CHEAT SHEETS
FOR EVERYTHING!
PENETRATION TESTING CHEAT SHEETS
FORENSICS CHEAT SHEETS
CISO AND WEBADMIN CHEAT SHEETS
MALWARE ANALYSIS AND REVERSE ENGINEERING
TEXT EDITORS
DEVELOPERS/BUILDERS
OWASP CHEAT-SHEETS STILL IN DRAFT/BETA STAGES

CYBER TIPS

INFO-SEC RELATED CHEAT SHEETS

PENETRATION TESTING CHEAT SHEETS

Mobile Application Pentesting: https://www.peerlyst.com/posts/mobile-application-penetration-testing-cheat-sheet

Nmap : https://pen-testing.sans.org/blog/2013/10/08/nmap-cheat-sheet-1-0/

Nmap (Not printable): https://hackertarget.com/nmap-cheatsheet-a-quick-reference-guide/

Nmap 5(older version): https://nmapcookbook.blogspot.lu/2010/02/nmap-cheat-sheet.html

Nmap 5 (older version, printable) http://www.cheat-sheets.org/saved-copy/Nmap5.cheatsheet.eng.v1.pdf

Java-Deserialization https://github.com/GrrrDog/Java-Deserialization-Cheat-Sheet

Metasploit https://www.tunnelsup.com/metasploit-cheat-sheet/

Another Metasploit: http://resources.infosecinstitute.com/metasploit-cheat-sheet/

Powerupsql https://github.com/NetSPI/PowerUpSQL/wiki/PowerUpSQL-CheatSheet

Scapy https://pen-testing.sans.org/blog/2016/04/05/scapy-cheat-sheet-from-sans-sec560#

HTTP Status codes: http://suso.suso.org/docs/infosheets/HTTP_status_codes.gif

Beacon https://github.com/HarmJ0y/CheatSheets/blob/master/Beacon.pdf

Powershellempire https://github.com/HarmJ0y/CheatSheets/blob/master/Empire.pdf

Powersploit https://github.com/HarmJ0y/CheatSheets/blob/master/PowerSploit.pdf

PowerUp https://github.com/HarmJ0y/CheatSheets/blob/master/PowerUp.pdf

PowerView https://github.com/HarmJ0y/CheatSheets/blob/master/PowerView.pdf

Vim https://people.csail.mit.edu/vgod/vim/vim-cheat-sheet-en.pdf

Attack Surface Analysis

XSS Filter Evasion

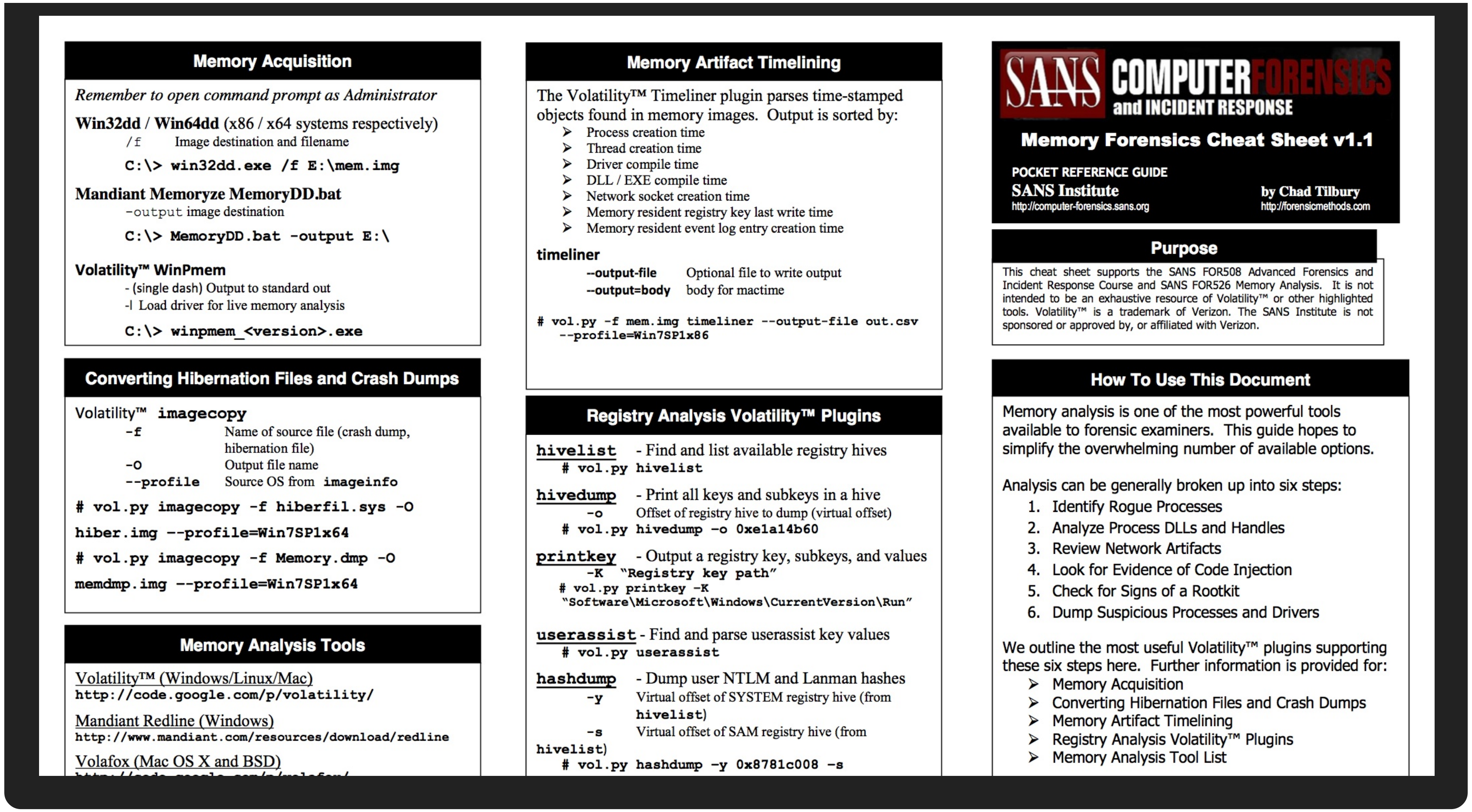
REST Assessment

Web Application Security Testing

Android Testing

IOS Developer

Mobile Jailbreaking



Memory Acquisition

Remember to open command prompt as Administrator

Win32dd / Win64dd (x86 / x64 systems respectively)

`/f` Image destination and filename

```
C:\> win32dd.exe /f E:\mem.img
```

Mandiant Memoryze MemoryDD.bat

`-output` image destination

```
C:\> MemoryDD.bat -output E:\
```

Volatility™ WinPmem

- (single dash) Output to standard out

-l Load driver for live memory analysis

```
C:\> winpmem_<version>.exe
```

Converting Hibernation Files and Crash Dumps

Volatility™ **`imagecopy`**

`-f`	Name of source file (crash dump, hibernation file)
`-O`	Output file name
`--profile`	Source OS from **`imageinfo`**

```
# vol.py imagecopy -f hiberfil.sys -O
hiber.img --profile=Win7SP1x64
# vol.py imagecopy -f Memory.dmp -O
memdmp.img --profile=Win7SP1x64
```

Memory Analysis Tools

Volatility™ (Windows/Linux/Mac)
`http://code.google.com/p/volatility/`

Mandiant Redline (Windows)
`http://www.mandiant.com/resources/download/redline`

Volafox (Mac OS X and BSD)

Memory Artifact Timelining

The Volatility™ Timeliner plugin parses time-stamped objects found in memory images. Output is sorted by:

- Process creation time
- Thread creation time
- Driver compile time
- DLL / EXE compile time
- Network socket creation time
- Memory resident registry key last write time
- Memory resident event log entry creation time

timeliner

--output-file	Optional file to write output
--output=body	body for mactime

```
# vol.py -f mem.img timeliner --output-file out.csv
   --profile=Win7SP1x86
```

Registry Analysis Volatility™ Plugins

`hivelist` - Find and list available registry hives

```
   # vol.py hivelist
```

`hivedump` - Print all keys and subkeys in a hive

`-o` Offset of registry hive to dump (virtual offset)

```
   # vol.py hivedump -o 0xe1a14b60
```

`printkey` - Output a registry key, subkeys, and values

`-K "Registry key path"`

```
  # vol.py printkey -K
  "Software\Microsoft\Windows\CurrentVersion\Run"
```

`userassist` - Find and parse userassist key values

```
   # vol.py userassist
```

`hashdump` - Dump user NTLM and Lanman hashes

`-y` Virtual offset of SYSTEM registry hive (from `hivelist`)

`-s` Virtual offset of SAM registry hive (from `hivelist`)

```
   # vol.py hashdump -y 0x8781c008 -s
```

SANS COMPUTER FORENSICS and INCIDENT RESPONSE

Memory Forensics Cheat Sheet v1.1

POCKET REFERENCE GUIDE

SANS Institute
http://computer-forensics.sans.org

by Chad Tilbury
http://forensicmethods.com

Purpose

This cheat sheet supports the SANS FOR508 Advanced Forensics and Incident Response Course and SANS FOR526 Memory Analysis. It is not intended to be an exhaustive resource of Volatility™ or other highlighted tools. Volatility™ is a trademark of Verizon. The SANS Institute is not sponsored or approved by, or affiliated with Verizon.

How To Use This Document

Memory analysis is one of the most powerful tools available to forensic examiners. This guide hopes to simplify the overwhelming number of available options.

Analysis can be generally broken up into six steps:

1. Identify Rogue Processes
2. Analyze Process DLLs and Handles
3. Review Network Artifacts
4. Look for Evidence of Code Injection
5. Check for Signs of a Rootkit
6. Dump Suspicious Processes and Drivers

We outline the most useful Volatility™ plugins supporting these six steps here. Further information is provided for:

- Memory Acquisition
- Converting Hibernation Files and Crash Dumps
- Memory Artifact Timelining
- Registry Analysis Volatility™ Plugins
- Memory Analysis Tool List

FORENSICS CHEAT SHEETS

Master boot record, guid partition table, NTFS volume boot record, Master file table record, standard information attribute, $Attribute list attribute, $file name attribute, and more forensics posters/cheat sheets: https://github.com/Invoke-IR/ForensicPosters

Mounting DD Images https://sift.readthedocs.io/en/latest/cheatsheet/

SANS Cheat sheet : http://digital-forensics.sans.org/community/cheat-sheets

---

CISO AND WEBADMIN CHEAT SHEETS

CSP cheat sheet https://scotthelme.co.uk/csp-cheat-sheet/#require-sri-for (via Scott Helme)

HTTP Status codes http://suso.suso.org/docs/infosheets/HTTP_status_codes.gif

The windows logging Cheat Sheet https://www.malwarearchaeology.com/s/Windows-Logging-Cheat-Sheet_ver_Oct_2016.pdf

The Windows Splunk Logging Cheat Sheet

The Windows File Auditing Logging Cheat Sheet

The Windows Registry Auditing Logging Cheat Sheet

The Windows PowerShell Logging Cheat Sheet

Curl HTTP : https://bagder.github.io/curl-cheat-sheet/http-sheet.html

Virtual Patching

MALWARE ANALYSIS AND REVERSE ENGINEERING

Malware analysis: http://r00ted.com/cheat%20sheet%20reverse%20v5.png

ADB: https://github.com/maldroid/adb_cheatsheet

TEXT EDITORS

VIM : https://people.csail.mit.edu/vgod/vim/vim-cheat-sheet-en.pdf

---

DEVELOPERS/BUILDERS

- 3rd Party Javascript Management
- Access Control
- AJAX Security Cheat Sheet
- Authentication (ES)
- Bean Validation Cheat Sheet
- Choosing and Using Security Questions
- Clickjacking Defense
- C-Based Toolchain Hardening
- Credential Stuffing Prevention Cheat Sheet
- Cross-Site Request Forgery (CSRF) Prevention
- Cryptographic Storage
- Deserialization
- DOM based XSS Prevention
- Forgot Password
- HTML5 Security

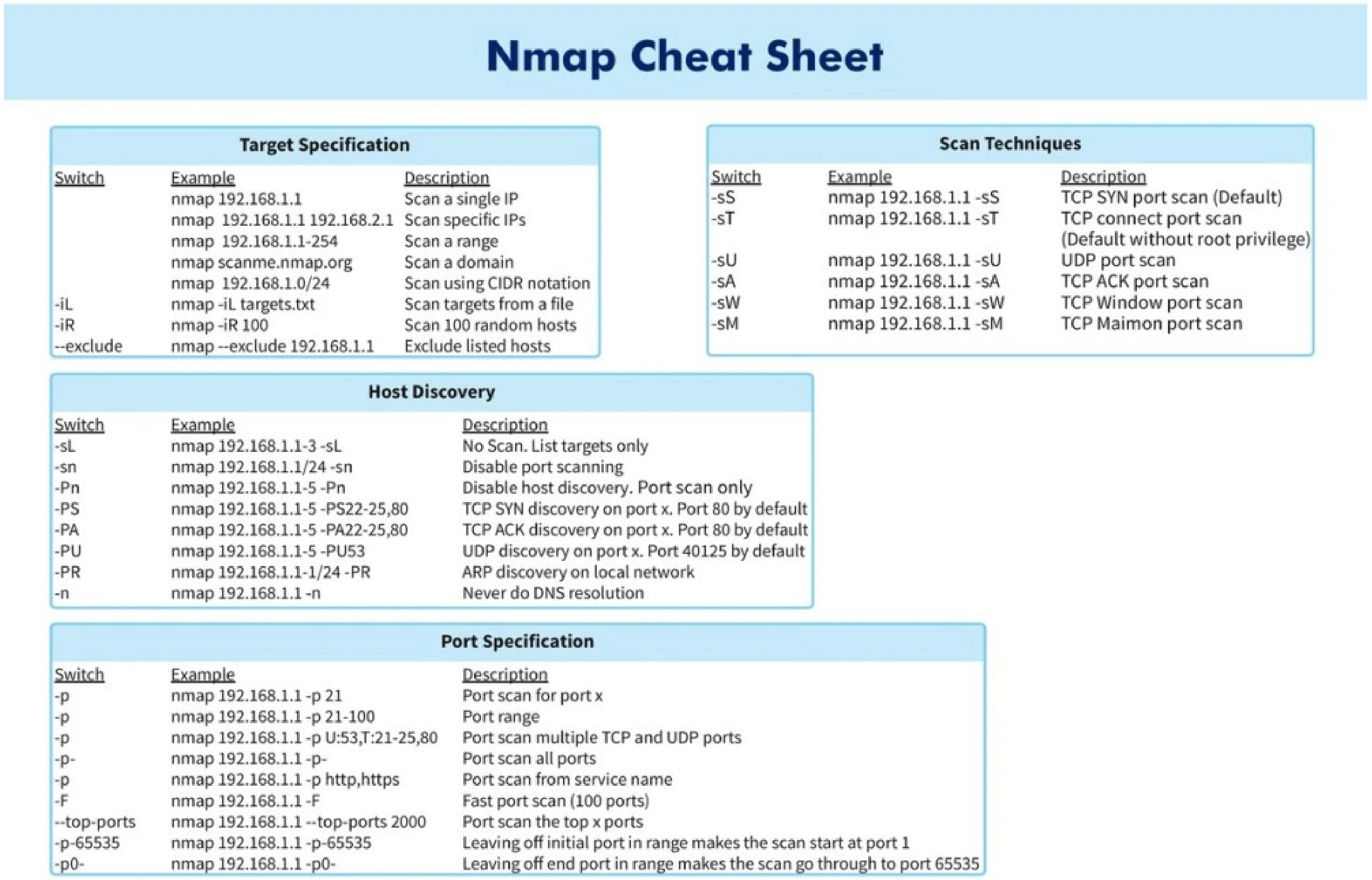
- HTTP Strict Transport Security
- Injection Prevention Cheat Sheet
- Input Validation
- JAAS
- LDAP Injection Prevention
- Logging
- Mass Assignment Cheat Sheet
- .NET Security
- OWASP Top Ten
- Password Storage
- Pinning
- Query Parameterization
- Ruby on Rails
- REST Security
- Session Management
- SAML Security
- SQL Injection Prevention
- Transaction Authorization
- Transport Layer Protection
- Unvalidated Redirects and Forwards
- User Privacy Protection
- Web Service Security
- XSS (Cross Site Scripting) Prevention

- XML External Entity (XXE) Prevention Cheat Sheet

OWASP CHEAT-SHEETS STILL IN DRAFT/BETA STAGES

- OWASP_Cheat_Sheet_Series
- Application Security Architecture
- Business Logic Security
- Command Injection Defense Cheat Sheet
- PHP Security
- Regular Expression Security Cheatsheet
- Secure Coding
- Secure SDLC
- Threat Modeling
- Grails Secure Code Review
- IOS Application Security Testing
- Key Management
- Insecure Direct Object Reference Prevention
- Content Security Policy

Some InfoGraph



Nmap Cheat Sheet

Target Specification

Switch	Example	Description
	nmap 192.168.1.1	Scan a single IP
	nmap 192.168.1.1 192.168.2.1	Scan specific IPs
	nmap 192.168.1.1-254	Scan a range
	nmap scanme.nmap.org	Scan a domain
	nmap 192.168.1.0/24	Scan using CIDR notation
-iL	nmap -iL targets.txt	Scan targets from a file
-iR	nmap -iR 100	Scan 100 random hosts
--exclude	nmap --exclude 192.168.1.1	Exclude listed hosts

Scan Techniques

Switch	Example	Description
-sS	nmap 192.168.1.1 -sS	TCP SYN port scan (Default)
-sT	nmap 192.168.1.1 -sT	TCP connect port scan (Default without root privilege)
-sU	nmap 192.168.1.1 -sU	UDP port scan
-sA	nmap 192.168.1.1 -sA	TCP ACK port scan
-sW	nmap 192.168.1.1 -sW	TCP Window port scan
-sM	nmap 192.168.1.1 -sM	TCP Maimon port scan

Host Discovery

Switch	Example	Description
-sL	nmap 192.168.1.1-3 -sL	No Scan. List targets only
-sn	nmap 192.168.1.1/24 -sn	Disable port scanning
-Pn	nmap 192.168.1.1-5 -Pn	Disable host discovery. Port scan only
-PS	nmap 192.168.1.1-5 -PS22-25,80	TCP SYN discovery on port x. Port 80 by default
-PA	nmap 192.168.1.1-5 -PA22-25,80	TCP ACK discovery on port x. Port 80 by default
-PU	nmap 192.168.1.1-5 -PU53	UDP discovery on port x. Port 40125 by default
-PR	nmap 192.168.1.1-1/24 -PR	ARP discovery on local network
-n	nmap 192.168.1.1 -n	Never do DNS resolution

Port Specification

Switch	Example	Description
-p	nmap 192.168.1.1 -p 21	Port scan for port x
-p	nmap 192.168.1.1 -p 21-100	Port range
-p	nmap 192.168.1.1 -p U:53,T:21-25,80	Port scan multiple TCP and UDP ports
-p-	nmap 192.168.1.1 -p-	Port scan all ports
-p	nmap 192.168.1.1 -p http,https	Port scan from service name
-F	nmap 192.168.1.1 -F	Fast port scan (100 ports)
--top-ports	nmap 192.168.1.1 --top-ports 2000	Port scan the top x ports
-p-65535	nmap 192.168.1.1 -p-65535	Leaving off initial port in range makes the scan start at port 1
-p0-	nmap 192.168.1.1 -p0-	Leaving off end port in range makes the scan go through to port 65535

Src: Peerlist

LIKE TO SHARE :)

Like

Be the first to like this.

RELATED

PENTESTING RESOURCES
In "Cyber Security"

The Vigilante Who Hacked Hacking Team
In "Digital Forensic"

FILELESS MALWARE ATTACKS : INTRO
In "Digital Forensic"

#Cheatsheet #memory #vapt #forensic #malware #hacking #hacker #guildeline #malware #sans #volatility #owasp #nmap #smartphone #jtag #smartphone #android #ios #ram #sop #php #websec #websecurity #web #

PREVIOUS POST

CHALLENGES IN MOBILE FORENSICS

NEXT POST

AWESOME THREAT INTELLIGENCE

Leave a Reply

Enter your comment here...

This site uses Akismet to reduce spam. Learn how your comment data is processed.

TRANSLATE

Select Language ▼

Powered by Google Translate

Search ...

Join 5,936 other followers

Enter your email address

GET LATEST UPDATES

CATEGORIES

Archive (3)

Cyber Security (20)

Cyber Tips (9)

Digital Forensic (17)

Forensics of Things (11)

Interview Bites (2)

slack area (4)

RECENT POSTS

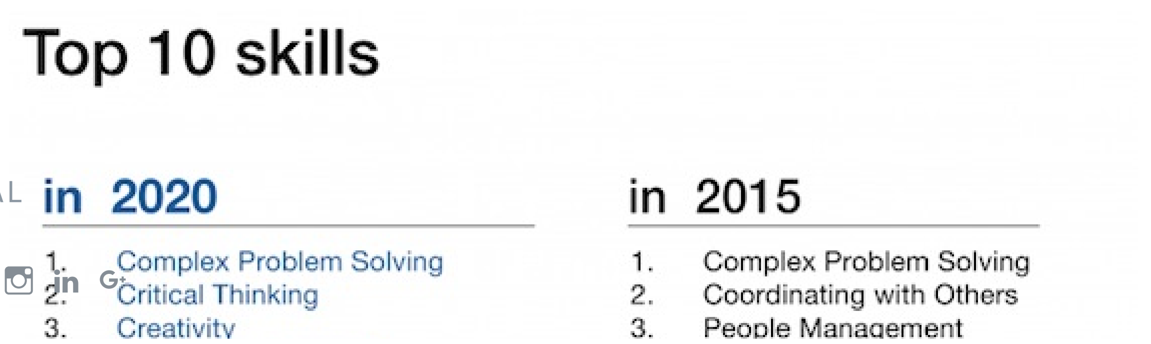
- DEFCON DFIR CTF 2019
- THREAT INTELLIGENCE
- PENTESTING RESOURCES
- TESTIMONIALS
- CYBERSECURITY CERTIFICATIONS -1
- SMARTPHONE FORENSICS – 2
- WHY OPEN SOURCE THREAT INTELLIGENCE

Tweets by @D3pak

DEEPAK KUMAR (D3) Retweeted

Brijesh Singh
@Brijeshbsingh

Changing skill landscape.



Top 10 skills

in 2020		in 2015	
1.	Complex Problem Solving	1.	Complex Problem Solving
2.	Critical Thinking	2.	Coordinating with Others
3.	Creativity	3.	People Management

Embed

View on Twitter

SOCIAL

SPAM BLOCKED

1,203 spam
blocked by **Akismet**

DIGITAL FORENSICS

DIGITAL FORENSICS IS A BRANCH OF FORENSIC SCIENCE FOCUSING ON THE RECOVERY AND INVESTIGATION OF RAW DATA RESIDING IN ELECTRONIC OR DIGITAL DEVICES. THE GOAL OF THE

PROCESS IS TO EXTRACT AND RECOVER ANY INFORMATION FROM A DIGITAL DEVICE WITHOUT ALTERING THE DATA PRESENT ON THE DEVICE.