

ClubHACKMag

1st Indian "HACKING" Magazine



Security by luck, not possible!

Issue 31 | August 2012

www.clubhack.com

TechGyan Malware Memory Forensics | **Mom's Guide** Apple iOS vulnerabilities |

ToolGyan Tamper data | **Matriux Vibhag** Matriux Ec-Centric |

Hello Readers! Here we are with the 31st issue, which will be released at c0c0n 2012 - International Cyber Security Conference. Talking about the conferences, this year let's have a Hack Night in ClubHack2012 Conference. A night where actual hackers spend time not to "break" into someone but to "make" something interesting. For more details check :- <http://www.clubhack.com/2012/>



Pankit Thakkar

Coming back to this issue we have Malware Memory Forensics in TechGyan, Apple iOS vulnerabilities in Mom's Guide, Tamper Data in ToolGyan and Matriux Ec-Centric which too will be released at c0c0n. The poster I designed would be very easy to understand for the Indian readers. Foreign readers, search for 7 chillies and 1 lemon on Google. You will understand what we are trying to say ;)

Keep your articles, feedbacks and suggestions flowing to info@chmag.in

Issue 31, August 2012.

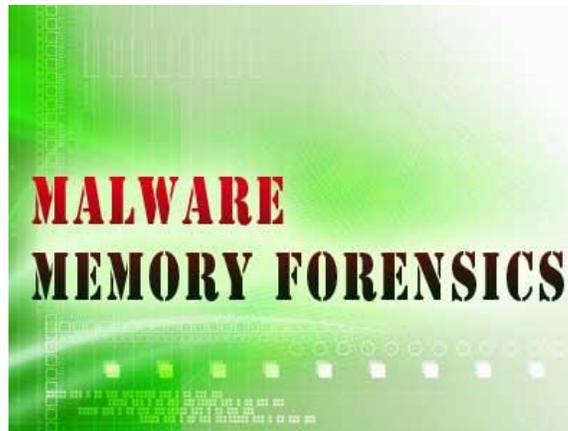
Team CHmag

- Rohit Srivastwa
rohit@clubhack.com
- Aarja Bhattacharyya
aarja@chmag.in
- Abhijeet R Patil
abhijeet@chmag.in
- Abhishek Nagar
abhishek@chmag.in
- Pankit Thakkar
pankit@chmag.in
- K.V.Prashant
good.best.guy@gmail.com
- Sagar Nangare
sagar@chmag.in
- Varun V Hirve
varun@chmag.in

www.chmag.in
info@chmag.in

CONTENTS

Pg	TechGyan
03	Network Security
Pg	ToolGyan
08	Tamper data
Pg	Mom'sGuide
13	Apple iOS vulnerabilities
Pg	LegalGyan
16	VARIOUS AUTHORITIES UNDER THE IT ACT
Pg	MatriuxVibhag
19	Matriux Ec-Centric



Malware Memory Forensics

Introduction

Memory Forensics is the analysis of the memory image taken from the running computer.

In this article, we will learn how to use Memory Forensic Toolkits such as Volatility to analyze the memory artifacts with practical real life forensics scenario.

Below are the list of steps involved in memory forensics:-

Why Memory Forensics?

Memory forensics can help in extracting forensics artifacts from a computer's memory like running process, network connections, loaded modules etc etc. It can also help in unpacking, rootkit detection and reverse engineering.

1. Memory Acquisition - This step involves dumping the memory of the target machine. on the physical machine you can use tools like Win32dd/Win64dd, Memoryze,

DumpIt, FastDump on the virtual machine, acquiring the memory image is easy, you can do it by suspending the VM and grabbing the ".vmem" file.

2. Memory Analysis - once a memory image is acquired, the next step is analyze the grabbed memory dump for forensic artifacts. tools like Volatility and Memoryze can be used to analyze the memory.

Volatility - A Quick Overview

Volatility is an advanced memory forensic framework written in python. It can be installed on multiple operating systems (Windows, Linux, Mac OS X), Installation details of volatility can be found here - <http://code.google.com/p/volatility>.

Volatility Syntax & Usage

* using -h or --help option will display help options and list of a available plugins.

Example: `python vol.py -h`

* Use -f and --profile to indicate the memory dump you are analyzing.

Example: `python vol.py -f mem.dmp -profile=WinXPSP3x86`

* To know the --profile info use below command:

Example: `python vol.py -f mem.dmp imageinfo`

Demonstration - Memory Forensics

In order to understand memory forensics and the steps involved. I have created a scenario, our analysis and flow will be based on the below scenario.

Demo Scenario

Your security device alerts, show malicious http connection to ip address 208.91.197.54 from a source ip 192.168.1.100 on 8th june 2012 at around 13:30hrs...you are asked to investigate and do memory forensics on that machine 192.168.1.100

Preparation Steps

To start with, acquire the memory image from 192.168.1.100, using memory acquisition tools. for the sake of demo, the memory dump file is named as "infected.dmp".

Demonstration - Memory Analysis

Now that we have acquired "infected.dmp", lets start our analysis.

Step 1: Start with what you know

We know from the security device alert that the host was making an http connection to 208.91.197.54. So let's look at the network connections.

Volatility's connections module, shows connection to the malicious ip made by pid 1748.

dolor sit amet Lorem Ipsum Dolor sit Amet
Loerm Ipsum dolor sit amet

Loerm Ipsum dolor sit amet Loerm Ipsum
dolor sit amet Loerm Ipsum dolor sit amet

```

root@bt:~/Volatility
File Edit View Terminal Help
root@bt:~/Volatility# python vol.py -f infected.dmp connections
Volatile Systems Volatility Framework 2.0
Offset(V) Local Address Remote Address Pid
-----
0x8943a558 192.168.1.100:1032 208.91.197.54:80 1748
root@bt:~/Volatility#
  
```

Step 2: Info about 208.91.197.54

Google search shows this ip 208.91.197.54 to be associated with malware, probably "SpyEye", we need to confirm that yet.



Step 3: Who is Pid 1748?

Since the network connection to the ip 208.91.197.54 was made by pid 1748, we need to determine which process is associated with pid 1748. "psscan" shows pid 1748 belongs to explorer.exe, also two process created during same time reported by security device (i.e. June 8th 2012).

```

root@bt:~/Volatility
File Edit View Terminal Help
root@bt:~/Volatility# python vol.py -f infected.dmp psscan
Volatile Systems Volatility Framework 2.0
-----
Offset      Name                PID      PPID     POB      Time created      Time exited
-----
0x0932b026 B6232F3A9F9.exe    1672     1748    0x0f9c02a0 2012-06-08 13:27:55
0x09339020 winiprvse.exe      584      880    0x0f9c0260 2012-02-26 12:07:19
0x0934c4a8 VMUpgradeHelper    428      700    0x0f9c0240 2012-02-26 12:07:19
0x09350740 vmtoolsd.exe       216      700    0x0f9c0220 2012-02-26 12:07:19
0x0935a360 explorer.exe       1748     1712    0x0f9c01c0 2012-02-26 12:07:17
0x093662b0 svchost.exe       964      700    0x0f9c0190 2012-02-26 12:07:11
0x094c6da0 svchost.exe       880      700    0x0f9c00e0 2012-02-26 12:07:11
0x095ffa58 cfmon.exe          1900     1748    0x0f9c0280 2012-02-26 12:07:18
0x0964c020 erm.exe      1648     1888    0x0f9c0280 2012-06-08 13:27:53
0x09656020 VMwareUser.exe    1888     1748    0x0f9c01e0 2012-02-26 12:07:10
0x09665630 winlogon.exe      656      376    0x0f9c0060 2012-02-26 12:07:11
0x097166a0 VMwareTray.exe 1880     1748    0x0f9c0180 2012-02-26 12:07:10
0x0971ea38 svchost.exe       1092     700    0x0f9c0140 2012-02-26 12:07:11
0x09732da0 csrss.exe         632      376    0x0f9c0040 2012-02-26 12:07:10
0x097ae0f0 services.exe    700      656    0x0f9c0080 2012-02-26 12:07:11
0x09811020 lsass.exe         712      656    0x0f9c00a0 2012-02-26 12:07:11
0x09821020 sss.exe          376      4      0x0f9c0020 2012-02-26 12:07:10
0x0984c8e0 svchost.exe    1124     700    0x0f9c0160 2012-02-26 12:07:11
0x0984e170 svchost.exe    1048     700    0x0f9c0120 2012-02-26 12:07:11
0x098523b0 vmacthlp.exe      868      700    0x0f9c00c0 2012-02-26 12:07:11
0x0992b030 System            4        0    0x00019000
root@bt:~/Volatility#
    
```

Step 4: Process handles of explorer.exe

Now that we know explorer.exe (which is an operating system process) was making connections to the malicious ip, there is a possibility that explorer.exe is infected.

Lets looks at the process handles of explorer.exe. The below screenshot shows Explorer.exe opens a handle to the B6232F3A9F9.exe, indicating explorer.exe might have created that process, which might also be malicious...Lets focus on explorer.exe for now.

```

root@bt:~/Volatility
File Edit View Terminal Help
root@bt:~/Volatility# python vol.py -f infected.dmp handles -p 1748 -t Process
Volatile Systems Volatility Framework 2.0
-----
Offset(V)  Pid  Type      Details
-----
0x0915a348 1748 Process  explorer.exe(1748)
0x0912b008 1748 Process  B6232F3A9F9.exe(1672)
0x0912b008 1748 Process  B6232F3A9F9.exe(1672)
root@bt:~/Volatility#
    
```

Step 5: API Hooks in explorer.exe

APIhooks module show, inline API hooks in explorer.exe and jump to an unknown location.

```

root@bt:~/Volatility
File Edit View Terminal Help
root@bt:~/Volatility# python vol.py -f infected.dmp apihooks -p 1748
Volatile Systems Volatility Framework 2.0
-----
Name      Type      Target      Value
-----
explorer.exe[1748]  inline   user32.dll!TranslateMessage(0x7e418bf6) 0x7e418bf6 JMP 0xbbb6ddc (UNKNOWN)
explorer.exe[1748]  inline   crypt32.dll!PPImportCertStore(0x772eff8f) 0x772eff8f JMP 0xbb70462 (UNKNOWN)
explorer.exe[1748]  inline   wininet.dll!HttpSendRequestA(0x7806c40) 0x7806c40 JMP 0xbb8233e (UNKNOWN)
explorer.exe[1748]  inline   wininet.dll!HttpSendRequestW(0x7805da59) 0x7805da59 JMP 0xbb8239c (UNKNOWN)
explorer.exe[1748]  inline   wininet.dll!InternetCookieHandle(0x7803465) 0x7803465 JMP 0xbb82cfa (UNKNOWN)
explorer.exe[1748]  inline   advapi32.dll!CryptEncrypt(0x70e3348) 0x70e3348 JMP 0xbb7c597 (UNKNOWN)
explorer.exe[1748]  inline   ntdll.dll!INtEnumerateValueKey(0x7c90d20) 0x7c90d20 JMP 0xbb6a7f6 (UNKNOWN)
explorer.exe[1748]  inline   ntdll.dll!NtQueryDirectoryFile(0x7c90d750) 0x7c90d750 JMP 0xbb74855 (UNKNOWN)
explorer.exe[1748]  inline   ntdll.dll!NtResumeThread(0x7c90d20) 0x7c90d20 JMP 0xbb861f8 (UNKNOWN)
explorer.exe[1748]  inline   ntdll.dll!NtSetInformationFile(0x7c90d40) 0x7c90d40 JMP 0xbb6a33a (UNKNOWN)
explorer.exe[1748]  inline   ntdll.dll!NtVdmControl(0x7c90d00) 0x7c90d00 JMP 0xbb7493b (UNKNOWN)
explorer.exe[1748]  inline   ntdll.dll!INtEnumerateValueKey(0x7c90d750) 0x7c90d750 JMP 0xbb6a7f6 (UNKNOWN)
explorer.exe[1748]  inline   ntdll.dll!INtEnumerateValueKey(0x7c90d750) 0x7c90d750 JMP 0xbb74855 (UNKNOWN)
explorer.exe[1748]  inline   ntdll.dll!INtResumeThread(0x7c90d20) 0x7c90d20 JMP 0xbb861f8 (UNKNOWN)
explorer.exe[1748]  inline   ntdll.dll!INtSetInformationFile(0x7c90d40) 0x7c90d40 JMP 0xbb6a33a (UNKNOWN)
explorer.exe[1748]  inline   ntdll.dll!INtVdmControl(0x7c90d00) 0x7c90d00 JMP 0xbb7493b (UNKNOWN)
explorer.exe[1748]  inline   ws2_32.dll!send(0x71ab4c27) 0x71ab4c27 JMP 0xbb7633e (UNKNOWN)
Finished after 17.2333598984 seconds
root@bt:~/Volatility#
    
```

Step 6: Exploring the Hooks

Disassembled hooked function (TranslateMessage), shows a short jump and then a long jump to malware location.

```

File Edit View Terminal Help
root@bt:~/Volatility# python vol.py -f infected.dmp volshell
Volatile Systems Volatility Framework 2.0
Current context: process System, pid=4, ppid=0 DTB=0x319000
Welcome to volshell! Current memory image is:
file:///root/Volatility/infected.dmp
To get help, type 'hh()'
>>> hh()
ps() : Print a process listing.
cc(offset=None, pid=None, name=None) : Change current shell context.
dd(address, length=128, space=None) : Print dwords at address.
db(address, length=128, width=16, space=None) : Print bytes as canonical hexdump.
hh(cmd=None) : Get help on a command.
dt(objct, address=None) : Describe an object or show type info.
list_entry(head, objname, offset=-1, fieldname=None, forward=True) : Traverse a _LIST_ENTRY.
dis(address, length=128, space=None) : Disassemble code at a given address.

For help on a specific command, type 'hh(<command>)'
>>> cc(pid=1748)
Current context: process explorer.exe, pid=1748, ppid=1712 DTB=0x9f9c01c0
>>> dis(0x7e418bf6, length=32)
0x7e418bf6 ebb1 JMP 0x7e418bf9
0x7e418bf8 c3 RET
0x7e418bf9 e9de31758d JMP 0xbbb6ddc
0x7e418bfe 086681 OR [ESI-0x7f], AH
0x7e418c01 7e08 JLE 0x7e418c0b
0x7e418c03 e509 IN EBX, 0x0
0x7e418c05 0f84667e0200 JZ 0x7e440a71
0x7e418c0b 6a08 PUSH 0x0
    
```

Step 7: Embedded EXE in explorer.exe

Printing the bytes at the hooked location, show the presence of embedded executable in explorer.exe.

ClubHACKMax

```
>>> db(0x0bb60000, length=256)
0bb60000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00  MZ.....
0bb60010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  .....@.....
0bb60020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0bb60030  00 00 00 00 00 00 00 00 00 00 00 00 e0 00 00 00  .....
0bb60040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0bb60050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0bb60060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0bb60070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0bb60080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0bb60090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0bb600a0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0bb600b0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0bb600c0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0bb600d0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0bb600e0  50 45 00 00 4c 01 02 00 92 60 ed 4d 00 00 00 00  PE...L...M...
0bb600f0  00 00 00 00 e0 00 02 01 0b 01 0a 00 00 a2 04 00  .....
```

Step 8: Dumping the embedded EXE

VadDump tool dumps the embedded exe from explorer.exe.

```
root@bt: ~/Volatility
root@bt:~/Volatility# python vol.py -f infected.dmp vaddump -p 1748 -D dump/
Volatile Systems Volatility Framework 2.0
Pid: 1748
root@bt:~/Volatility#
```



Step 9: VirusTotal Submission

Submission to VirusTotal, confirms the dumped executable as component of "SpyEye".

AntiVirus	Result	Update
AntiLab V3	ParasitW02/Malware	20120609
AntiVir	TR/Dropper.Gen	20120609
Avira AVL	-	20120609
Avast	Win32/Spyeye.KY [Trj]	20120609
BitDefender	-	20120609
BytePass	-	20120609
CAT-QuickHeal	-	20120609
ClamAV	-	20120609
ComodoScan	-	20120609
Comodo	-	20120609
Emsisoft	Trojan-Win32/Spyeye.K	20120609
eScan	-	20120607
F-Secure	-	20120609
F-Secure	-	20120609
Fortinet	-	20120609
GData	Win32/Spyeye.KY	20120609
Norma	Trojan-Win32/Spyeye	20120609

Step 10: Can we get more info?

Strings extracted from the dumped executable, show reference to interesting artifacts (executable and the registry key), it also shows the path to the suspicious executable B6232F3A9F9.exe.

```
Connection: close
Connection:
Content-Length:
Content-Length:
Content-Encoding: deflate
Content-Encoding: gzip
Transfer-Encoding: chunked
Content-Length: %d
HTTP/
User-Agent:
Accept-Encoding:
Keep-Alive:
Connection: keep-alive
Proxy-Connection: keep-alive
SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN
f98u
^t
860l
C:\WINDOWS\system32\WININET.dll
C:\Recycle.Bin\A705B3960358085
C:\WINDOWS\system32\ntdll.dll
C:\WINDOWS\system32\USER32.dll
C:\WINDOWS\system32\ntdll.dll
C:\WINDOWS\system32\CRYPT32.dll
C:\Recycle.Bin\B6232F3A9F9.exe
C:\Recycle.Bin\B6232F3A9F9.exe
A705B3960358085
s1PSg1LF.exe
C:\DOCUME-1\ADMINI-1\LOCALS-1\Temp\
```

Step 11: Printing the Registry Key

Printing the registry key determined from the above step (step 10) shows that, malware creates registry key to survive the reboot.

```

root@kali:~/Volatility# python vol.py -f infected.dmp printkey -K "SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN"
Volatile Systems Volatility Framework 2.0
Legend: (S) = Stable (V) = Volatile
-----
Registry: \Device\HarddiskVolume1\Documents and Settings\LocalService\NTUSER.DAT
Key name: Run (S)
Last updated: 2011-10-31 15:07:20
Subkeys:
Values:
-----
Registry: \Device\HarddiskVolume1\WINDOWS\system32\config\default
Key name: Run (S)
Last updated: 2011-10-31 20:28:57
Subkeys:
Values:
-----
Registry: \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT
Key name: Run (S)
Last updated: 2012-06-08 13:27:56
Subkeys:
Values:
REG_SZ          ctfmon.exe          : (S) C:\WINDOWS\system32\ctfmon.exe
REG_SZ          4Y3Y0C3A1F7XZHZWACQCLUD : (S) C:\Recycle.Bin\B6232F3A9F9.exe

```

Step 12: Finding the Malicious EXE on Infected Machine

Now that we know the path to the suspicious executable, let's find it on the infected machine. Finding the malicious sample from the infected host and VirusTotal submission confirms SpyEye infection.



Address	Host	Update
192.168.1.1	192.168.1.1	2019/08/05
192.168.1.2	192.168.1.2	2019/08/05
192.168.1.3	192.168.1.3	2019/08/05
192.168.1.4	192.168.1.4	2019/08/05
192.168.1.5	192.168.1.5	2019/08/05
192.168.1.6	192.168.1.6	2019/08/05
192.168.1.7	192.168.1.7	2019/08/05
192.168.1.8	192.168.1.8	2019/08/05
192.168.1.9	192.168.1.9	2019/08/05
192.168.1.10	192.168.1.10	2019/08/05

Kaspersky	Trojan-Spy.Win32.SpyEye.AK
McAfee	FWG-Spyeye4
McAfee-GW-Editin	Heuristic:BehavesLike.Win32.Malware.PFI.C
Microsoft	Trojan-Win32.EyeN
NOD32	a variant of Win32/Spy.SpyEye.CA
Norman	W32/Suspicious_Gen2.CDFJX
eProtect	Trojan-W32.Agent.820511.DW
Panda	
PCTools	Trojan-Spyeye

Conclusion

Memory forensics is a powerful technique and with a tool like Volatility it is possible to find and extract the forensic artifacts from the memory which helps in incident response, malware analysis and reverse engineering.

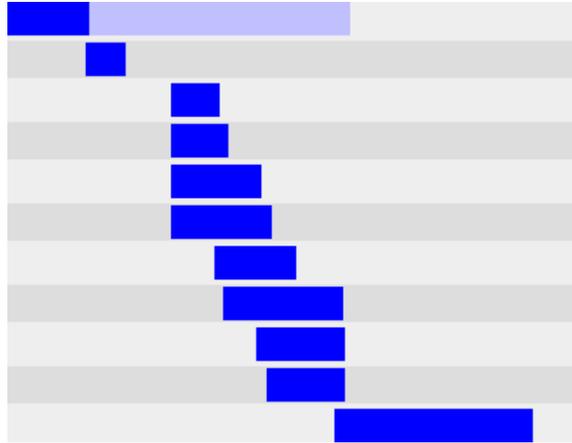
References

1. [Reversing Training Session 6 – Malware Memory Forensics](#)
2. [Volatility - An advanced memory forensics framework](#)
3. [Volatility - Volatile memory analysis research](#)
4. [MoonSols Windows Memory Toolkit](#)



Monappa

Monappa is one of the core members of the management panel at SecurityXploded. He has rich experience of about 5 years in the security domain with core expertise in exploit development and malware analysis. In addition to publishing interesting research articles, he has also delivered training sessions on 'Reversing & Malware Analysis' at SecurityXploded. Currently, he is working at Cisco Systems as an Information Security Investigator.



Tamper data

What is Tamper Data?

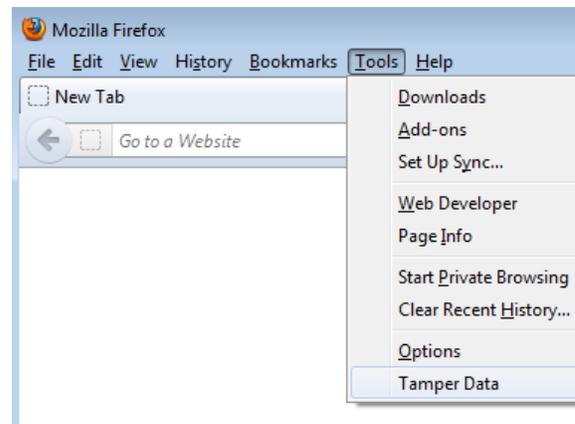
Tamper data, an add-on (extension) for Mozilla Firefox, is a fast, simple yet effective tool which can be used to do penetration testing.

Tamper Data basically gives us the power to view, record and even modify outgoing HTTP requests. Since Tamper Data is integrated into the browser, so it has no problems with the HTTPS connections, client authentication certificates or other features that the browser supports. We can trace and time the http/https connections, responses and parameters being sent.

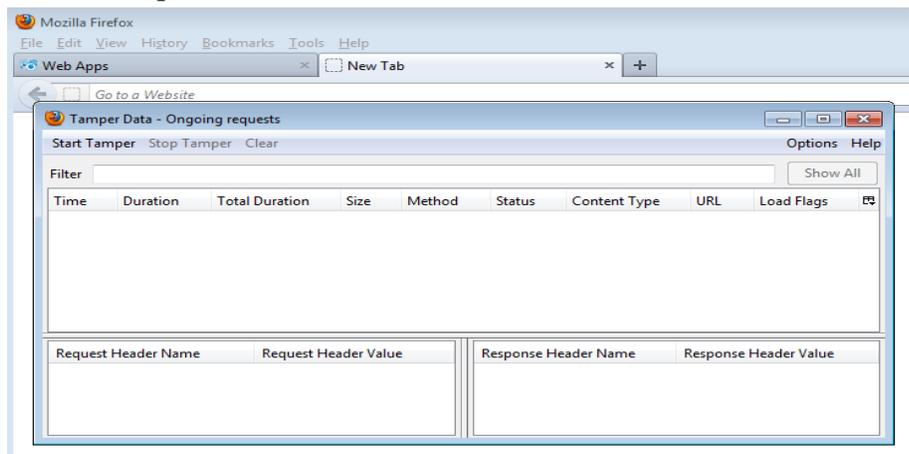
How to use Tamper Data?

Tamper Data is a plugin available for Firefox. We can easily download the xpi from Mozilla and install it. After installation of the add-on it would ask for restarting the Firefox.

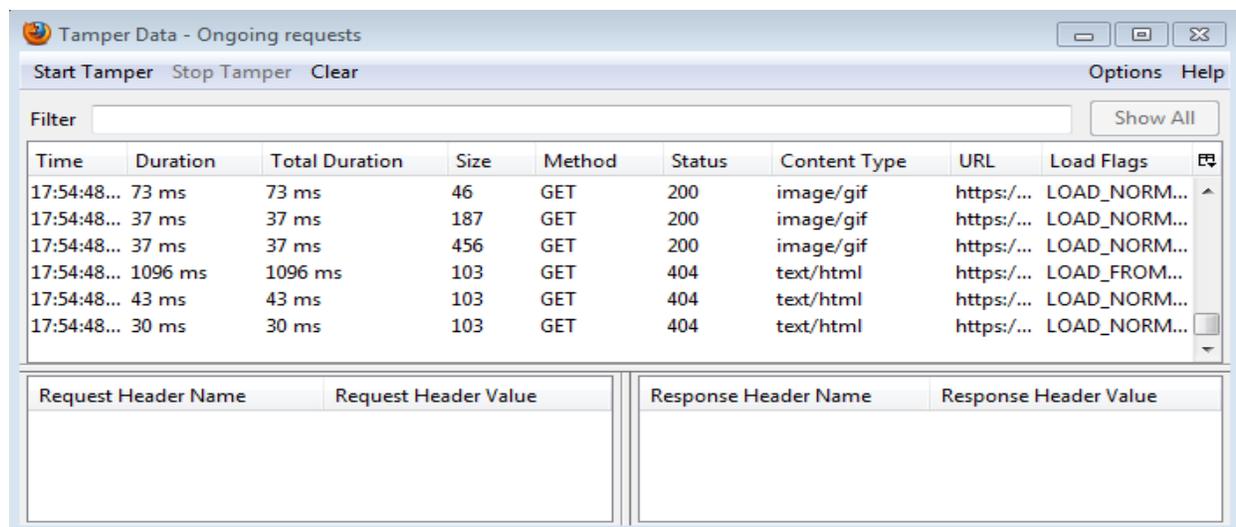
Now once we have restarted the Firefox we can visit the Tamper Data by going to TOOLS → Tamper Data



After clicking the Tamper Data the “Tamper Data - Ongoing Requests” window opens up.



As soon as this window is opened, Tamper data will start reading the HTTP requests. The window will look like this when it starts reading the requests.



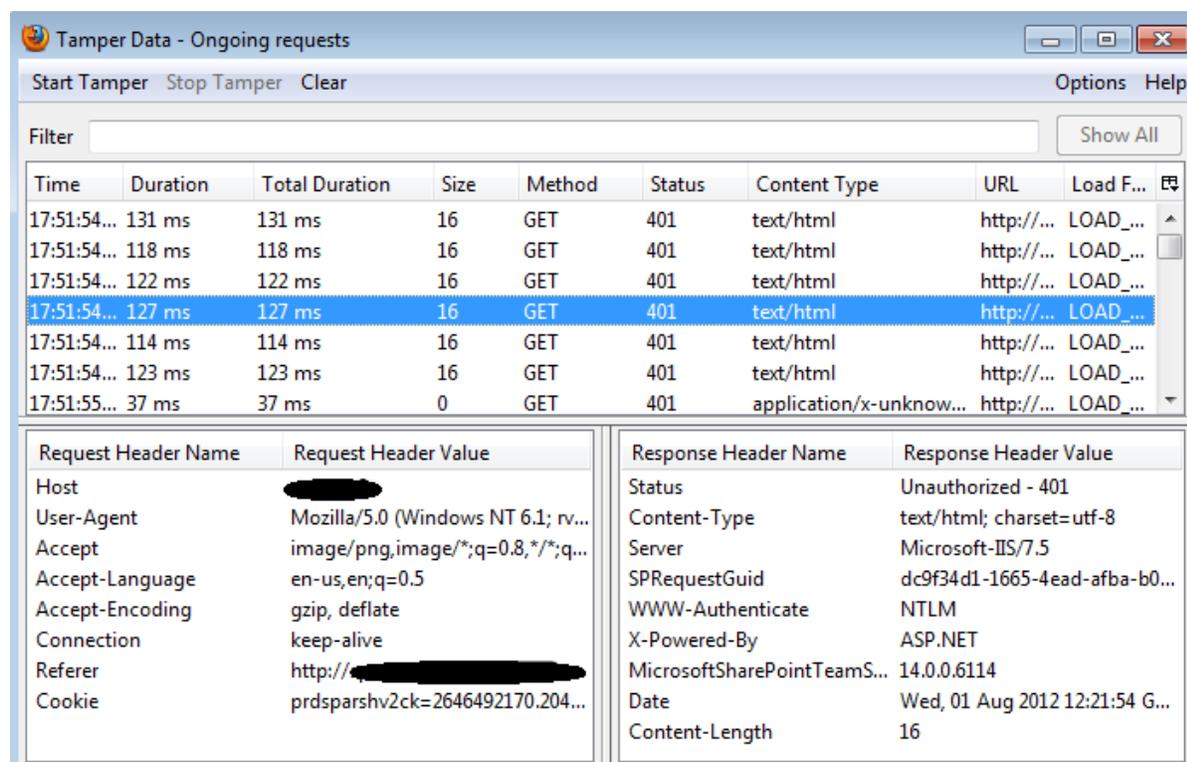
The screenshot shows the 'Tamper Data - Ongoing requests' window. It has a menu bar with 'Start Tamper', 'Stop Tamper', 'Clear', 'Options', and 'Help'. Below the menu is a 'Filter' input field and a 'Show All' button. The main area contains a table with the following columns: Time, Duration, Total Duration, Size, Method, Status, Content Type, URL, and Load Flags. The table lists several requests, with the last one selected.

Time	Duration	Total Duration	Size	Method	Status	Content Type	URL	Load Flags
17:54:48...	73 ms	73 ms	46	GET	200	image/gif	https://...	LOAD_NORM...
17:54:48...	37 ms	37 ms	187	GET	200	image/gif	https://...	LOAD_NORM...
17:54:48...	37 ms	37 ms	456	GET	200	image/gif	https://...	LOAD_NORM...
17:54:48...	1096 ms	1096 ms	103	GET	404	text/html	https://...	LOAD_FROM...
17:54:48...	43 ms	43 ms	103	GET	404	text/html	https://...	LOAD_NORM...
17:54:48...	30 ms	30 ms	103	GET	404	text/html	https://...	LOAD_NORM...

Below the table are two panes for header information:

Request Header Name	Request Header Value	Response Header Name	Response Header Value

On selection of an item, its HTTP Request and Response information's are opened on the lower two left and right panes respectively.



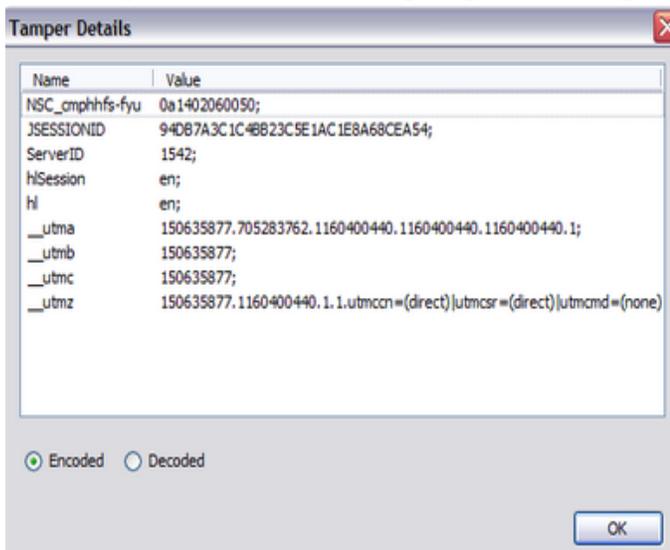
The screenshot shows the 'Tamper Data - Ongoing requests' window with the same table as above. The row with Time '17:51:54...' and Status '401' is selected. The lower panes now show the request and response headers for this request.

Request Header Name	Request Header Value	Response Header Name	Response Header Value
Host	[REDACTED]	Status	Unauthorized - 401
User-Agent	Mozilla/5.0 (Windows NT 6.1; rv...	Content-Type	text/html; charset=utf-8
Accept	image/png,image/*;q=0.8,*/*;q...	Server	Microsoft-IIS/7.5
Accept-Language	en-us,en;q=0.5	SPRequestGuid	dc9f34d1-1665-4ead-afba-b0...
Accept-Encoding	gzip, deflate	WWW-Authenticate	NTLM
Connection	keep-alive	X-Powered-By	ASP.NET
Referer	http://[REDACTED]	MicrosoftSharePointTeamS...	14.0.0.6114
Cookie	prdsparshv2ck=2646492170.204...	Date	Wed, 01 Aug 2012 12:21:54 G...
		Content-Length	16

Here we get a detailed view of what is going on in the request. If the selected request contains cookie information then we will see a cookie line in the left side pane or set cookie line in the right side pane or both.

Now if we double click an entry then the “Tamper Details” window opens up, which provides us easy access to that request element’s data.

Thus using the above process we can easily monitor what is going on during

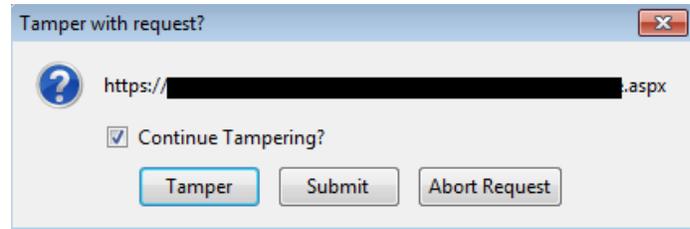


the browsing session.

However till now we have just monitored the request. Tamper data can certainly do more than that. Now comes the sweet part of tampering the requests being made.

To begin we have to click on the option “Start Tamper”.

From now on whenever the browser will make a request we will be prompted with three options...

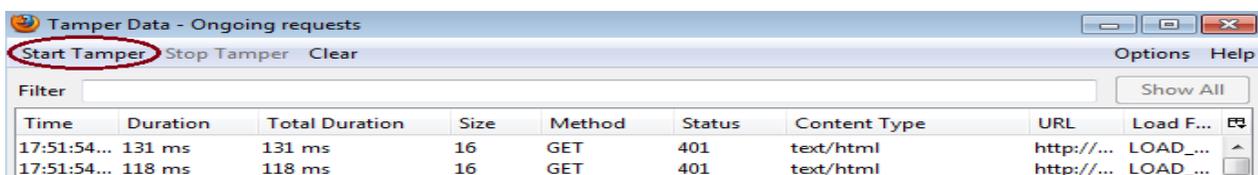


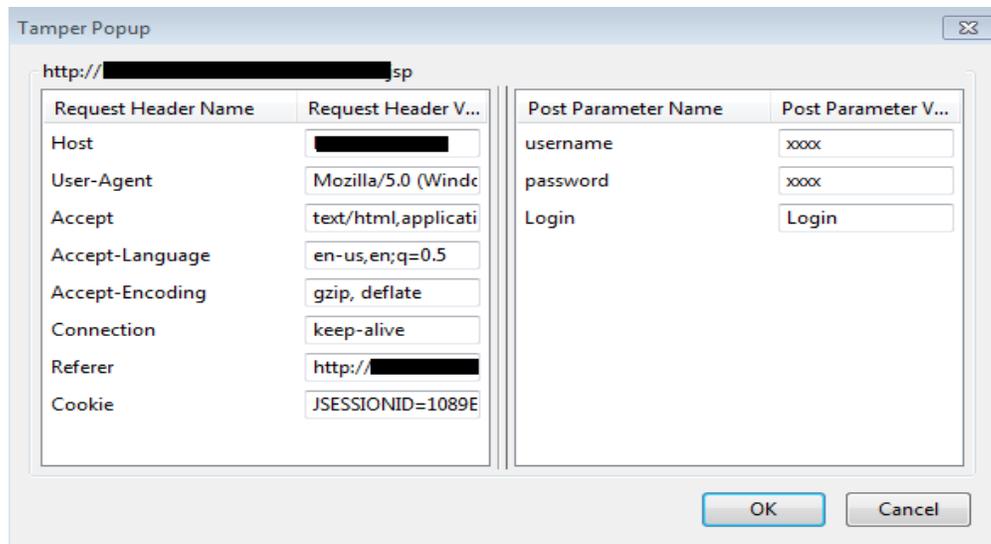
The three options are :

Submit → This just sends the request as it is without modifying it.

Abort Request → This, as the name suggests, will abort the request, i.e. will stop the request from being sent.

Tamper → This is the option which has made Tamper Data so famous and handy as well. When we click on this option, i.e. we want to tamper with the data, then a new window opens up.

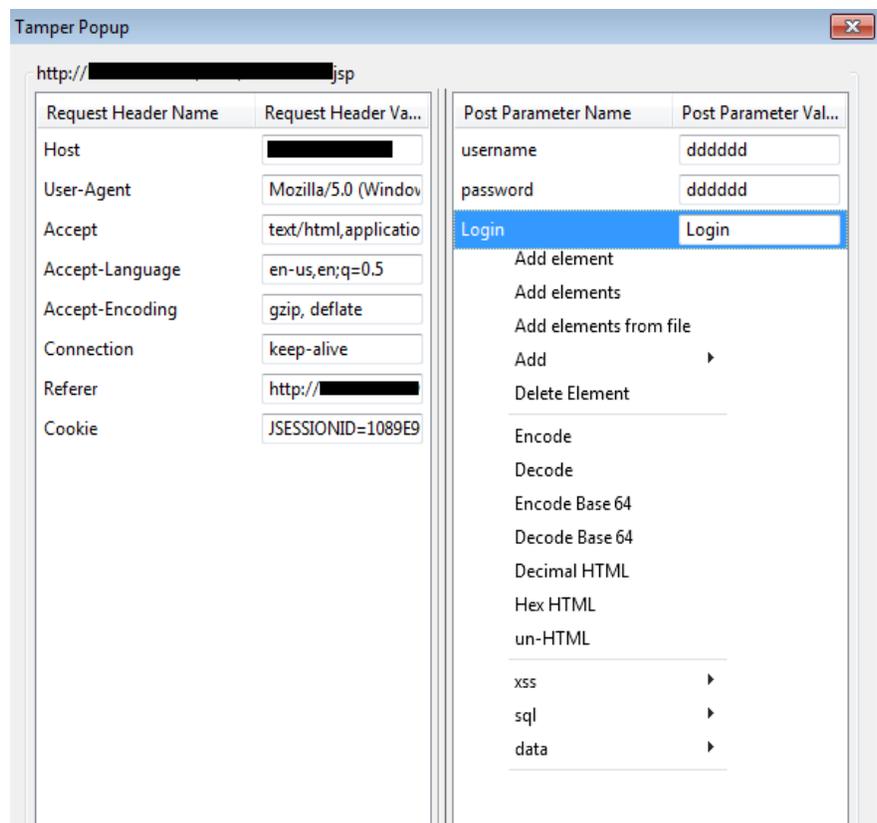




On the top of the window (starting with the HTTP) is the URL the request is being sent to. On the left hand side pane we have the Request headers and their values. We can not only read them but also modify them as per our needs. On the right hand side pane we have the POST data of the request. Here we can see what POST fields are being sent and what are its values. We have a number of options here. On right clicking in the right hand pane we get the option to include our own elements which were not there earlier. Also on right clicking on the elements present we again get a handful number of options. These are very useful in manipulating the form in our own way and wish.

These numerous options saves us the effort of bypassing client side restrictions on what values may be sent or to submit an element not part of the form.

The only limitation of Tamper data is



that it can't modify http GET parameters. Otherwise Tamper Data is a neat tool by which we can easily see what our web application is doing, what are the parameters being passed etc.. With the help of this tool parameter manipulation has become literally a piece of cake.

Thus install this Firefox extension and enjoy the various offerings it has to offer.

Happy Hacking... ☺



Ramesh Chandra Bhattacharjee

Ramesh works for Infosys and is a beginner to information security domain.



Apple iOS vulnerabilities

Introduction

Apple iOS has successfully emerged as one of the most widely used Operating System today. It runs on Apple devices such as iPhone, iPad, iPod touch and Apple TV.

Apple AppStore has the highest number of applications (500,000) with 25 billion apps downloaded till date. However, the iOS developers aren't bothered about the secure aspect of the applications before they launch it on the AppStore. This huge number of apps and carelessness of developers has lured the hackers to steal data from the applications.

There has been numerous exploits exposed on the iOS platform. In this article, three such exploits will be discussed. Please note that these vulnerabilities can be only be exploited if the iPhone is jail broken and apps are installed out of Apple Store.

Cut & Paste Feature

Copy and Paste feature was introduced in iOS 3.0 which involves having a common buffer for all the applications in iOS. This feature can be exploited to steal sensitive data from an application into a malicious application.



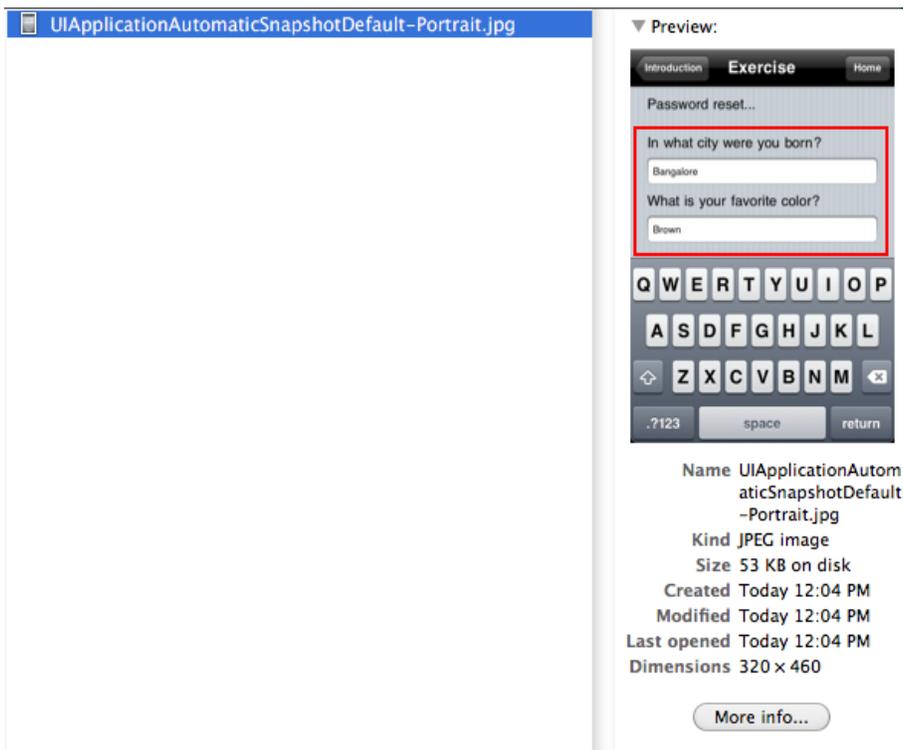
As shown in the above figure, at the top image, the credit card number is copied from the application, and it is stored in the common buffer within the iOS. Below, the malicious application is silently stealing data from the buffer and this application can also remotely send the data to a remote server.

This issue can be remediated by the developer by either clearing the copy paste buffer every time the application exits, or disable the copy paste feature for applications which deal with sensitive data.

iOS Backgrounding

The Apple wanted to provide iOS device users an aesthetically pleasing effect when the application is entered or exited. Hence they introduced the concept of saving the last screenshot when the application goes into the background.

This feature can be exploited as the screenshot which is saved on the device may contain sensitive data like credit card details, Password recovery information etc.



The Fig. shows the user's password reset information stored in the screenshot. This screenshot is stored inside the iOS at the following location.

/private/var/mobile/Library/Caches/Snapshots

This issue can be remediated by the developer by writing a code snippet to clear the contents of the page on application exit.

Auto Correct Feature

Inside the iOS, there exist a file called dynamic-text.data, which is a binary keyboard cache containing ordered phrases of text entered by the user. This text is cached as part of the operating system's autocorrect feature, and may appear from entering text within an application on the device. Think of this as keyboard logger. Hence to avoid writing data to this cache, turn autocorrect off in text fields whose input should remain private, or consider writing your own keyboard class for your application.

This file can be found on the device at the following location.

/private/var/mobile/Library/Keyboard/dynamic-text.dat



Sensitive data can appear in the autocorrect feature as a suggestion

As shown in the Fig-3, the username and passwords which are stored in dynamic text can appear on the screen asking for the user to choose that. The username and passwords will also be saved in the dynamic text file as it is being stored in a dictionary.

References:

- 1) OWASP Mobile Top 10 Risks.
- 2) Hacking & Securing iOS applications by Jonathan Zdziarski.



Anil Pai

anilpai@anilpai.com

Anil Pai is a Mobile App Security Analyst at Tata Consultancy Services with more than 2 years of experience. He is involved with security assessments of Apple iOS and Windows Phone 7 applications.



VARIOUS AUTHORITIES UNDER THE IT ACT

(1) Controller of Certifying Authorities (CCA)

CCA is appointed by the Central Government under *section 17* of the IT Act.

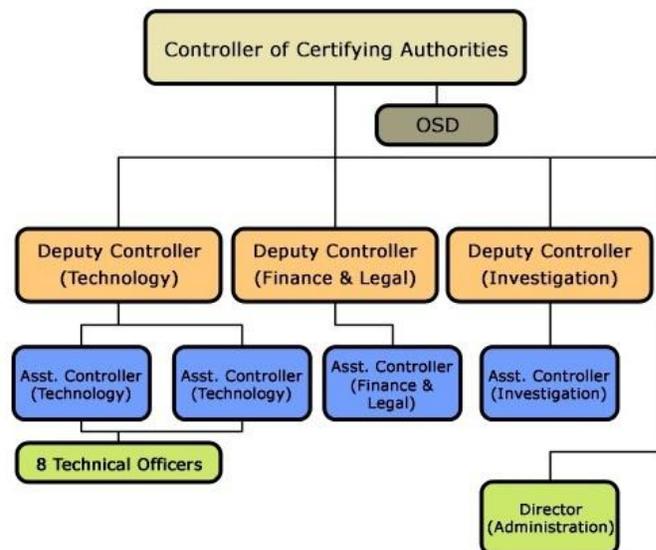
Some of the functions of CCA are -

- Act. To exercise supervision over the activities of Certifying Authorities;
- To supervise public keys of the Certifying Authorities;
- To lay down the standards to be maintained by the Certifying Authorities;
- To specify the qualifications and experience which employees of the Certifying Authorities should possess;

- To specify the conditions subject to which the Certifying Authorities shall conduct their business;

Functions of the CCA are discussed in detail under *section 18* of the IT Act.

Organizational chart of the office of CCA



Controller's power to investigate the contraventions

Section 28 of the Act provides that, the Controller or any officer authorized by him shall have power to investigate contraventions as laid down in the provisions of this Act.

Important sections regarding CCA under the IT Act -

- **Section 17** - Appointment of Controller and other officers
- **Section 18** - Functions of Controller
- **Section 28** - Power to investigate contraventions

Website of the office of CCA - www.cca.gov.in

(2) Certifying Authority

Certifying Authorities (CA) has been granted a license to issue a digital signature certificate under *section 24* of the IT Act.

(3) Adjudicating officer (AO)

AO is appointed under section 46 of the IT Act to adjudicate offences under Chapter IX.

As per Rule 3 of the Information Technology (Qualification and Experience of Adjudicating Officers and Manner of Holding Enquiry) Rules, 2003, it has been declared that the *Secretary of Department of Information Technology of every State and Union Territory shall serve as Adjudicating officer.*

Important sections regarding AO under the IT Act –

- **Section 46** - Power to adjudicate

- **Section 47** - Factors to be taken into account by the adjudicating officer

(4) Cyber Appellate Tribunal (CAT)

Cyber Appellate Tribunal has been established under the IT Act under the aegis of Controller of Certifying Authorities. It is established under Section 48(1) of the IT Act. Any person aggrieved by an order made by Controller or an adjudicating officer under this Act may prefer an appeal to a Cyber Appellate Tribunal.

Chapter X - Sections 48 to 64 of the IT Act has provisions regarding CAT.

Website of the office of CAT - <http://www.catindia.gov.in>

(5) Indian Computer Emergency Response Team (ICERT)

ICERT is the National Incident Response Centre for major computer security incidents in its constituency, i.e. Indian Cyber Community.

Under section 70B of the IT Act, ICERT has been empowered to serve as national agency for incident response.

Website of the office of CERT-In - <http://www.cert-in.org.in/>

(6) National Technical Research Organisation (NTRO)

NTRO is designated as the national nodal agency in respect of Critical Information Infrastructure Protection under Sec. 70A of the IT Act.

(7) Cyber Crime Cell

Cyber Crime Cell is a wing of law enforcement agencies like Police, CID, CBI, etc. established to expedite the investigation of Cyber Crimes.

Kindly note that, Cyber Crime Cell is not a Police station where one can go and register a complaint.

In India, Bangalore is the only city which has a Cyber Crime Police Station where one can register a complaint and can get a copy of the First Investigation Report (FIR).

Some of the duties of Cyber Crime Cell are –

- To assist law enforcement in investigating cyber crimes
- To spread awareness about cyber crimes and preventive measures in its territory
- To act as an expert in giving opinions on cyber crime related issues

Power to investigate offences under the IT Act is with the police officer not below the rank of Inspector as per Sec. 78 of the IT Act.



Sagar Rahrurkar

contact@sagarahrurkar.com

Sagar Rahrurkar is a Law graduate. He is a techno-legal consultant and a Senior Faculty at Asian School of Cyber Laws.

He specializes in Cyber Law, Cyber Crime Investigation, Computer Forensics and Intellectual Property Laws.

He teaches and provides consultancy to corporates, law enforcement agencies and education institutes across India.

He can be contacted at contact@sagarahrurkar.com.



Matriux Ec-Centric

Hello every readers,

Matriux is been successfully running and getting a big support over the past 2 years and we have been working hard to provide the best security solutions and quality tools for all the penetration testing and forensic needs. On occasion this issue is to be released at cocon 2012, we are also proud to announce our upcoming release Matriux version 2.0 Kod3 name “Ec-Centric”. We have been working hard over the release of the new version, so this month Matriux Vibhag will feature the Matriux Ec-Centric edition.

Features

- The “Arsenal” now includes around 325 powerful penetration testing and forensic tools and this time we included webshells making it the true arsenal for testing and cyber forensics.
- Also including the latest of tools and applications released at BlackHat 2012 US being the first distribution to include.
- Based on Debian Squeeze featuring the latest kernel 3.3.4.
- Custom compiled Kernel for high support for your hardware drivers along with squashfs and aufs modules.
- Lighter, Elegant and Faster UI with Gnome.
- Also making it more fast and easy to use.
- MID 0.3b to make the installation easier.
- Tools from Matriux Community.
- And the best part – runs easily on your 10 year old computer with p-IV and 256MB RAM with 6GB of HDD space.

```

/bin/bash (as superuser)
homophones.rb      inflector.rb      report-google.com.txt  urlcrazy
root@matriux:/pt/scanning/urlcrazy# cd ..
root@matriux:/pt/scanning# cd webscanners/
root@matriux:/pt/scanning/webscanners# ls
digdug  gggooglescan  icmpquery  JHijack  minimysqlat0r  wafp  Xssploit
root@matriux:/pt/scanning/webscanners# ls -la
.  ..  digdug  gggooglescan  icmpquery  JHijack  minimysqlat0r  wafp  Xssploit
root@matriux:/pt/scanning/webscanners# cd r
root@matriux:/pt/scanning/webscanners/minir#
bash: s: command not found
root@matriux:/pt/scanning/webscanners/minir#
common_tables.txt  mms.jar
root@matriux:/pt/scanning/webscanners/minir#
root@matriux:/pt/scanning/webscanners# cd v
root@matriux:/pt/scanning/webscanners/wafp#
CREDITS  HOWTO  LICENSE  scan_wafp.
fprints_wafp.db  lib  README  utils
root@matriux:/pt/scanning/webscanners/wafp#
No LSB modules are available.
Distributor ID: Matriux
Description:    Matriux Ec-Centric
Release:       2.04b

```

Menu items:

- Arsenal
- Accessories
- Internet
- Programming
- Sound & Video
- System Tools
- Reconnaissance
- Scanning
- Gain Access
- Frame Work
- DNS
- HTTrack
- chaosreader
- dmitry
- Dradis Framework
- dsniff
- Etherape
- Etherape (as root)
- fragroute
- peepdf
- quickrecon
- tcpdump
- tcpslice
- tcptrace
- tcptraceroute
- wireshark
- xtrace

Notable Updates

- More tools for VOIP and Forensics.
- Emphasis on Mobile forensics and malware analysis through security applications for Android and iPhone.
- Shell scripts.
- Build Update tool called “MUT”.
- And many more to check out at cocon 2012, be there as we will be there ;)



Team Matriux

<http://matriux.com/>

Happy Hacking and from next month we will continue with our regular tutorials on various security tools.

Get the best real-world Android training anywhere!



Attend

AnDevCon IV

The Android Developer Conference

December 4-7, 2012
San Francisco Bay Area

Choose from more than 65 classes and workshops!



AnDevCon is the biggest, most info-packed, most practical Android conference in the world!

- Learn from the top Android experts, including speakers straight from Google!
- Attend sessions that cover app development, deployment, management, design and more
- Network and connect with hundreds of experienced developers and engineers like yourself

"AnDevCon is a fantastic conference! There is no better place to experience the latest and greatest technologies and techniques in the field of Android development. If you attend one conference this year, this one should be it!"

—Jay Dellinger, Senior Software Engineer, Manheim

Register Early
and SAVE BIG!

www.AnDevCon.com

Follow us: twitter.com/AnDevCon

A BZ Media Event

AnDevCon™ is a trademark of BZ Media LLC. Android™ is a trademark of Google Inc. Google's Android Robot is used under terms of the Creative Commons 3.0 Attribution License.



Security by luck, not possible!

Design : @pankit_thakkar