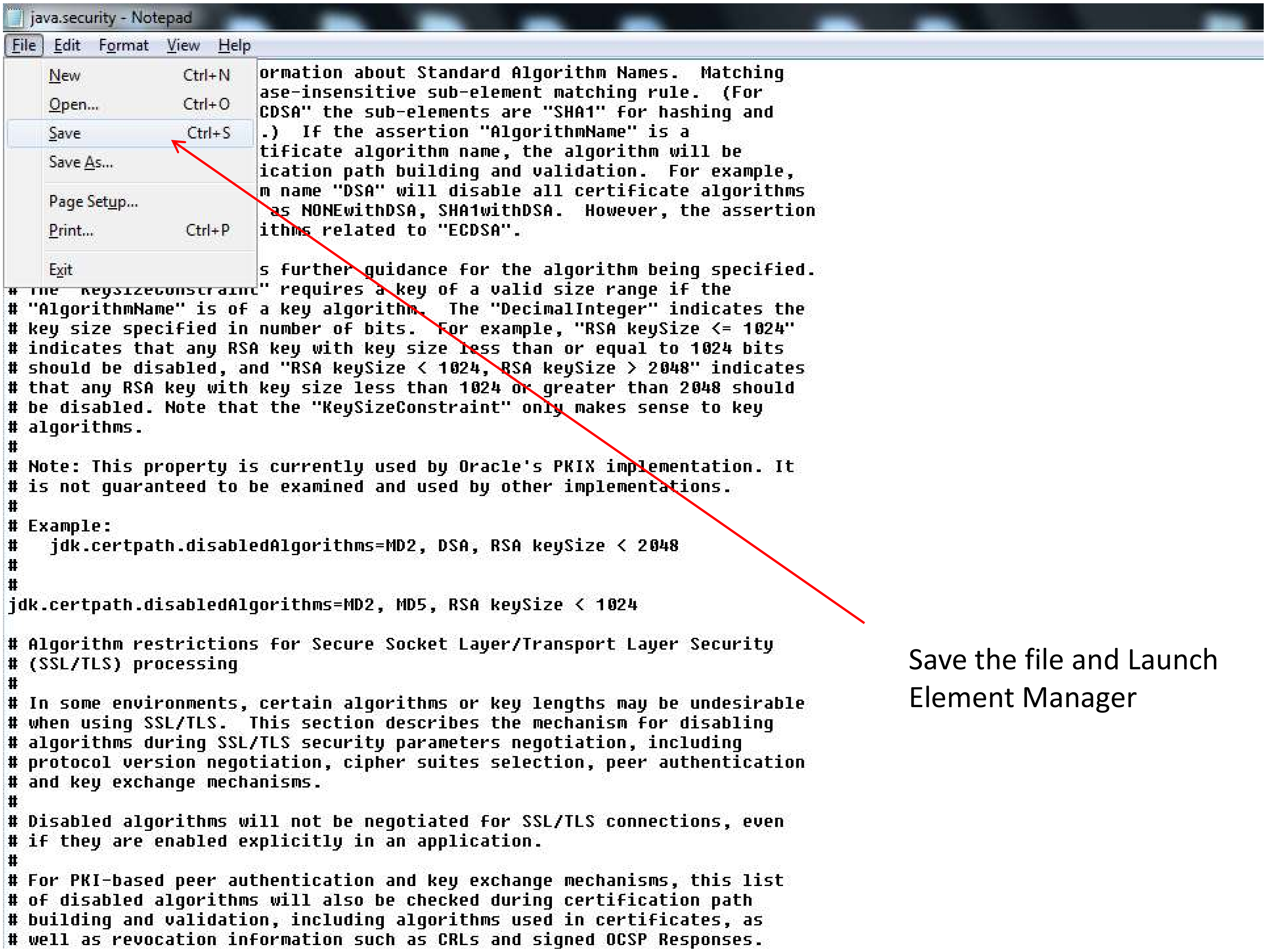
Remove MD5withRSA



```
# See the specification of "jdk.certpath.disabledAlgorithms" for the
# syntax of the disabled algorithm string.
#
# Note: This property is currently used by Oracle's JSSE implementation.
# It is not guaranteed to be examined and used by other implementations.
#
# Example:
#   jdk.tls.disabledAlgorithms=MD5, SSLv3, DSA, RSA keySize < 2048
jdk.tls.disabledAlgorithms=SSLv3, RC4, MD5withRSA, DH keySize < 768

# Legacy algorithms for Secure Socket Layer/Transport Layer Security (SSL/TLS)
# processing in JSSE implementation.
```

Delete MD5withRSA,



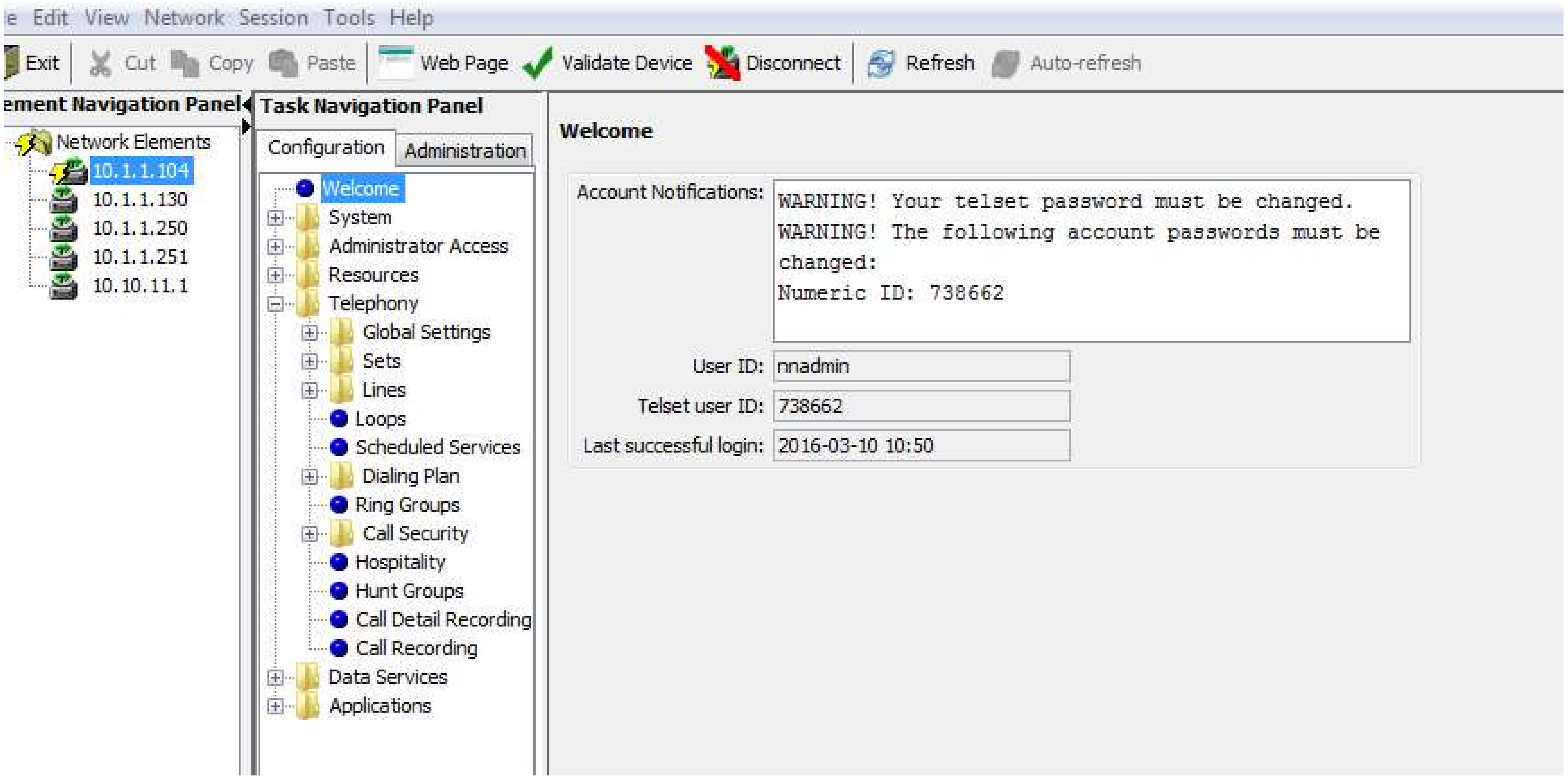
```
ormation about Standard Algorithm Names.  Matching
ase-insensitive sub-element matching rule.  (For
CDSA" the sub-elements are "SHA1" for hashing and
.)  If the assertion "AlgorithmName" is a
tificate algorithm name, the algorithm will be
ication path building and validation.  For example,
m name "DSA" will disable all certificate algorithms
 as NONEwithDSA, SHA1withDSA.  However, the assertion
ithms related to "ECDSA".

s further guidance for the algorithm being specified.
# The "KeySizeConstraint" requires a key of a valid size range if the
# "AlgorithmName" is of a key algorithm.  The "DecimalInteger" indicates the
# key size specified in number of bits.  For example, "RSA keySize <= 1024"
# indicates that any RSA key with key size less than or equal to 1024 bits
# should be disabled, and "RSA keySize < 1024, RSA keySize > 2048" indicates
# that any RSA key with key size less than 1024 or greater than 2048 should
# be disabled. Note that the "KeySizeConstraint" only makes sense to key
# algorithms.
#
# Note: This property is currently used by Oracle's PKIX implementation. It
# is not guaranteed to be examined and used by other implementations.
#
# Example:
#   jdk.certpath.disabledAlgorithms=MD2, DSA, RSA keySize < 2048
#
#
jdk.certpath.disabledAlgorithms=MD2, MD5, RSA keySize < 1024

# Algorithm restrictions for Secure Socket Layer/Transport Layer Security
# (SSL/TLS) processing
#
# In some environments, certain algorithms or key lengths may be undesirable
# when using SSL/TLS.  This section describes the mechanism for disabling
# algorithms during SSL/TLS security parameters negotiation, including
# protocol version negotiation, cipher suites selection, peer authentication
# and key exchange mechanisms.
#
# Disabled algorithms will not be negotiated for SSL/TLS connections, even
# if they are enabled explicitly in an application.
#
# For PKI-based peer authentication and key exchange mechanisms, this list
# of disabled algorithms will also be checked during certification path
# building and validation, including algorithms used in certificates, as
# well as revocation information such as CRLs and signed OCSP Responses.
```

Save the file and Launch Element Manager

Launch Element Manager and connect.



Welcome

Account Notifications:

```
WARNING! Your telset password must be changed.
WARNING! The following account passwords must be
changed:
Numeric ID: 738662
```

User ID:	nnadmin
Telset user ID:	738662
Last successful login:	2016-03-10 10:50