

COMP 5350 / 6350

Digital Forensics

Windows Registry Analysis – Shellbags
Regular Expressions



Windows Registry Analysis – Shellbags

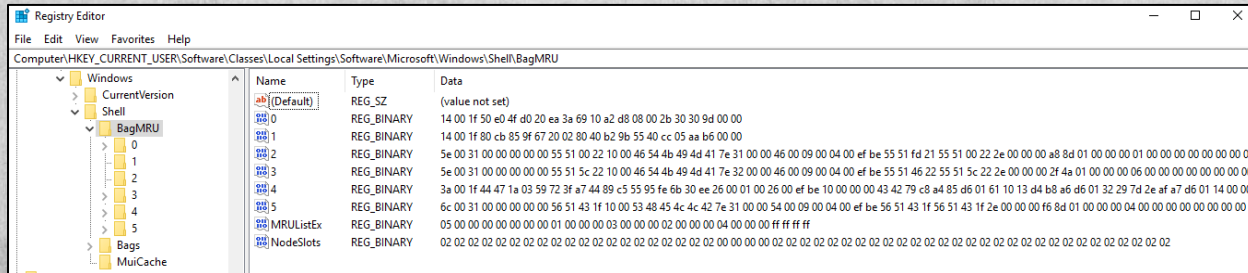
Shellbag Forensic Utility

- Shellbags can highlight numerous forensically valuable pieces of information about users:
 - ✓ Directory usage
 - Local, External, and Networked
 - ✓ Provides date and time changes within a directory
 - ✓ Shellbag registry keys retain historical data even after
 - Directory deletion
 - Unmounting of external and network directories

ShellBag Keys

- The ShellBag information is composed of two main registry keys:
 - ✓ BagMRU
 - Stores folder names and records folder paths
 - Represents the user desktop
 - ✓ Bags
 - Stored view preferences including window size, location, and view mode

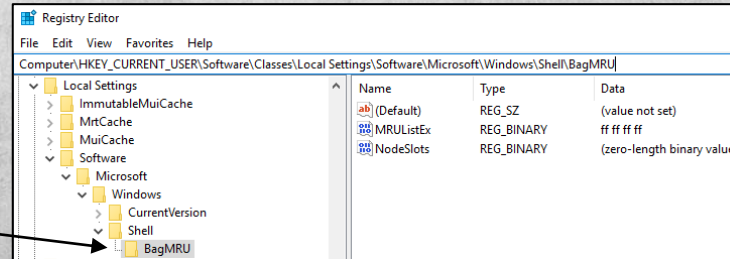
Computer\HKEY_CURRENT_USER\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\BagMRU



ShellBag Artifacts

- A clean installation of Windows provides a starting point on how changes modify ShellBag contents
- Initial Windows installation process results in a single BagsMRU key
 - ✓ MRUListEx: Empty Most Recently Used List
 - ✓ NodeSlots: Empty
- No Bags registry key will be created under the BagMRU key until a user makes modifications to preferences

User Desktop



GUID Mapping


- When new folders are generated on a Windows system, BagMRU and Bags keys add Globally Unique Identifiers (GUIDs) which correlate to different types of objects, in this case directory types
 - ✓ Downloads – 885A186E-A440-4ADA-812B-DB871B942259
 - ✓ Generic – 5C4F28B5-F869-4E84-8E60-F11DB97C5CC7
 - ✓ Documents – 7D49D726-3C21-4F05-99AA-FDC2C9474656
 - ✓ Music – 94D6DDCC-4A68-4175-A374-BD584A510B78
 - ✓ Pictures – B3690E58-E961-423B-B687-386EBFD83239
 - ✓ Video – 5FA96407-7E77-483C-AC93-691D05850DE8
 - ✓ Contacts – DE2B70EC-9BF7-4A93-BD3D-243F7881D492
 - ✓ Home Folder – 24CCB8A6-C45A-477D-B940-3382B9225668
 - ✓ User Libraries – C4D98F09-6124-4FE0-9942-826416082DA9
 - ✓ Programs – D674391B-52D9-4E07-834E-67C98610F39D
 - ✓ User Files – CD0FC69B-71E2-46E5-9690-5BCD9F57AAB3
 - ✓ Searches – 0B0BA2E3-405F-415E-A6EE-CAD625207853

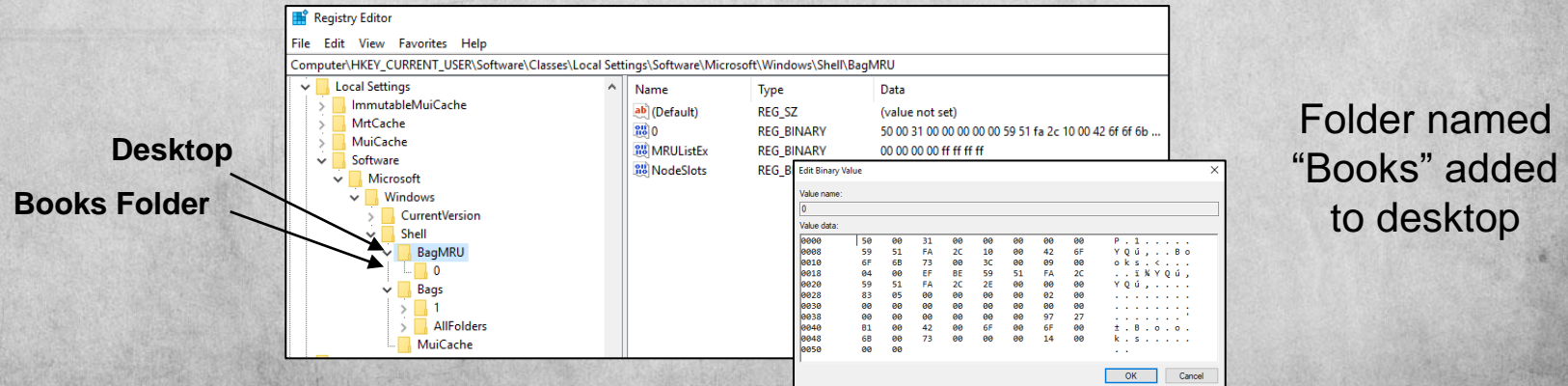
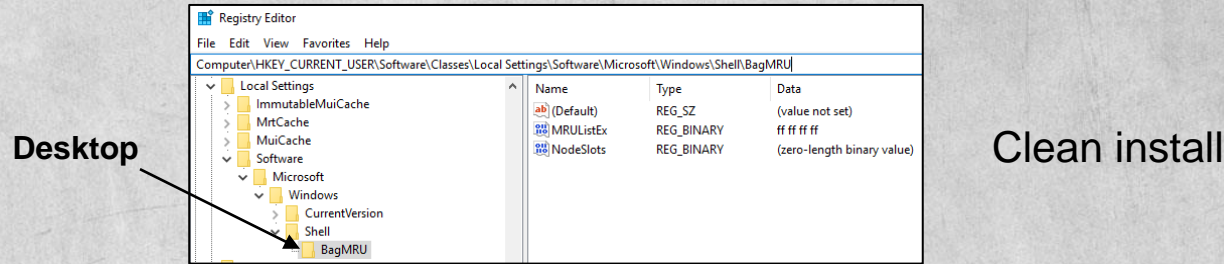
Scenarios

- Different scenarios highlight Shellbag changes within the registry:
 - ✓ Folder Access
 - User Desktop Folder Access
 - Folder Access Across Different Paths
 - Attached Media Folder Access
 - Directory Traversal
 - ✓ Folder Deletion
 - User Desktop Folder Deletion
 - Folder Deletion Across Different Paths
 - Removal of Attached Media
- It is important to recognize that several registry keys must be used in concert to provide accurate information about a users activities

Folder Access

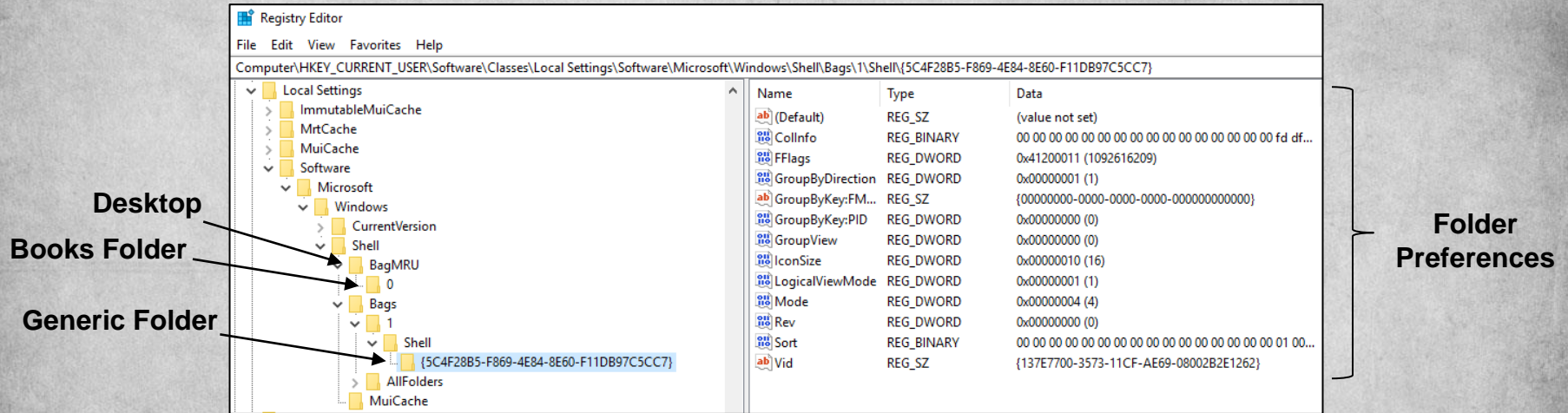
Folder Access on User Desktop

- The first scenario is modification of desktop folder and folder contents
 - Note that any values under the BagsMRU key are accesses in the user desktop
- 
- A screenshot of the Windows Registry Editor. The left pane shows the tree structure expanded to Computer\HKEY_CURRENT_USER\Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\Explorer\BagMRU. The right pane shows a single value named 'BagMRU' with a data type of 'REG_SZ' and a value of 'C:\Users\user\Desktop\'.



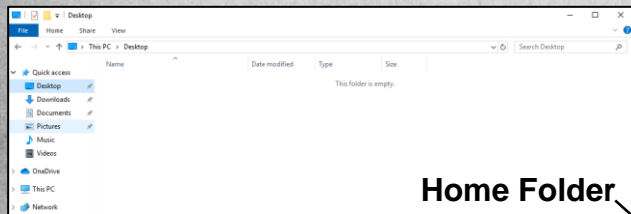
Folder Access on User Desktop

- Expansion of the Bags key shows information relating to the type of folder created (i.e. Generic) and folder configuration settings

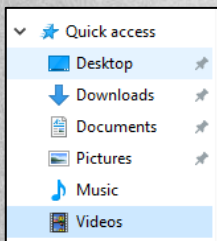


Folder Access Across Different Paths

User View

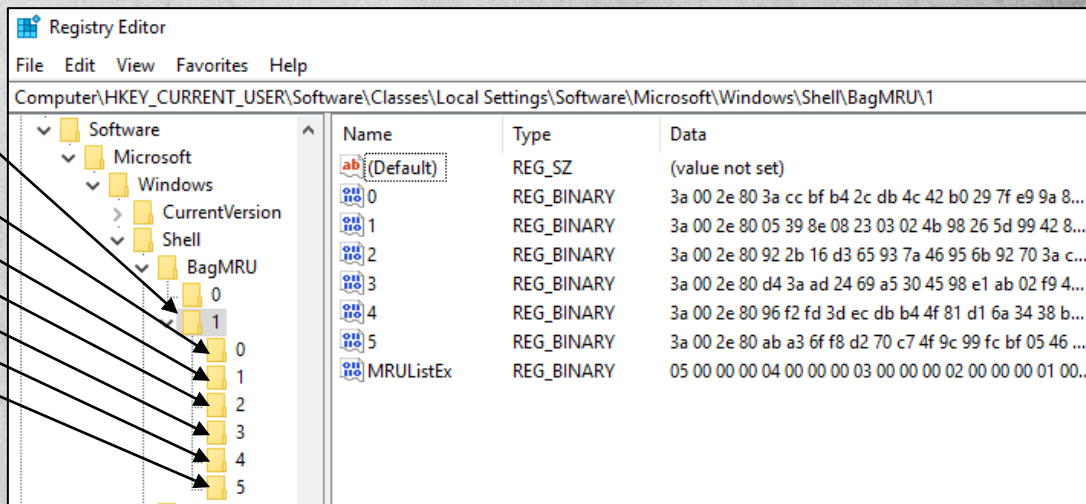


Home Folder



Order of Access

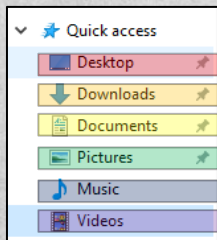
- 1) File Explorer
- 2) Desktop
- 3) Downloads
- 4) Pictures
- 5) Music
- 6) Videos



Folder Access Across Different Paths

Order of Directory Access

- 1) File Explorer
- 2) Desktop
- 3) Downloads
- 4) Pictures
- 5) Music
- 6) Videos



User View

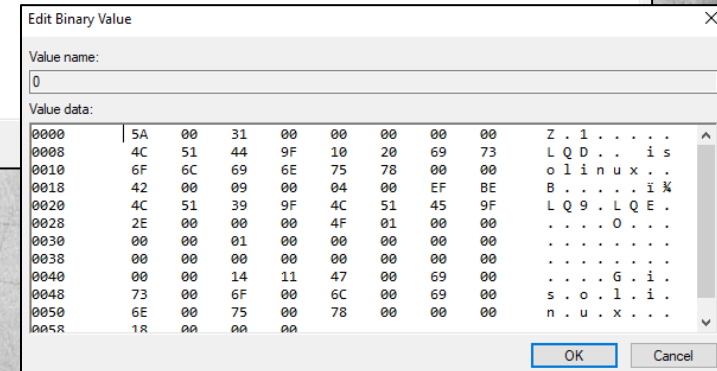
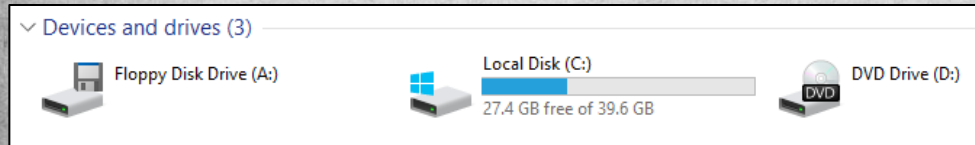
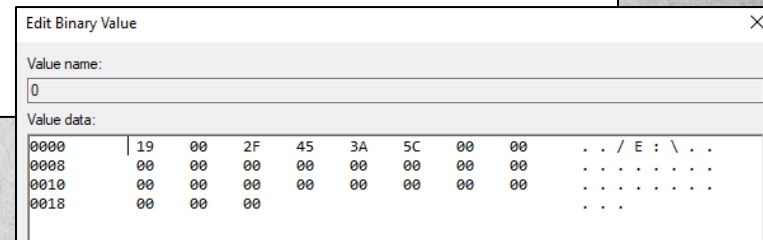
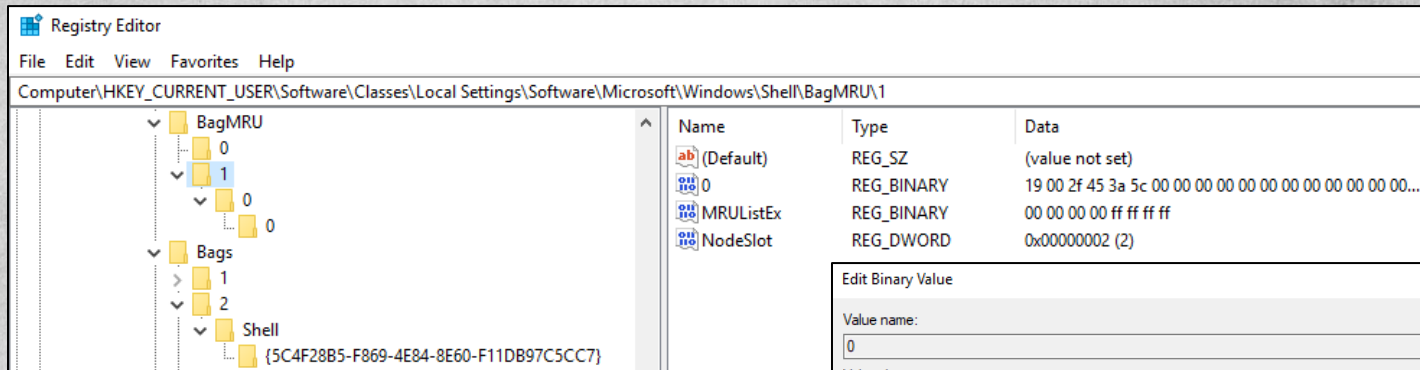
The Registry Editor window shows the path `Computer\HKEY_CURRENT_USER\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\BagMRU\0`. The left pane displays a tree structure of the registry, and the right pane shows the values for the selected key.

Name	Type	Data
(Default)	REG_SZ	(value not set)
MRUListEx	REG_BINARY	ff ff ff ff
NodeSlot	REG_DWORD	0x00000001 (1)

Arrows point from the registry tree to the corresponding folder names on the right:

- Home Folder
- Generic
- Generic
- Documents
- Pictures
- Music
- Video

Attached Media Folder Access



- Comparison of existing and previously attached external media

Directory Traversal

Edit Binary Value

Value name:
0

Value data:

0000	19	00	2F	43	3A	5C	00	00	.. / C : \ . .
0008	00	00	00	00	00	00	00	00
0010	00	00	00	00	00	00	00	00
0018	00	00	00					

Edit Binary Value

Value name:
0

Value data:

0000	56	00	31	00	00	00	00	00	V . 1
0008	59	51	D8	03	10	00	57	69	Y Q 0 . . W i
0010	6E	64	6F	77	73	00	40	00	n d o w s . @ .
0018	09	00	04	00	EF	BE	2F	4D	. . . i % / M

Edit Binary Value

Value name:
0

Value data:

0000	4E	00	31	00	00	00	00	00	N . 1
0008	2F	4D	3A	3C	10	00	42	6F	/ M : < . . B o
0010	6F	74	00	00	3A	00	09	00	o t
0018	04	00	EF	BE	2F	4D	3A	3C	. . i % / M : <
0020	2F	4D	3A	3C	2E	00	00	00	/ M : < . . .

Edit Binary Value

Value name:
0

Value data:

0000	4A	00	31	00	00	00	00	00	J . 1
0008	2F	4D	3A	3C	10	00	45	46	/ M : < . . E F
0010	49	00	38	00	09	00	04	00	I . 8
0018	EF	BE	2F	4D	3A	3C	59	51	i % / M : < Y Q
0020	38	1B	2E	00	00	00	99	08	;
0028	00	00	00	00	01	00	00	00
0030	00	00	00	00	00	00	00	00
0038	00	00	00	00	F1	CB	E6	00 h E .
0040	45	00	46	00	49	00	00	00	E . F . I . .
0048	12	00	00	00				

OK Cancel

C:\Windows\Boot\EFI

Registry Editor

File Edit View Favorites Help

Computer\HKEY_CURRENT_USER\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\BagMRU\3\0

BagMRU

- 0
- 1
- 2
- 3
- 0
- 0

Bags

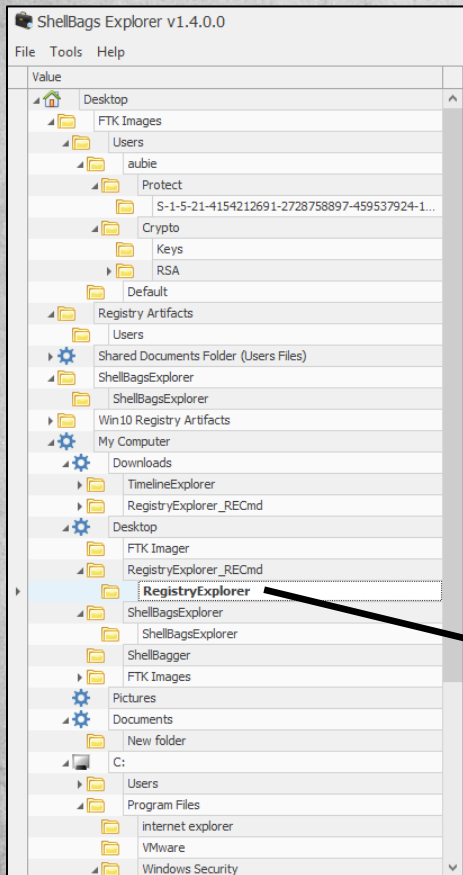
- 1
- 2
- 3
- 4
- 5

Shell

(SC4F28B5-F869-4E84-8E60-F11DB97C5CCT7)

Name	Type	Data
(Default)	REG_SZ	(value not set)
0	REG_BINARY	56 00 31 00 00 00 00 00 59 51 d8 03 10 00 57 69 6e 6...
MRUListEx	REG_BINARY	00 00 00 00 ff ff ff ff
NodeSlot	REG_DWORD	0x00000005 (5)

Shellbags Explorer



- Although registry editor provides detailed key and value information, it fails to consolidate information into a comprehensive view
- Shellbags Explorer provides a graphical view of which directories were accessed, when they were accessed, and the registry keys that were generated

Summary	Details	Hex
Name: Registry Explorer		
Absolute path: Desktop\My Computer\Desktop\RegistryExplorer_RECmd\RegistryExplorer		
Key-Value name path: BagMRU\0\1\4-0		
Target timestamps		
Created on: 2020-10-22 23:28:42.000		
Modified on: 2020-10-22 23:28:54.000		
Last accessed on: 2020-10-22 23:28:54.000		
Miscellaneous		
Shell type: Directory		
Node slot: 60		
MRU position: 0		
# of child bags: 0		

Content Searches

Content Search Test Data

- When learning about content searches, it is helpful to use large data sets to understand search patterns
 - ✓ Enron email data set
 - An email dataset containing over 500,000 emails generated by employees of the Enron Corporation which was obtained by the Federal Energy Regulatory Commission during its investigation of Enron's collapse

```
sansforensics@siftworkstation: ~/RegEx
$ ls -la
total 2213180
drwxr-xr-x  3 sansforensics sansforensics      4096 Oct 28 14:50 .
drwxr-xr-x 22 sansforensics sansforensics      4096 Oct 27 22:27 ..
-rw-rw-r--  1 sansforensics sansforensics 1823016960 Oct 26 22:05 enron_mail_20150507.tar
-rw-rw-r--  1 sansforensics sansforensics  443254787 Oct 28 14:37 enron_mail_20150507.tar.gz
drwxr-xr-x 152 sansforensics sansforensics      4096 Apr  1  2011 maildir
```


Content Searches

- Once digital artifacts are collected and processed, there are several methods of conducting forensics analysis
 - ✓ Signature Match
 - ✓ Event Counts
 - ✓ Pattern Match
- Some common commands for content searches include:
 - ✓ strings
 - Display printable strings in files
 - ✓ awk
 - Shell scripting language to manipulate and filter data
 - ✓ sed
 - Stream editor used to manipulate and filter text from files or pipelines
 - ✓ grep
 - Global Regular Expression Printer

strings

- For each file given, GNU strings prints character sequences of at least 4 characters long
- This is a very helpful application when evaluating sting contents in firmware, but can be equally helpful with extracting forensic information
- Options that can help tailor ASCII string search
 - ✓ -n
 - Specify string length
 - ✓ -t <d,o,x>
 - Print offset
 - ✓ -a
 - Scan all file sections

awk

- Awk, named after the inventors Aho, Weinberger, and Kernighan, is a scripting language used to manipulate data and generate reports that provides programming structures including variables, numeric functions, string functions, and logical operators
- Awk is commonly used for pattern scanning and processing and can search one or more files to identify pattern matches
- The most common awk functions include:
 - ✓ Scanning files line by line
 - ✓ Splitting input line into fields
 - ✓ Comparing input fields to a pattern
 - ✓ Performing actions on a matched line

sed

- SED stands for stream editor and performs numerous functions on input data:
 - ✓ Search
 - ✓ Find and Replace
 - ✓ Insert
 - ✓ Delete
- Files do not have to be opened for SED to edit them which makes finding and replacing content more efficient

```
The quick brown fox jumps over the lazy dog  
The quick brown cat jumps over the lazy dog
```

Original Text

```
$ sed 's/dog/pig/' sedTestFile  
The quick brown fox jumps over the lazy pig  
The quick brown cat jumps over the lazy pig
```

- How to read the command: Use sed to substitute the word dog with the word pig in file sedTestFile
 - ✓ Only output changes, the original file is unchanged

grep

- g/re/p, which stands for globally search for a regular expression and print matching lines, is a utility that provides pattern matching of plain-text data sets
 - ✓ Uses a Non-Deterministic Finite Automaton
- As with any Linux command, grep can be scripted to provide analysis across numerous directories and files
- An improvement to grep, known as egrep provided a broader set of search patterns, but is generally slower than traditional grep
 - ✓ Uses a Deterministic Finite Automaton

grep Options

- The following grep options provide flexibility during pattern matches:

✓ -A #

- Display lines after a match

✓ -B #

- Display lines before a match

✓ -C #

- Display lines before and after a match

✓ -c

- Count number of matches

✓ -i

- Ignore case

✓ -l

- List filename match

✓ -r

- Recursive match

✓ -v

- Do not match

✓ -w

- Match whole words

✓ -x

- Exact match

grep Patterns

- Although options specify pattern usage, pattern development is a more essential skillset for forensic analysis
 - ✓ Groups and Ranges
 - ✓ Character Classes
 - ✓ Quantifiers
 - ✓ Anchors
 - ✓ Assertions
 - ✓ Special Characters
 - ✓ String Replacement
 - ✓ Metacharacters
 - ✓ Pattern Modifiers

Groups and Ranges

- Groups are used to test potential matches in a compact form
- Capturing vs. Non-Capturing Groups
 - ✓ Capturing groups use parentheses to capture text matched by the regular expression and reuse it for future matches with a backreference
 - ✓ Non-capturing groups use parentheses to group the regular expression, but without capturing anything
- Ranges expand upon groups by allowing a broader set of matches

Symbol	Description
.	Any character except new line (\n)
(a b)	a or b
(...)	Group
(?:...)	Passive (non-capturing) group
[abc]	Range (a or b or c)
[^abc]	Not (a or b or c)
[a-z]	Lowercase from a to z
[A-Z]	Uppercase from A to Z
[0-9]	Digit from 0 to 9

Character Classes

- Character classes fall into two categories
 - ✓ Standard Expression
 - ✓ Bracket Expression (i.e. POSIX)

Standard
Expression

Expression	Description
\c	Control Character
\s	White Space
\S	Not White Space
\d	Digit
\D	Not Digit
\w	Word
\W	Not Word
\xhh	Hex Character

Bracket
Expression

Expression	Description
[:upper:]	Uppercase Letters
[:lower:]	Lowercase Letters
[:alpha:]	All letters
[:alnum:]	Digits and Letters
[:digit:]	Digits
[:xdigit:]	Hexadecimal Digits
[:punct:]	Punctuation
[:blank:]	Space and Tab
[:space:]	Blank Characters
[:word:]	Digits, letters and underscore

Quantifiers

- Quantifiers give the ability to match based on the number of items
- Greedy vs. Non-greedy
 - ✓ Greedy matches will match as many repetitions of the quantified pattern as possible
 - ✓ Non-greedy matches will try to match as few repetitions of the quantified pattern as possible

Symbol	Description
*	0 or more (Greedy)
*?	0 or more (Non-greedy)
+	1 or more (Greedy)
+?	1 or more (Non-greedy)
?	0 or 1 (Greedy)
??	0 or 1 (Non-greedy)
{3}	Exactly 3 matches
{3,}	3 or more matches
{3,5}	3, 4, or 5 matches
{3,5}?	3, 4, or 5 matches, ungreedy

Anchors

- Unless specified, regular expressions are run across all objects
- The purpose of anchors is to specify where a pattern should be applied to
- The anchor options available are:
 - ✓ Start
 - Word, String, or Line
 - ✓ End
 - Word, String, or Line

Symbol	Description
^	Start of string or line
\A	Start of string
\$	End of string or line
\Z	End of string
\b	Word boundary
\B	Not word boundary
\<	Start of word
\>	End of word

Pattern Examples

- Using groups and ranges, character classes, quantifiers, and anchors develop regular expression patterns for:

- ✓ Addresses

```
grep -Eor '[0-9]+\s+\w+\s+(?:Avenue|Lane|Road|Boulevard|Drive|Street)' <Directory>
```

- How to read this regular expression: “A general regular expression that recursively searches a directory only for one or more numbers followed by a space, word, space, and the word Avenue, Lane, Road, Boulevard, Drive, or Street”

- ✓ City, State, and Zip Code

```
grep -Erai '[a-z]+\s+[a-z]+\s+([0-9]{5}(?:\-[0-9]{4})?)$' <Directory>
```

- How to read this regular expression: “A general regular expression that recursively searches a directory case insensitively and displays all patterns in line with city, state, and zip code information”

Basic Examples

- Using groups and ranges, character classes, quantifiers, and anchors develop regular expression patterns for:
 - ✓ Dollar Amounts

```
grep -Ear '\$[0-9]+(\(|.|)*[0-9]+)' <Directory>
```

- How to read this regular expression: “A general regular expression that recursively searches a directory and prints all lines containing a \$ sign followed by one or more digits followed by one or more groups that starts with a , or . and groups of one or more digits ”

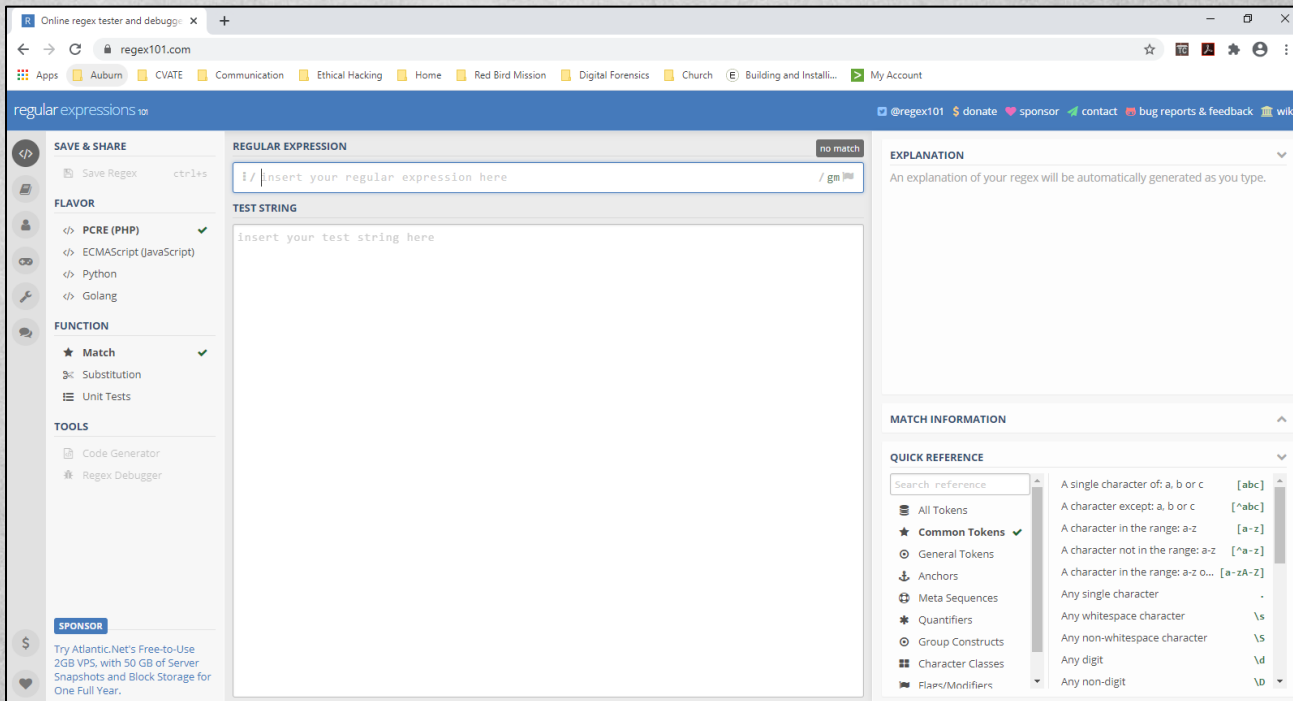
- ✓ Dates

```
grep -Ear '(([1-2][0-9])|([1-9])|(3[0-1]))/((1[0-2])|([1-9]))/[0-9]{4}' <Directory>
```

- How to read this regular expression: “A general regular expression that recursively searches and displays all output of a line that contains the data format XX/XX/XXXX”

Regular Expression Evaluator

- Provides a dynamic method of generating regular expressions as well as providing proper syntax for different languages



Forensic Analysis with Regular Expressions

Data Categories

- Forensically valuable data that regular expressions can find:
 - ✓ File extensions
 - ✓ Uniform Resource Locators (URLs)
 - ✓ Social Security Numbers
 - ✓ MAC Addresses
 - ✓ IP Addresses
 - ✓ Credit Cards
 - ✓ Email Addresses
 - ✓ Phone Numbers
- Regular expression and stream processing can be expanded to aid in:
 - ✓ Log and Audit Analysis
 - ✓ Intrusion Detection
 - ✓ Malware Detection

File Extensions

- Based on file analysis commonly used extensions:
 - ✓ Image Files – JPEG / GIF / PNG / TIFF / BMP
 - ✓ Audio Files – WAV / MPEG / ASF / WMA /
 - ✓ Video Files – MPEG / AVI / MOV
 - ✓ Archive Files – ZIP / RAR / 7Z
 - ✓ Document Files – CSV / DOC / DOCX / XLS / PDF
 - ✓ Other Notable Extensions – LOG / DAT / TEX / MSG / PST
- Notice that this analyzes file extensions and does not use signature analysis (i.e. magic numbers)
- Searching for file extensions along with signature analysis can indicate manipulation of file types

File Extensions

- Use a regular expression that lists files of different extensions
`grep -Eri '\.(docx?|log|msg|odt|csv|dat|rtf|tex|txt)$' <Directory>`
- How to read this regular expression:
 - ✓ A general regular expression that is case insensitive and recursively searches a directory for strings containing a “.” followed by either doc or docx, log, msg, odt, csv, dat, rtf, tex, or txt

```
maildir/smith-m/sent_items/33.:http://home.enron.com:84/messaging/signup.doc
maildir/smith-m/inbox/69.:http://home.enron.com:84/messaging/signup.doc
maildir/smith-m/_sent_mail/68.: - ALEXIS.DOC
maildir/semperger-c/sent_items/22.:http://www.oaticerts.com/repository/Company_Information.doc
maildir/semperger-c/sent_items/204.: File: Enron Comp log request 11-14-01.doc
maildir/semperger-c/sent_items/33.:http://www.oaticerts.com/repository/company_information.doc
maildir/semperger-c/inbox/15.:http://www.oaticerts.com/repository/company_information.doc
maildir/semperger-c/deleted_items/147.:http://www.oaticerts.com/repository/company_information.doc
maildir/semperger-c/deleted_items/46.:http://www.oaticerts.com/repository/Company_Information.doc
maildir/semperger-c/deleted_items/230.:http://www.oaticerts.com/repository/company_information.doc
maildir/semperger-c/deleted_items/43.:http://www.oaticerts.com/repository/Company_Information.doc
maildir/semperger-c/deleted_items/45.:http://www.oaticerts.com/repository/Company_Information.doc
maildir/semperger-c/deleted_items/13.:http://www.oaticerts.com/repository/Company_Information.doc
maildir/cash-m/data_protection/4.: - DPP-Master Data Protection Agmt.doc
maildir/cash-m/data_protection/4.: - DPP-MDCK markup.doc
maildir/cash-m/data_protection/4.: - Memo-Data Protection Safe Harbor.doc
maildir/cash-m/resumes_for_recruiting/12.: - Ed Resume 3 01.doc
maildir/cash-m/sent/325.:> Name: August25Plan.doc
maildir/cash-m/sent/545.: - Redline AMPS Assignment Agreement.doc
```


File Extension vs. File Signature

- Use a regular expression that matches on file signatures

```
hexdump -C ntfs.dd | grep -E '(%PDF|7z¼~|GIF87a|GIF89a|^PK.....|RIFF....AVI LIST|ftypqt|PNG....)' -C 3
```

- How to read this regular expression:

- ✓ Taking the hexdump output of ntfs.dd filter out results that contain a file signature and show three lines before and after each match

```
$ hexdump -C ntfs.dd | grep -E '(%PDF|7z¼~|GIF87a|GIF89a|^PK.....|RIFF....AVI LIST|ftypqt|PNG....)' -C 3
0012dfd0 c0 81 b7 b7 ed 22 c6 43 db cc 6c 67 26 9d b2 4b |.....".C..lg&..K|
0012dfe0 f5 50 a9 71 05 f6 66 66 51 e3 0a fd a8 a4 c1 65 |.P.q...ffQ.....e|
0012dff0 de 8c 40 e8 70 18 15 84 11 c8 51 ac 07 f2 bb 98 |..@.p.....Q.....|
0012e000 25 50 44 46 2d 31 2e 33 0d 25 e2 e3 cf d3 0d 0a |%PDF-1.3.%.....|
0012e010 33 30 20 30 20 6f 62 6a 0d 3c 3c 20 0d 2f 4c 69 |30 0 obj.<< ./Li|
0012e020 6e 65 61 72 69 7a 65 64 20 31 20 0d 2f 4f 20 33 |nearized 1 ./O 3|
0012e030 32 20 0d 2f 48 20 5b 20 39 38 30 20 32 33 37 20 |2 ./H [ 980 237 |
---
0014afd0 6d 40 6b d0 0a b4 04 2d f0 9b 6e 8e 9f ac 19 68 |m@k....-..n....h|
0014afe0 8a df 74 13 5c dd 18 bf 50 23 d0 10 3f 6e 03 fc |..t.\...P#..?n...|
0014aff0 42 26 6e 89 05 f5 41 3d 10 a3 ed e1 42 5d 6d 4f |B&n...A=...B]mO|
0014b000 25 50 44 46 2d 31 2e 33 0d 25 e2 e3 cf d3 0d 0a |%PDF-1.3.%.....|
0014b010 35 36 20 30 20 6f 62 6a 0d 3c 3c 20 0d 2f 4c 69 |56 0 obj.<< ./Li|
0014b020 6e 65 61 72 69 7a 65 64 20 31 20 0d 2f 4f 20 35 |nearized 1 ./O 5|
0014b030 38 20 0d 2f 48 20 5b 20 39 38 30 20 32 35 37 20 |8 ./H [ 980 257 |
---
00173fd0 50 a4 67 74 26 35 68 d5 88 8d 9c 47 76 9b 0d e7 |P.gt&5h....Gv...|
00173fe0 07 ca ef d3 47 0e f0 da 02 e0 0d 66 3f c0 38 c7 |....G.....f?.8.|
00173ff0 89 24 3f c5 13 7c aa 6e f4 eb 0a 50 b2 63 3e c0 |.$.?..|.n...P.c>.|
00174000 25 50 44 46 2d 31 2e 33 0d 25 e2 e3 cf d3 0d 0a |%PDF-1.3.%.....|
00174010 32 31 20 30 20 6f 62 6a 0d 3c 3c 20 0d 2f 4c 69 |21 0 obj.<< ./Li|
00174020 6e 65 61 72 69 7a 65 64 20 31 20 0d 2f 4f 20 32 |nearized 1 ./O 2|
00174030 33 20 0d 2f 48 20 5b 20 39 38 30 20 32 32 38 20 |3 ./H [ 980 228 |
```

Uniform Resource Locators

- Analyzing files for web related traffic can assist with identifying user intent
- Known high level domains
 - ✓ .com, .org, .net, .mil, .gov, .edu
- A regular expression to identify URL's

```
grep -Eri 'https?:\W[a-zA-Z0-9._-]+(V?)*' <Directory>
```
- How to read this regular expression:
 - ✓ A general regular expression that is case insensitive and recursively searches a directory for strings containing either “http://” or “https://” with 1 or more groupings of the character set [a-zA-Z0-9._-] followed by either nothing or a “/”

```
maildir/lokey-m/all_documents/2144.:http://esource.enron.com
maildir/lokey-m/all_documents/281.:Click on, http://www.ots.enron.com/united , to see the excitement in action=
maildir/lokey-m/all_documents/281.:(http://unitedway.enron.com), starting August 9 to make your United Way=20
maildir/lokey-m/all_documents/2103.:click here to access - http://nahou-wwwrms02p.ets.enron.com/ . Your ID and
maildir/lokey-m/all_documents/2165.:http://www.ets.enron.com/Services/Solution_Center/default.htm
maildir/lokey-m/all_documents/2165.:Self-Directed Task Force http://www.ets.enron.com/Services/SDWT/default.htm
maildir/lokey-m/all_documents/2165.:Safety Site http://www.ots.enron.com/safety/default.htm
maildir/lokey-m/all_documents/2165.:Engineering Standards http://www.ots.enron.com/docs/stds/0000-1.htm
maildir/lokey-m/all_documents/2137.:URL: http://www20.cera.com/eprofile?u=35&m=2234
maildir/lokey-m/all_documents/2137.:http://www20.cera.com/ceraweek/
```


URL Filtering

- The original grep search filters out known URLs, but additional commands can help to identify URL frequency

```
grep -Eroi 'https?:V[a-zA-Z0-9._-]+(V?)*' <Directory> | cut -d ":" -f2-3 | sort -n | uniq -c | sort -n
```

```
1661 http://a676.g.akamaitech.net/  
1716 http://www.expedia.com/  
1732 http://www.sfgate.com/  
1761 http://wordsmith.org/  
1791 http://continentalairlines.rsc01.net/  
1793 http://www.economist.com/  
1848 http://enews.buy.com/  
1848 http://football292.fantasy.sportsline.com/  
1866 http://www.businessweek.com/  
1882 http://www.cera.com/  
1906 http://www.pmaconference.com/  
1997 http://itcapps.corp.enron.com/  
2043 http://www.multexinvestor.com/  
2058 http://www.amazon.com/  
2083 http://explorer.msn.com  
2181 http://travelcity1.m0.net/  
2447 http://www.powermarketers.com/  
3575 http://www.sportsline.com/  
3827 http://pubs.bna.com/  
3972 http://www.energycentral.com/  
4250 http://football299.fantasy.sportsline.com/  
4837 http://football222.fantasy.sportsline.com/  
7145 http://www.fool.com/  
7241 http://www.carrfut.com/  
8832 http://www.rigzone.com/  
12556 http://www.nytimes.com/
```


Social Security Numbers

- Personally Identifiable Information (PII) such as SSNs may be contained on a computer under investigation or exfiltrated during a breach
- Social Security Number Format: XXX-XX-XXXX

```
grep -Er '[0-9]{3}-[0-9]{2}-[0-9]{4}' <Directory>
```

- How to read this regular expression:
 - ✓ Create a general regular expression that recursively searches a directory for strings that start with three consecutive digits followed immediately by a “-”, followed by two consecutive digits followed by a “-”, followed by four consecutive digits

```
$ grep -Er '[0-9]{3}-[0-9]{2}-[0-9]{4}' maildir/  
maildir/derrick-j/contacts/213.:011-44-7776-183363 (M)  
maildir/derrick-j/contacts/205.:011-44-7768-736-209 (M)  
maildir/derrick-j/inbox/162.:Business and home number: 011-44-1732-832-807  
maildir/derrick-j/inbox/162.:Cell Number: 011-44-7811-4549-33  
maildir/martin-t/contacts/11.:464-41-6228  
maildir/martin-t/contacts/16.:SS # 345-64-6203  
maildir/giron-d/inbox/29.:Subject:=0902-01-2002 Key Points  
maildir/kaminski-v/sent_items/2601.:?@*?@TEL?@0566-98-1781?@FAX?@0566-98-4794 *  
maildir/kaminski-v/sent_items/2599.:?@*?@TEL?@0566-98-1781?@FAX?@0566-98-4794 *  
maildir/kaminski-v/sent_items/2591.:?@*?@TEL?@0566-98-1781?@FAX?@0566-98-4794 *  
maildir/kitchen-l/_americas/esvl/408.:And Michael's home number is 011-44-1582-760-140  
maildir/haedicke-m/inbox/221.:And Michael's home number is 011-44-1582-760-140  
maildir/haedicke-m/inbox/315.:Business and home number: 011-44-1732-832-807  
maildir/haedicke-m/inbox/315.:Cell Number: 011-44-7811-4549-33  
maildir/delaine-d/inbox/6.:Business and home number: 011-44-1732-832-807  
maildir/delaine-d/inbox/6.:Cell Number: 011-44-7811-4549-33  
maildir/richey-c/r/3.:Institute for Forest Biometrics & Phone: ++49-551-39-12107  
maildir/richey-c/r/3.:Applied Computer Science Fax : ++49-551-39-3465
```

- Notice the mixture of true and false positives
- What can we do to improve this search?

Network Information

- There are numerous network related artifacts that can be collected during a forensic analysis including MAC and IP address information
 - ✓ MAC Address Format: XX:XX:XX:XX:XX:XX (Hexadecimal Values)
 - ✓ IP Addresses Format: XXX.XXX.XXX.XXX (0 – 255 for each octet)
- Identify known commands and locations for network related data
 - ✓ ifconfig
 - ✓ arp
 - ✓ netstat

MAC and IP Addresses

- MAC Address Search:

```
grep -Eri '([0-9a-f]{2}:){5}[0-9a-f]{2}' <Directory>
```

- How to read this regular expression: “Create a general regular expression that recursively searches a directory for two case insensitive hexadecimal values followed immediately by a “:” that are in 5 groups, followed by two consecutive hexadecimal values”
- IP Address Search:

```
grep -Ero '([0-9]{1,3}\.){3}[0-9]{1,3}' <Directory>
```
- How to read this regular expression: “Create a general regular expression that recursively searches a directory for three-digit values followed by a “.” that are in three groups, followed immediately by one to three digits”

Sorting Expressions

`grep -Ero '([0-9]{1,3}\.){3}[0-9]{1,3}' maildir/ | sort -n | uniq -c | sort -n`

```
15 maildir/lokey-m/deleted_items/59.:193.128.182.100
15 maildir/lokey-m/publications/3.:193.128.182.100
15 maildir/watson-k/deleted_items/304.:193.128.182.100
15 maildir/watson-k/e_mail_bin/91.:193.128.182.100
17 maildir/farmer-d/all_documents/987.:216.32.31.189
17 maildir/farmer-d/discussion_threads/1955.:216.32.31.189
17 maildir/farmer-d/personal/278.:216.32.31.189
17 maildir/holst-k/deleted_items/87.:64.214.225.20
18 maildir/farmer-d/all_documents/1177.:216.32.31.189
18 maildir/farmer-d/discussion_threads/1786.:216.32.31.189
18 maildir/farmer-d/personal/263.:216.32.31.189
18 maildir/kaminski-v/deleted_items/1241.:199.97.97.79
18 maildir/lewis-a/deleted_items/1145.:209.114.223.147
19 maildir/lokey-m/all_documents/578.:216.32.31.189
19 maildir/lokey-m/discussion_threads/545.:216.32.31.189
19 maildir/lokey-m/personal/178.:216.32.31.189
19 maildir/lokey-m/personal/30.:216.32.31.189
19 maildir/lucci-p/deleted_items/373.:64.215.193.68
20 maildir/lucci-p/deleted_items/246.:63.209.29.152
33 maildir/cuilla-m/deleted_items/160.:209.208.252.77
33 maildir/cuilla-m/deleted_items/453.:209.208.252.77
33 maildir/cuilla-m/inbox/24.:209.208.252.77
49 maildir/cuilla-m/deleted_items/297.:63.148.233.25
171 maildir/kaminski-v/deleted_items/1241.:199.97.97.163
```

Direct Network Access

- When a forensic analysis allows direct access to system(s), regular expressions can be used in conjunction with common networking commands

- MAC Address Search:

```
ifconfig -a | grep -Eoi '([0-9a-f]{2}:){5}[0-9a-f]{2}'
```

```
arp -a | grep -Eoi '([0-9a-f]{2}:){5}[0-9a-f]{2}'
```

- How to read this regular expression: “Take the output of ifconfig command and filter out two case insensitive hexadecimal values followed immediately by a “:” that are in 5 groups, followed by two consecutive hexadecimal values”
- IP Address Search:

```
ifconfig | grep -Eo '([0-9]{1,3}\.){3}[0-9]{1,3}'
```

- How to read this regular expression: “Take the output of the ifconfig command and filter out three-digit values followed by a “.” that are in three groups, followed immediately by one to three digits”

Direct Network Access Regular Expressions

```
ifconfig | grep -Eo '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'  
ifconfig | grep -Eo '([0-9]{1,3}\.){3}[0-9]{1,3}'
```

```
$ ifconfig | grep -Eo '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'  
172.17.0.1  
255.255.0.0  
172.17.255.255  
192.168.48.151  
255.255.255.0  
192.168.48.255  
127.0.0.1  
255.0.0.0
```


Credit Card Regular Expressions

- Another forensically valuable piece of data is credit card or account information
- Credit card format: XXXX-XXXX-XXXX-XXXX

```
grep -Er '([0-9]{4}-){3}[0-9]{4}' <Directory>
```

- How to read this regular expression: “A general regular expression that recursively searches a directory for groups of four consecutive digits followed immediately by “-” and immediately followed by four consecutive digits”
- This is a good example of how if a regular expression is not properly configured, it can lead to numerous false positives

```
$ grep -Er '([0-9]{4}-){3}[0-9]{4}' RegEx/maildir/  
RegEx/maildir/wolfe-j/10_saved/31.:1398-0764-6929-7815-1330  
RegEx/maildir/wolfe-j/inbox/156.:1398-0764-6929-7815-1330  
RegEx/maildir/love-p/personal/18.:> 0597-5369-4945-1793-6300  
RegEx/maildir/watson-k/inbox/345.:0152-5832-4793-8519-6897  
RegEx/maildir/baughman-d/discussion_threads/52.:0242-7961-6496-3300-2166  
RegEx/maildir/baughman-d/all_documents/53.:0242-7961-6496-3300-2166  
RegEx/maildir/baughman-d/personal/internet/26.:0242-7961-6496-3300-2166
```

Company Specific CC Regular Expressions

- Some of the more well-known credit card companies will have specific formats
 - Visa

```
grep -Er '^4[0-9]{12}(:[0-9]{3})?$', <Directory>
```

- MasterCard

```
grep -Er '^(?:5[1-5][0-9]{2}|222[1-9]|22[3-9][0-9]|2[3-6][0-9]{2}|27[01][0-9]|2720)[0-9]{12}$', <Directory>
```

- American Express

```
grep -Er '^3[47][0-9]{13}$', <Directory>
```

- Discover

```
grep -Er '^6(?:011|5[0-9]{2})[0-9]{12}$', <Directory>
```


Email Address Regular Expressions

- Known high level domains

- .com, .org, .net, .mil, .gov, .edu

```
grep -Eroi '[A-Za-z0-9\._%\+\-]+\@[A-Za-z0-9\._%\+\-]+\.(com|org|net|mil|gov|edu)' <Directory>
```

- How to read this regular expression: “A general regular expression that is case insensitive and recursively searches a directory for a string that has at least 1 printable character before the “@” symbol followed immediately by at least 1 printable character, followed by a “.” and the high-level domain”

```
RegEx/maildir/lucci-p/sent_items/80.:bob.williams@elpaso.com
RegEx/maildir/lucci-p/sent_items/80.:Bob.Williams@ElPaso.com
RegEx/maildir/lucci-p/sent_items/237.:t..lucci@enron.com
RegEx/maildir/lucci-p/sent_items/237.:christina_122367@hotmail.com
RegEx/maildir/lucci-p/sent_items/237.:christina_122367@hotmail.com
RegEx/maildir/lucci-p/sent_items/237.:christina_122367@hotmail.com
RegEx/maildir/lucci-p/sent_items/237.:Paul.T.Lucci@ENRON.com
RegEx/maildir/lucci-p/sent_items/237.:christina_122367@hotmail.com
RegEx/maildir/lucci-p/sent_items/237.:enron.messaging.administration@enron.com
RegEx/maildir/lucci-p/sent_items/126.:t..lucci@enron.com
RegEx/maildir/lucci-p/sent_items/126.:christina_122367@hotmail.com
RegEx/maildir/lucci-p/sent_items/126.:christina_122367@hotmail.com
RegEx/maildir/lucci-p/sent_items/126.:christina_122367@hotmail.com
RegEx/maildir/lucci-p/sent_items/126.:plucci@enron.com
RegEx/maildir/lucci-p/sent_items/278.:t..lucci@enron.com
RegEx/maildir/lucci-p/sent_items/278.:mark.whitt@enron.com
RegEx/maildir/lucci-p/sent_items/278.:tyrell.harrison@enron.com
RegEx/maildir/lucci-p/sent_items/278.:MMALINOW@westernngas.com
RegEx/maildir/lucci-p/sent_items/278.:enerfaxdaily@enerfax.com
RegEx/maildir/lucci-p/sent_items/278.:mmalinow@westernngas.com
```


Combining Regular Expressions

- Instead of searching for single regular expressions at a time, the generated regular expressions can be added to a file and referenced
- Example of file with added regular expressions:

```
GNU nano 2.5.3
'\.(docx?|log|msg|odt|csv|dat|rtf|tex|txt)'
'https?://.+\. (com|org|net|mil|gov|edu)'
'[0-9]{3}(\ |-)[0-9]{2}(\ |-)[0-9]{4}'
'([0-9a-f]{2}:){5}[0-9a-f]{2}'
'([0-9]{1,3}\.){3}[0-9]{1,3}'
'([0-9]{4}-){3}[0-9]{4}'
'[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.(com|org|net|mil|gov|edu)'
```

- Regular expression search using a regular expression list:
`grep -Erif <RegularExpressionFile> <Directory>`
- How to read this regular expression: “A general regular expression that is case insensitive, recursively searches a directory, and uses the file named <RegularExpressionFile> to match different regular expressions”