

# AVG Community Powered Threat Report



Q4 2011

## Contents

Introduction .....	3
Key Points – Q4 2011 .....	4
Web Threats – First round for free, second round with a threat .....	4
PC Threats – Rootkits are getting smarter and smarter .....	4
Mobile Threats – Stolen digital certificates in the Mobile malware world are now a reality, as they are in the PC world .....	4
Printed malware – QR symbols are becoming popular for mobile users to insert text and URLs into the mobile device without typing, malware included..	4
Metrics - Web Threats .....	5
Top 10 Web Threats Prevalence Table Q4 2011.....	5
Top 10 Malware Threats Prevalence Table Q4 2011 .....	6
Behavior Categories Chart Q4 2011.....	6
Top Exploit Toolkits Seen in Q4 2011.....	7
Distribution of Android Threats - Q4 2011 .....	7
Metrics - Email Threats .....	8
Top Domains in Spam Messages Q4 2011      Top 5 Languages in Spam Messages Q4 2011 .....	8
Top Countries of Spam Senders Q4 2011 .....	8
Web Risks & Threats .....	9
Second-Click Redirect Mechanism.....	9
Behind the Scenes.....	11
Prevalence of the Second-Click Mechanism .....	11
Rootkits .....	13
About Rootkits .....	13
Evolution .....	13
ZeroAccess .....	13
Malicious Activity .....	15
Recommendations .....	15
Mobile Devices Risks & Threats .....	16
Trusted Malware? .....	16
Background .....	16
Case #1: Using a Fake Google Certificate – if you can't have it, fake it .....	17
Case #2: Lorenz Leaked Certificate Case – Don't leave the (private) key out there, unprotected.....	20
Case #3: Using AOSP certificate by malware – a home run.....	21
Recommendations .....	23
Printed Malware .....	24
Overview .....	24
QR codes as URL shortening are not 100% risk-free .....	24
QR codes serving malware - Example .....	24
Try it yourself .....	25
Other reports from AVG Technologies .....	26
AVG and Ponemon Institute: 'Smartphone Security - Survey of U.S. consumers' .....	26
Anatomy of a major Blackhole attack.....	26
AVG Community Powered Threat Report Q1 2011 .....	26
AVG Community Powered Threat Report Q2 2011 .....	26
AVG and Future Laboratories: 'Cybercrime Futures' .....	26
AVG and GfK: 'AVG SMB Market Landscape Report 2011' .....	26
AVG Community Powered Threat Report Q3 2011 .....	26
About AVG Technologies .....	26



## Introduction

The AVG Community Powered Threat Report is based on the Community Protection Network traffic and data, collected over a three-month period, analyzed by AVG Threat Labs. It provides an overview of web, mobile devices, Spam risks and threats. The statistics referenced are obtained from the AVG Community Protection Network.

AVG Community Protection Network is an online neighborhood watch, helping everyone in the community to protect each other. Information about the latest threats is collected from customers who choose to participate in the product improvement program and shared with the community to make sure everyone receives the best possible protection.

AVG has focused on building communities that help millions of online participants support each other on computer security issues and actively contribute to AVG's research efforts.

## Q4 2011 Highlights

Web Threats	
<a href="#">Blackhole Exploit Kit</a>	The most active threat on the Web, <b>47.6%</b> of detected malware
<a href="#">Blackhole</a>	The most prevalent exploit toolkit in the wild, accounts for 80.2% of toolkits
<b>58%</b>	Exploit toolkits account for 58% of all threat activity on malicious websites
<b>12.08%</b>	Of malware are using external hardware devices (e.g. flash drives) as a distribution method (AutoRun)
Mobile Threats	
<a href="#">com.depositmobi</a>	The most popular malicious Android application
<b>~1 Million</b>	Malicious events detected during Q4 2011
Messaging Threats (Spam)	
<a href="#">United States</a>	Is the top Spam source country
<b>45%</b>	Of Spam messages originated from the USA followed by the UK with 9.1%
<a href="#">Facebook.com</a>	Top domain in Spam messages
<a href="#">English</a>	Is the top language used in Spam messages (70.1%)

## Key Points – Q4 2011

Free offerings from cyber criminals are getting smarter and also available on mobile platforms. This is how we summarize the recent trends in the malware world during Q4 2011. Web, PC and mobile threats are sharing similar techniques, becoming smarter and always have a monetization method behind them, even if they are initially offered for free.

Malware targeting mobile devices evolves frighteningly fast and the magnitude has the potential of being even more destructive than before. At the end of 2010, numbers already indicating that new mobile devices were overtaking new purchased PCs. The numbers are very impressive; vendors shipped 100.9 million smart phones during the fourth quarter of 2010, while IDC logged 92.1 million PC shipments during the same time period<sup>1</sup>. Smart phones surpassed PC shipments much faster than expected.

While consumers are going mobile, so are the cyber criminals. We have witnessed the use of the same malicious intent tactics targeting mobile devices: social engineering, stolen or fake certificates to sign malware, rootkits and other tactics.

The main stories spotted by AVG Threat Labs during Q4/2011:

### Web Threats – First round for free, second round with a threat

“Old habits die hard”: cyber criminals are up to their old tricks, apparently the use of fake antivirus product is still successful. The only difference is the infection method. In this report, we cover an infection method called ‘2<sup>nd</sup> click redirection mechanism’ which eventually redirects to a Fake AV scanner (Rogue AV) page that tries to lure users into downloading and paying for an AV scanner which “removes” fictitious malware. Another important aspect of this story is showing that the ‘under world’ of cyber crimes is organized. Malicious websites do not only share traffic, they also share owners.

### PC Threats – Rootkits are getting smarter and smarter

If you think that rootkits are history, think again. Rootkits are alive and kicking. Rootkits are evolving to be much more sophisticated, and some interesting samples show up every few months. Rootkits evolved from commercial use (SONY DRM<sup>2</sup>) through to financial use (Greek wiretapping case<sup>3</sup>) to cyber warfare with a very specific target (Stuxnet, Duqu<sup>4</sup>). Currently we are witnessing the first phase of rootkits evolution on mobile devices (CarrierIQ<sup>5</sup>). We anticipate that the mobile rootkit evolution will be similar to the PC rootkit’s evolution.

### Mobile Threats – Stolen digital certificates in the Mobile malware world are now a reality, as they are in the PC world

During 2011, we often reported on the rapid growth of malware targeting Android devices (see AVG’s [Q2 threat report](#)); we presented various examples of malicious code and infection methods. This trend continues to grow, against a backdrop of enormous growth of activated Android devices in the past 6 months, from 100 Million devices (May 2011) to 200 million devices (Nov 2011) and over 550,000 activations daily<sup>6</sup>:



Figure 1 - Android Activated Devices (Source: [engadget.com](#))

In this report, we will cover how malware targeting mobile devices is following the steps of its ancestor, PC malware. Malware writers are using the same extremely sophisticated techniques when targeting mobile devices. In the last year we have seen an increase of using legitimate certificates, issued by certificate authorities, to sign malware. This has already become a serious problem on the PC world. In this report we will demonstrate that this problem is going to be much more significant in the smart phone market, especially Android based devices.

### Printed malware – QR symbols are becoming popular for mobile users to insert text and URLs into the mobile device without typing, malware included.

In this report we will review a new technique used by hackers to mislead users to scan QR codes that download malware into their mobile devices. This new technique is expected to gain momentum in 2012 and beyond, as the user does not know what lies behind the QR code until the malware is already installed and running.

<sup>1</sup> <http://www.pcmag.com/article2/0,2817,2379665,00.asp>

<sup>2</sup> [http://www.businessweek.com/technology/content/nov2005/tc20051117\\_444162.htm](http://www.businessweek.com/technology/content/nov2005/tc20051117_444162.htm)

<sup>3</sup> [https://www.pcworld.com/article/134398/greek\\_spying\\_case\\_uncovers\\_first\\_phone\\_switch\\_rootkit.html](https://www.pcworld.com/article/134398/greek_spying_case_uncovers_first_phone_switch_rootkit.html)

<sup>4</sup> [https://www.computerworld.com/s/article/9185919/Is\\_Stuxnet\\_the\\_best\\_malware\\_ever](https://www.computerworld.com/s/article/9185919/Is_Stuxnet_the_best_malware_ever)

<sup>5</sup> <http://www.guardian.co.uk/technology/2011/dec/15/carrier-iq-faces-us-probe>

<sup>6</sup> <http://www.engadget.com/2011/11/16/google-200-million-android-devices-activated-over-550-000-acti/>



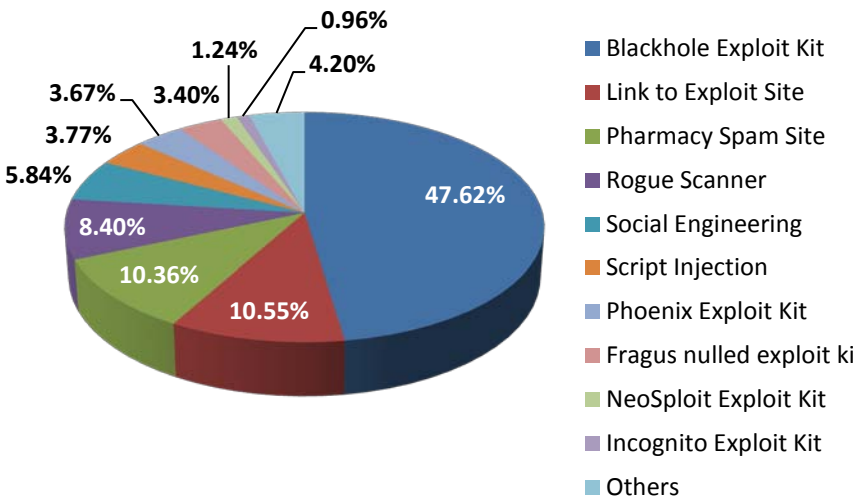
Metrics - Web Threats

Top 10 Web Threats Prevalence Table Q4 2011

This prevalence table shows top web threats as reported by the AVG community regarding Web Threats

Blackhole Exploit Kit	Pages containing script code characteristics of the Blackhole exploit kit, which is used to install a range of malware
Link to Exploit Site	These pages contain links to known exploit sites. In some cases, malicious code is automatically downloaded without any user intervention
Pharmacy Spam Site	The Pharmacy Spam sites appear to be legitimate online pharmacies, but usually are facsimiles of real sites. These fake pharmacies often supply generic, or even fake, drugs rather than the brands advertised, and reportedly often deliver no drugs at all
Rogue Scanner	Pages containing fake virus scanners, or appear to be pages pushing fake antivirus products. Such pages intend either (or both) to lure end user to buy worthless software, or to install malware under the cover of seemingly useful software
Social Engineering	These pages contain code/information which tries to lure people into downloading malicious code
Script Injection	Injection of code by an attacker, into a computer program to change the course of execution
Phoenix Exploit Kit	Exploit toolkit which is used to install a range of malware
Fragus nulled exploitkit	Exploit toolkit which is used to install a range of malware
NeoSploit Exploit kit	Exploit toolkit which is used to install a range of malware
Incognito Exploit kit	Exploit toolkit which is used to install a range of malware

Top 10 Web Threats Prevalence Chart Q4 2011



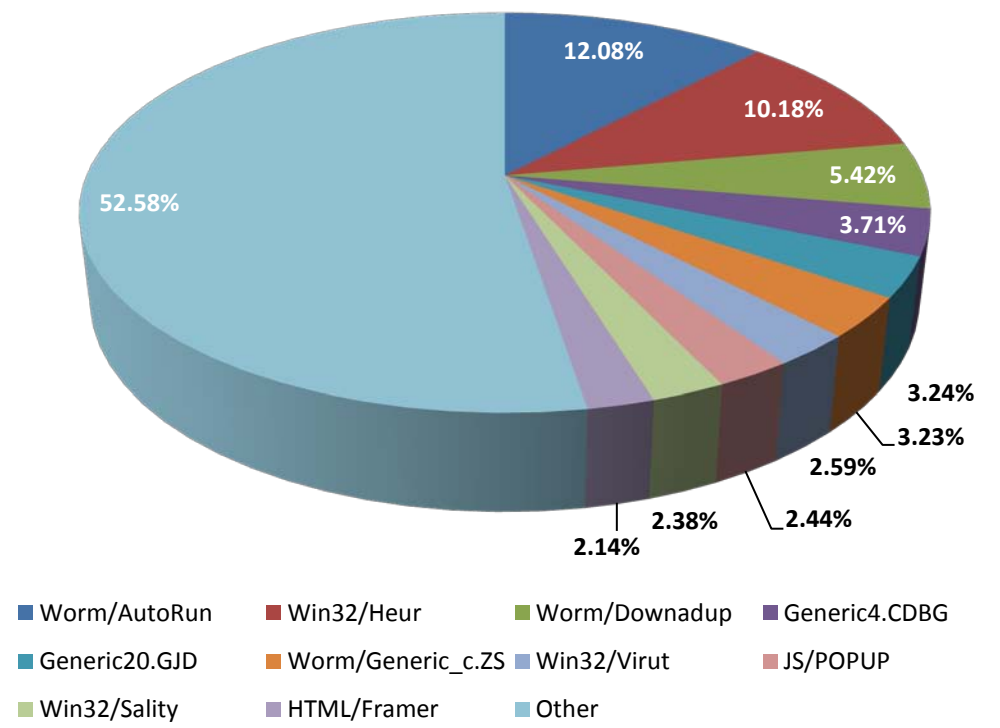


### Top 10 Malware Threats Prevalence Table Q4 2011

This table presents top traditional malware as detected by AVG Threat Labs

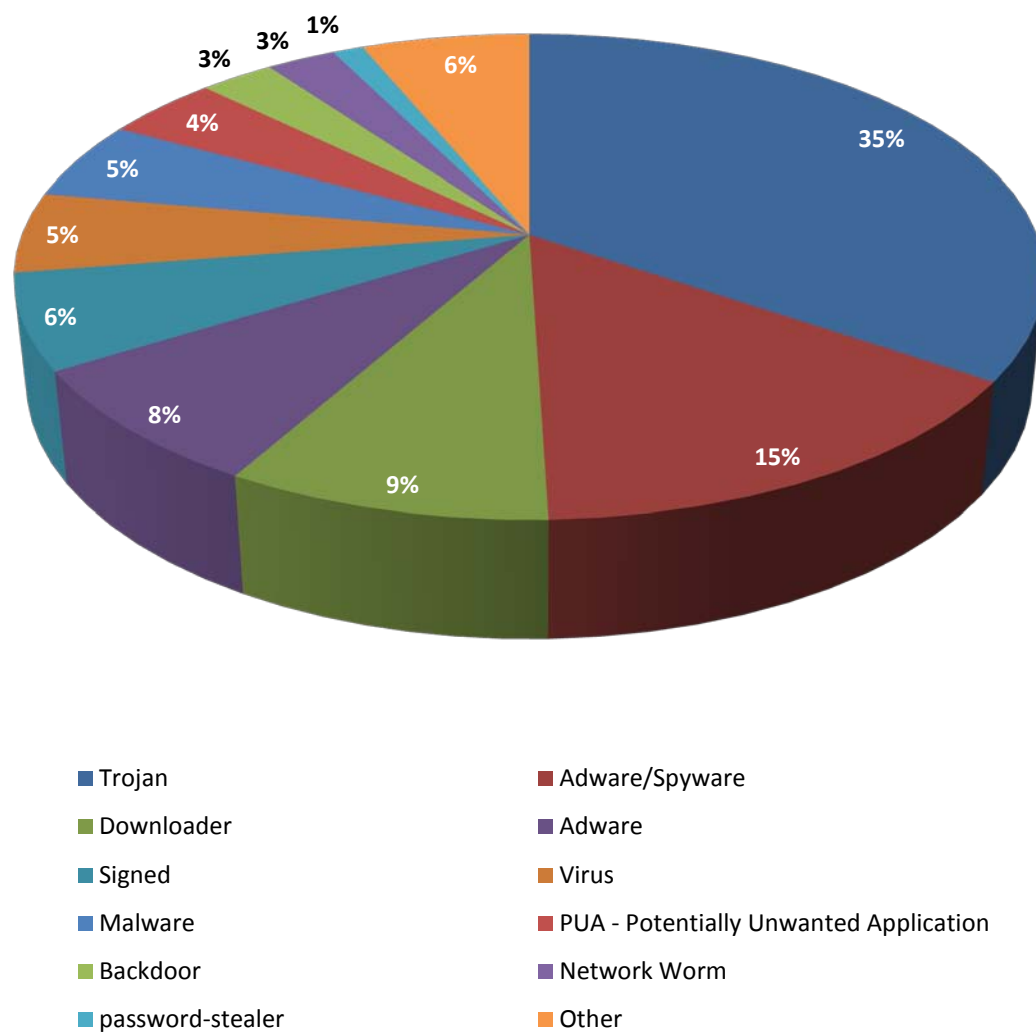
Worm/AutoRun	12.08%
Win32/Heur	10.18%
Worm/Downadup	5.42%
Generic4.CDBG	3.71%
Generic20.GJD	3.24%
Worm/Generic_c.ZS	3.23%
Win32/Virut	2.59%
JS/POPU	2.44%
Win32/Sality	2.38%
HTML/Framer	2.14%

### Top 10 Malware Prevalence Chart Q4 2011



### Behavior Categories Chart Q4 2011

This table presents threats prevalence as detected by AVG's Identity Protection engine. This patent-pending technology looks at what the software does during execution. Using various classifiers and advanced algorithms, this technology determines the hostile behavior of files and prevents their execution





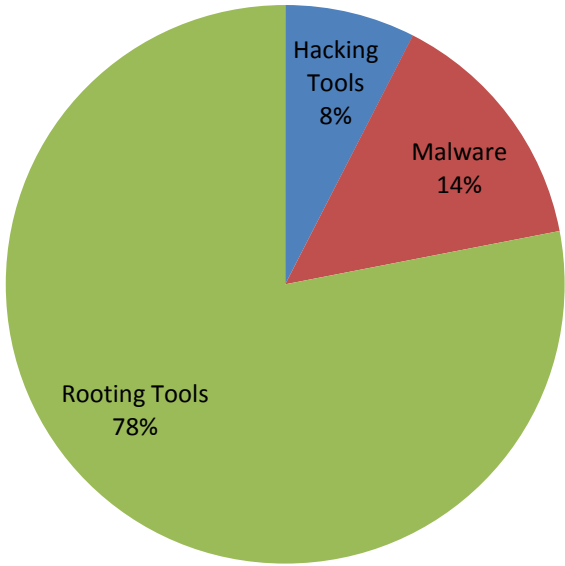
Top Exploit Toolkits Seen in Q4 2011

These metrics present the top five exploit toolkits in terms of malicious web activities. Criminals are increasingly utilizing toolkits to carry out cyber attacks. In many cases, using these attack toolkits does not require technical expertise

1	Blackhole	80.2%
2	Fragus	10.8%
3	Seosploit	4.2%
4	Neosploit	4.0%
5	Bleeding Life	0.5%
6	Others	0.3%

Distribution of Android Threats - Q4 2011

Mobile Threats Q4 2011
















Metrics - Email Threats






Top Domains in Spam Messages Q4 2011

Top domains used in Spam messages

1		Facebook.com	7.3%
2		twitter.com	4.2%
3		gmail.com	3.1%
4		akamai.net	2.7
5		yahoo.com	2.4%
6		hotmail.com	2.4%
7		chta.com	1.9%
8		linkedin.com	1.5%
9		amazonaws.com	1.5%











Top 5 Languages in Spam Messages Q4 2011

Top languages used in global Spam messages

1		English	70.1%
2		Portuguese	6.9%
3		French	3.6%
4		German	2.8%
5		Dutch	2.7%

Top Countries of Spam Senders Q4 2011

Top Spam source countries

1		United States	45.5%
2		United Kingdom	9.1%
3		Germany	5.4%
4		France	4.5%
5		Brazil	4.5%
6		Netherlands	3.0%
7		Canada	2.7%
8		Australia	2.1%
9		Italy	1.6%
10		South Africa	1.5%



## Web Risks & Threats

### Second-Click Redirect Mechanism

First round comes free, second round comes with a threat.

The AVG Threat Labs have been monitoring a second-click redirect mechanism that delivers rogue security products. The redirect technique can be seen on many dubious websites. In Q4 2011, AVG's LinkScanner has blocked almost 8 millions attempts to redirect visitors to an installation of rogue antivirus software (Fake AV) and exploit kits, using this technique. The 8 millions attempts were detected on ~1700 domains, which also were using a specific rotator script to share traffic with other websites.

The use of a multi-click rotation script and traffic sharing between partner's websites becomes common these days; this method is not necessarily being used to spread malware, it can be used for legitimate purposes as well. A user can go to a specific site, click on an image and will be redirected to another site (which offers pay-per-click revenue, for example) on the first click on the images. In case the user is persistent enough to hit the browser "back" button and click the original thumbnail picture or a completely different thumbnail, he will be redirected to another website that has partnered with the original site to exchange traffic.

The malicious use of a multi-click rotation script is done as follows. There are many sites on the internet that simply present a large number of thumbnail pictures that lead visitors to other sites to see free movies, photos or offer many other related services. When a user clicks on a thumbnail photo, he is redirected to another website as expected, however, when returning to the original site and selecting a second thumbnail photo (could be the same one), he will be redirected to a site that tries to install Fake Antivirus software.

The behavior of a Fake antivirus is well known already:

Fake AV claims to 'scans' the user's machine and presents fake detection results at the end of the scan, the user is immediately offered a "free" removal tool (figure 2).

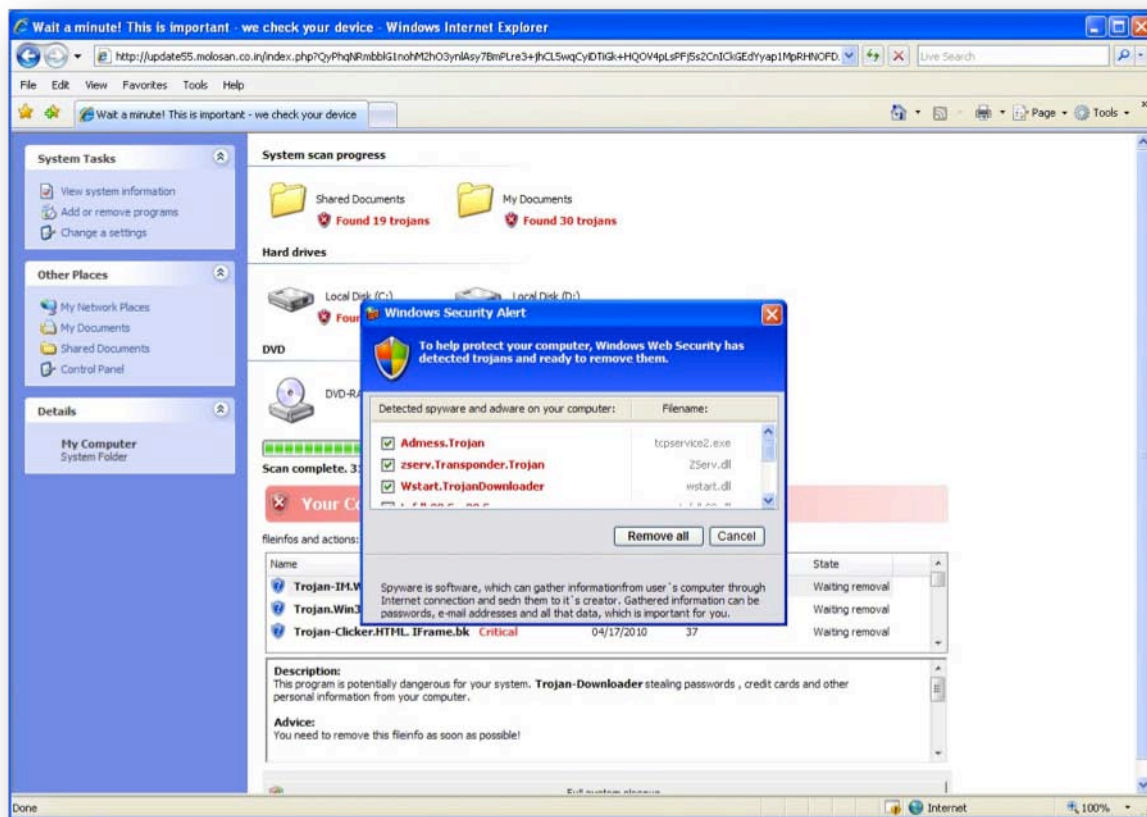


Figure 2 - Fake Detections

Following that, the installation of the Fake AV will initiate (Figure 3).

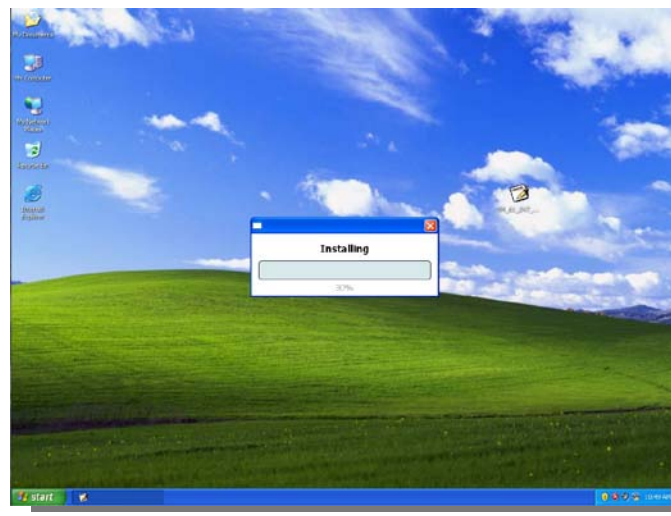


Figure 3 - installation Process

Following the installation, the Fake AV claims to have detected additional malware (Figure 4,5)

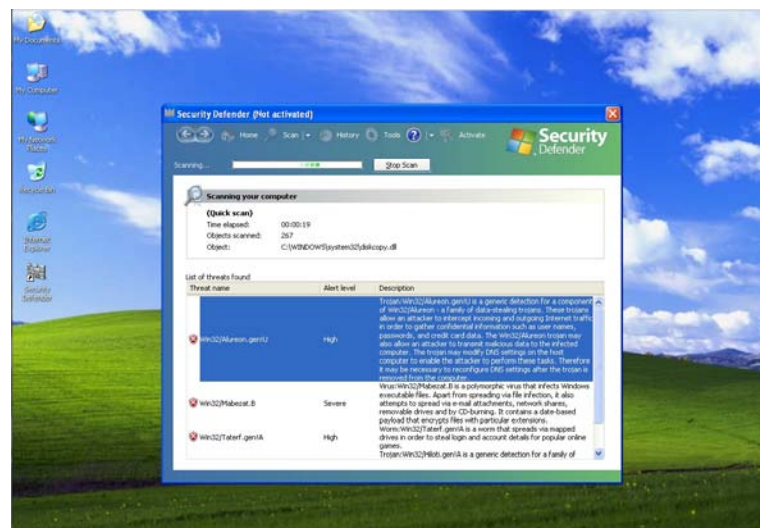


Figure 4 - Additional "Malware" Detected

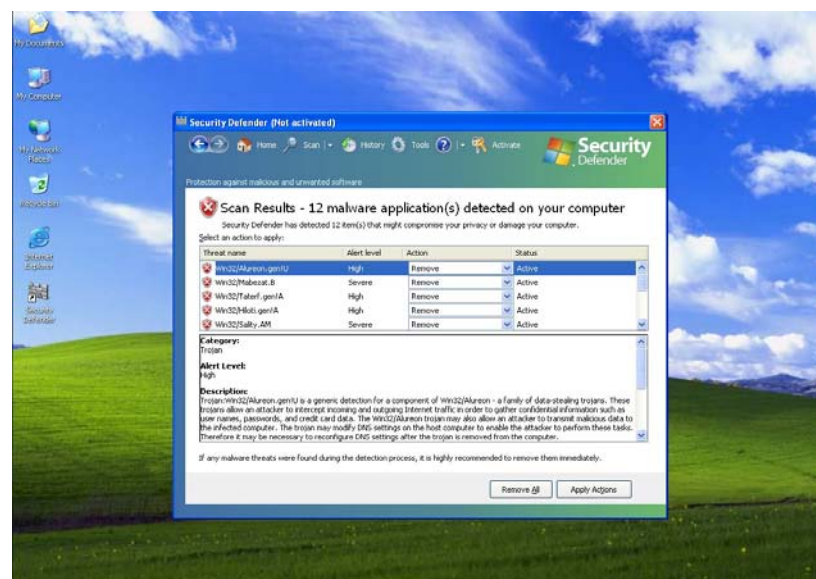


Figure 5 - Additional "malware"



At this point, in order to remove the threats, the user is requested to buy a license of the “security software” which obviously does not remove any “non-existent threats” (Figure 6). This is the most common monetization technique hackers are using on the web.

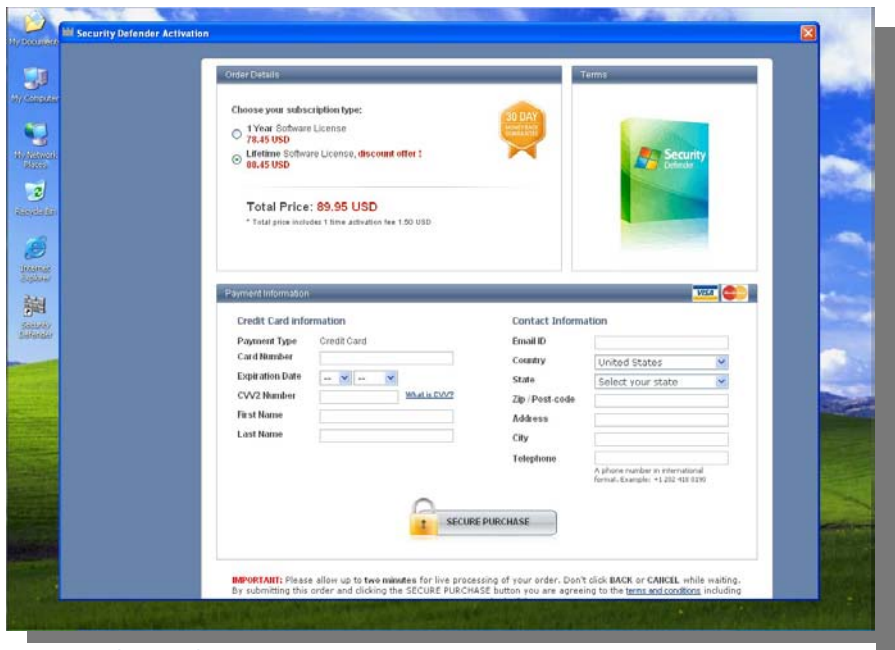


Figure 6 - The Punch Line

Behind the Scenes

The second-click logic is controlled by a cookie saved on user’s machine, and server-based PHP code. Each of the thumbnails on these sites is linked to a server-side PHP script with a long list of arguments, including what sites to redirect to. The destination site is usually, though not always, related to the thumbnail image.

The first time the PHP script is called (on the first click), the PHP code sets a cookie’s time stamp (for example: tstate="1322768477.0"), the timestamp is followed by a zero (decimal point) and sends the browser to the actual requested website.

On the second call, when the user clicks on a thumbnail, the PHP code checks the cookie time stamp to see whether the user already visited the first time, it then adds the current time (tstate= "1322768477.1322768612") and redirects to a malicious website that serves a Fake AV.

Once the user’s machine has the full tstat cookie set, it is no longer redirected to a malicious site. **Users only get redirected to the fake scan page the second time they click a thumbnail.** Therefore disabling cookies protects the user, since tstat is never set.

Prevalence of the Second-Click Mechanism

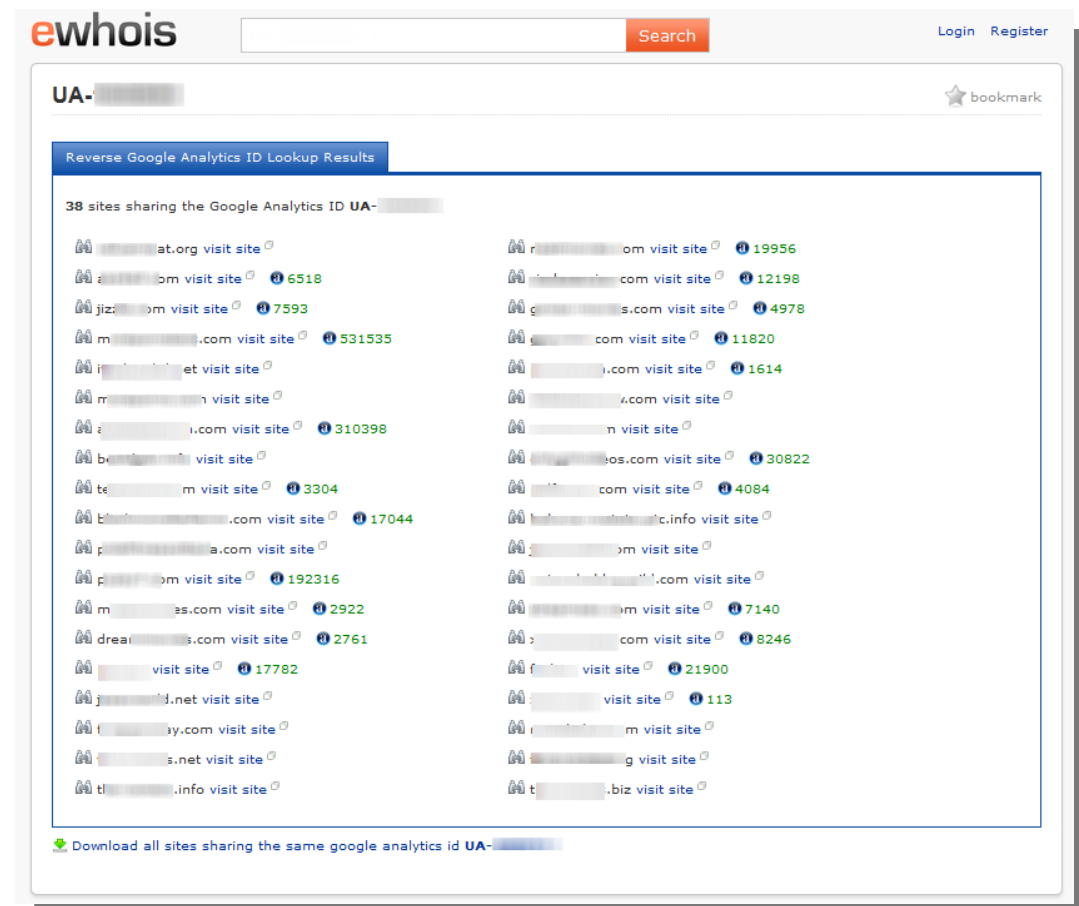
Although we found the second-click mechanism on ~8 million pages, the majority of hits came from a much smaller number of domains (~1700). Below is a list of the top 25 sites that were reported by AVG community members (figure 7)

http://www.magies.com/	238,087
http://www.dreamies.com/	231,317
http://www.teens.com/	171,076
http://www.strea.com/	169,322
http://www.gonzies.com/	167,451
http://www.milfi.com/	142,881
http://www.xnxx.es.com/	135,623
http://www.tightts.com/	89,495
http://www.bunnies.com/	75,883
http://www.jizzle.com/	73,441
http://www.ass4.com/	73,175
http://www.home-videos.com/	65,109
http://www.mon.com/	62,521
http://www.gigam.com/	59,524
http://www.porn.com/	55,631
http://www.atkm.com/	54,686
http://www.foox.com/	50,402
http://www.blackterterror.com/	50,216
http://www.neat.com/	48,288
http://www.18to.com/	47,158
http://www.onlylees.com/	45,178
http://www.bada.com/	44,695
http://www.hairy.com/	43,841
http://www.cind.com/	43,414
http://www.sweetegirls.com/	42,820

Figure 7 - Top Leading Sites Using the 2nd Click Mechanism

We have discovered that these sites (figure 7), do not only share traffic or redirect visitors to malicious sites, **they also share the same owners!** Many of these sites are connected in such a way that their owner is monitoring statistics using the same Google analytics account.

Even though the websites' owner's contact details are obviously hidden (the majority of them are hidden but not all), we found that they are connected.



### Figure 8 - List of Sites Being Monitored by the Same Owner

Figure 8 shows a list of sites being monitored by the same owner using the same Analytics account. **Some of them are highly ranked by Alexa.**

## Recommendations

- Be careful out there. Pay attention where you surf on the web, especially when dubious websites are concerned.
- Keep your Anti Malware software enabled and up-to-date

Scammers are out there, be aware that “All that GLITTERS isn’t gold”, not all “security software” is really security software.



## Rootkits

### About Rootkits<sup>7</sup>

A rootkit is a piece of malware that allows privileged access (root level access) to a computer while actively hiding its presence from administrators. It can do this by subverting standard operating system functionality or other applications.

Most rootkits are classified as malware, because the payloads they are bundled with are malicious.

A Cyber Criminal installs a rootkit on a computer after first obtaining root-level access, either by exploiting a known vulnerability or by obtaining a password. Once a rootkit is installed, it allows the cyber criminal to mask the ongoing intrusion and maintain privileged access to the computer.

Rootkit detection is challenging because rootkits may be able to subvert the software that is intended to find it. Detection methods include using an alternative and trusted operating system; behavioral-based methods; signature scanning; difference scanning and memory dump analysis. The removal process can be complicated, particularly in cases where the rootkit resides in the kernel.

### Evolution

Malicious Rootkits started to appear around 1999 and still represent a big challenge for security vendors. Today's rootkits evolved to be much more sophisticated and some interesting samples show up every few months. While some of them start and end as proofs of concepts, many of them are used by cyber criminals, and we can see them 'in the wild' – usually protecting something really nasty.

Rootkits raised public's awareness with the infamous Sony DRM scandal back in 2005 through to the Greek wiretapping case (2004–2005) and with the publicity of Stuxnet and Duqu, which were used in Cyber-warfare.

Rootkits evolved from commercial use through a financial use to cyber warfare with a very specific target.

In this report, we will cover one of the most interesting samples recently detected, ZeroAccess (aka ZAccess).

### ZeroAccess

ZeroAccess is a family of Kernel mode rootkits which targeting windows based machines. ZeroAccess is very sophisticated, very effective and uses advanced anti-forensic features. The main purpose of these rootkits is to provide a platform for distribution of other malware similar to Win32/Alureon rootkit in the past (fake antivirus or custom key-logger components, for example). Each rootkit is aimed to be stealthy, undetectable and, of course irremovable. Zero Access rootkit use undocumented system features, low-level API calls and altogether, it could be considered as state-of- art malware. Let's have a look at it.

This piece of malware is a kernel mode rootkit, which is designed to hide itself, as well as eventually load custom kernel modules into kernel memory and execute them. PE (Portable executable) file infection has been observed as well, in order to increase survivability on an infected computer. The infection itself consists of several phases:

1. The Dropper – the dropper of Win32/ZeroAccess is an executable file. The machine gets infected with this dropper through drive-by download or through social engineering, which lures the end user to download an executable. Upon execution, the dropper chooses a system driver as a target for infection, backups the original driver and runs the infected one. The dropper is being deleted right after this action.
2. The Driver – it randomly selects a driver to infect by enumerating all kernel modules, using the undocumented call to **ZwQuerySystemInformation** (figure 9):

```
Status = NtQuerySystemInformation(SystemModuleInformation,
                                   pSysModules,
                                   Length,
                                   &Length);
```

Figure 9 - undocumented call to ZwQuerySystemInformation

Because the rootkit uses a Windows driver to infect, it must also keep the original driver image to keep the boot process unharmed. The original driver is stored within the rootkit's encrypted storage (a folder within the Windows directory, figure 10). Furthermore, the rootkit hides the content of the infected driver. It uses the same method that has been used by Win32/Alureon at times when it infected a random system driver. The trick is based on creating artificial incoherence between system cache and the on-disk file. The file stored is then encrypted using RC4 algorithm.

<sup>7</sup> <http://en.wikipedia.org/wiki/Rootkit>

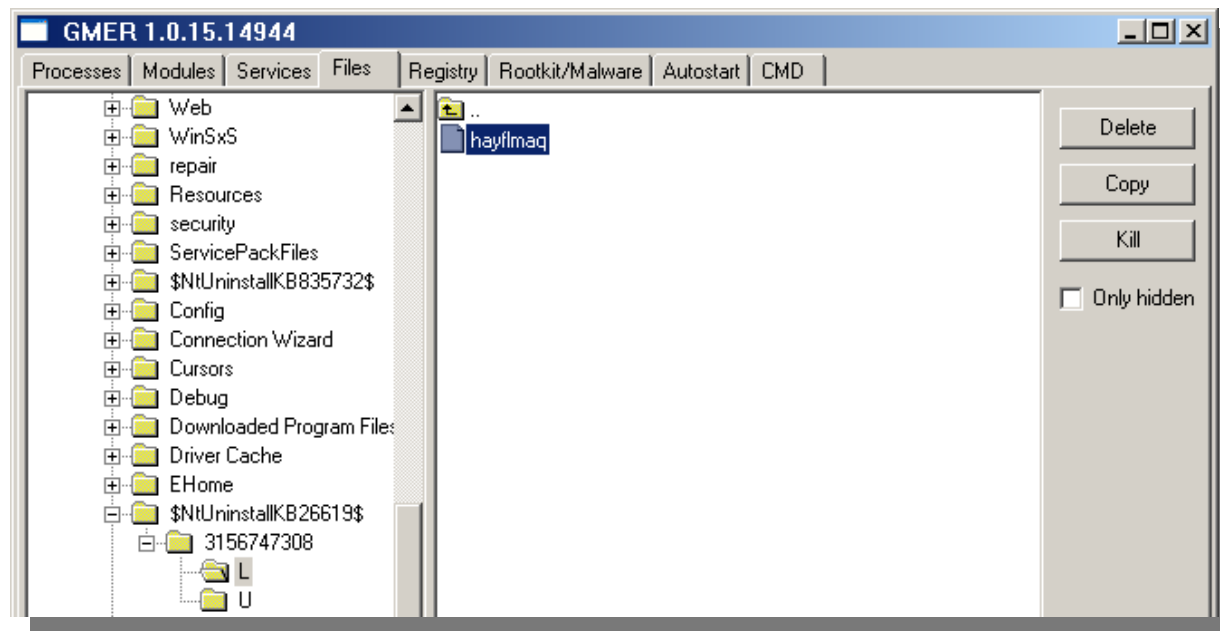


Figure 10 - The Original Driver is Stored Within the Rootkit's Protected Storage in a File

3. The most interesting part of this rootkit is the method used to protect itself from anti-malware software, the "bait" (tripwire) – this rootkit ensures staying on the system and running properly:
  - a. The first method relies on the fact that security software often scans Windows alternate data streams (aka ADS<sup>8</sup>) of files. The rootkit creates a "bait" service with a random name (installed as \systemroot\740567228:1138219987.exe, figure 11). The service is of the structure type SERVICE\_OWN\_PROCESS and its image file is stored in the alternate data stream (ADS).

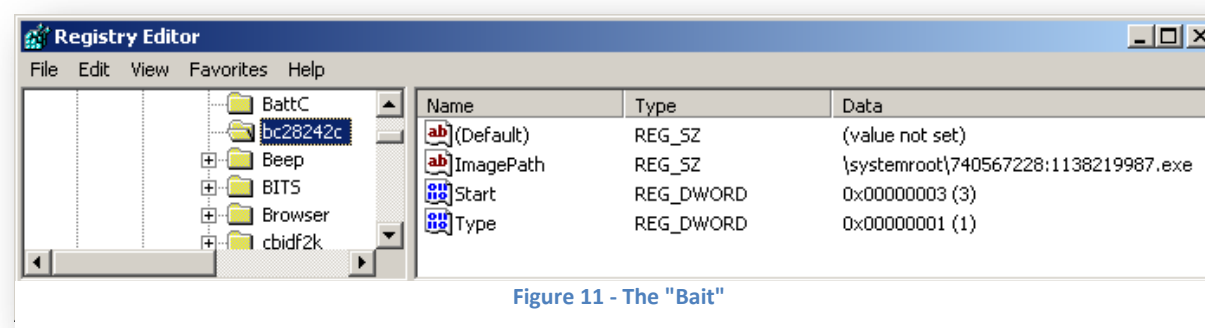


Figure 11 - The "Bait"

The protection of the process against AV scanners is performed by hooking the `nt!IoIsOperationSynchronous` function ( figure 12):

```

804ee860 894218      mov     dword ptr [edx+18h],eax
804ee863 5b          pop     ebx
804ee864 c9          leave
804ee865 c20800      ret     8
804ee868 cc          int     3
804ee869 e8f70f3902 call     8287f865
nt!IoIsOperationSynchronous:
804ee86e ebf9        jmp     nt!IoReuseIrp+0x8b (804ee869)
804ee870 55          push    ebp
804ee871 8bec        mov     ebp,esp
804ee873 8b4508      mov     eax,dword ptr [ebp+8]

```

Figure 12 - `nt!IoIsOperationSynchronous` Hooking

This kernel-mode function is often called during file open operations. **When an application or a driver is caught opening the alternate data stream on that service, the process is killed and the image file's security descriptor is modified so that it is not allowed to be executed again.**

- b. The second anti-security-software method (ab)uses the fact that most of the security software scans system services from the system registry, such as HKLM\System\CurrentControlSet\Services. The rootkit installs a registry callback and observes attempts to query the value of "ImagePath" registry value. This registry value, presented in service database, contains the name of the binary that is used to create the service. **Once an application/process reads this value more than 64-times in a row, it is killed and disabled as mentioned above.**

<sup>8</sup> ADS: A stream is a sequence of bytes. In the NTFS file system, streams contain the data that is written to a file, and that gives more information about a file than attributes and properties. <http://msdn.microsoft.com/en-us/library/aa364404.aspx>



#### Malicious Activity

- ZeroAccess logs and reports user's activity to a remote server.
- ZeroAccess can hide its connections to a remote server
- ZeroAccess can terminate security software, which is running on the machine.
- ZeroAccess is waiting for instructions from the Command and Control server.

#### Recommendations

- Check before you click: be aware not to fall into the traps set by cyber criminals using social engineering, do not install any unknown / unfamiliar executable file.
- Avoid accessing suspicious websites
- AVG products detect this malware and its variants, the whole family
- AVG provides free utility to remove ZeroAccess rootkit, available for download on our website.



## Mobile Devices Risks & Threats

### Trusted Malware?

The title seems familiar to you? Of course it does; we have covered this subject in our [Q2 Threat Report](#) from the web point of view. In Q2, we anticipated that 'stolen keys' such as digital certificates, tokens and passwords will eventually become a significant problem. This trend also covers mobile platforms.

It appears that mobile malware is getting to be more and more like PC malware. Only two quarters later, cybercriminals shifted quickly from PC to mobile devices and use the same tricks attacking Android based devices.

### Background<sup>9</sup>

In order to deploy any application on Android devices (or on the Android Market), the application needs to be digitally signed. The private key is held by the application's developer. The certificate does not need to be signed by a central authority; it can be signed by a third-party (OEM, operator, alternative market) or self-signed.

There are several reasons why applications need to be signed:

- Trust - The Android system uses the certificate as a means of identifying the developer of an application and establishing trust relationships between users and the developer.
- Application upgrade - Code signing allows developers to update their application seamlessly to the new version, without creating complicated interfaces and permissions.
- Application modularity – The Android system allows applications that are signed by the same certificate to run in the same process (the system treats them as a single application).
- Code/data sharing through permissions – The Android system provides signature-based permissions enforcement, so that an application can expose functionality to another application that is signed with a specified certificate.

### Digital Signature Pitfalls

As we learned from the Windows world, a **digitally signed application does not necessarily equal trustworthiness**, especially when it is self signed.

Maintaining the security of the private key is of critical importance, both to the developer and to the user. If the private key falls into the wrong hands, the authoring identity and the trust of the user are compromised.

Stealing or faking a private key of a trusted source (developer), will allow cyber criminals to sign their malicious applications with the same key as the trusted developer. By doing so, the cyber criminal could sign and distribute applications that maliciously replace the authentic applications or corrupt them. Such a person could also sign and distribute applications under the developer's identity that attack other applications or the system itself, corrupt or steal user data. **The reputation as a developer entity depends on securing the private key properly.**

By impersonating as the trusted developer the cyber criminals can utilize the already established trust between the users and the developer and can achieve faster distribution and high success rates. Cyber criminals could also fake a trusted signature and achieve the same advantages as described above.

AVG Threat Labs have witnessed several cases, which illustrate the above pitfalls:

- [Case #1 :Using fake certification used by malware](#)
- [Case #2: Using leaked personal developer certificate](#)
- [Case #3: Using Android Open sources Project \(AOSP\) certificate by malware](#)

---

<sup>9</sup> <http://developer.android.com/guide/publishing/app-signing.html>



### Case #1: Using a Fake Google Certificate – if you can't have it, fake it

In the first example, we show how malware writers are using a fake Google certificate. Just for comparison, in figure 13, you can see a legitimate Google certificate which is being used in a legitimate calendar application:

```
[
  Version: V3
  Subject: CN=Android, OU=Android, O=Google Inc., L=Mountain View, ST=California, C=US
  Signature Algorithm: MD5withRSA, OID = 1.2.840.113549.1.1.4

  Key: Sun RSA public key, 2048 bits
  modulus: 276762005270839880203434052095350186283566770007377065428048161648738
  53564809440589025034441404374281076306080441620574894434640671669027025279635619
  32930040224025890501029366745473098499566909006979517670015637706247259588147331
  33128036824328968158866344052798404339547924678808456990689679680745278615376077
  172084506135550658354134557258306268256535541506355230833116003605301529491580
  6807884329226640232776970208196303057741987219525651968059883145048289583605853
  13324008698354806985183216376838436292021045233669009787366254734962373162725725
  57833391695092662950221754216391581758455171460879064884662263628037
  public exponent: 3
  Validity: [From: Thu Nov 26 02:07:02 IST 2009,
    To: Mon Apr 13 03:07:02 IDT 2037]
  Issuer: CN=Android, OU=Android, O=Google Inc., L=Mountain View, ST=California, C=US
  SerialNumber: [ 809ef517 f081c24b]

  Certificate Extensions: 3
  [1]: ObjectID: 2.5.29.14 Criticality=false
  SubjectKeyIdentifier [
    KeyIdentifier [
      0000: 58 80 F9 5E 7F 59 F5 9D EE B9 D8 E4 F9 71 50 9A X..^Y.....qP.
      0010: 3A 74 0D 1D .....t..
    ]
  ]
]
```

Figure 13 - Certificate of Legitimate Calendar Application

AVG Threat Labs have seen a malware by the name of 'DroidKungFu' (package name 'com.adwo.android.games.mine') that is using a fake Google Certificate (Figure 14); this piece of malware circulates among alternative Chinese app markets:

```
[
  Version: V3
  Subject: OU=Google Inc., C=US
  Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5

  Key: Sun RSA public key, 1024 bits
  modulus: 138323794793924324684680368238501344507354686663960553219177457634976999802149757806076963505904493498846561
  25535388237216266541561932825683674764725572714849958595597723882412042199507681331348506026034692870599627472525919251
  71882708116498726230967786535709439533149511334105873775271139465270933070018161
  public exponent: 65537
  Validity: [From: Wed May 04 06:31:13 IDT 2011,
    To: Thu Apr 10 06:31:13 IDT 2110]
  Issuer: OU=Google Inc., C=US
  SerialNumber: [ 4dc0c881]
]
```

Figure 14 - Fake Google certificate used by a malware

Below (Figure 15), you can see the difference between the legitimate Google certificate (on the left) and the fake Google certificate (on the right)

Original legitimate Google certificate	DroidKungFu – Signed with a 'fake' certificate
<pre>-----certificate----- Version: V3 Subject: CN=Android, OU=Android, O=Google Inc., L=Mountain View, ST=California, C=US Signature Algorithm: MD5withRSA, OID = 1.2.840.113549.1.1.4  Key: Sun RSA public key, 2048 bits modulus: 276762005270839880203434052095350186283566770007377065428048161648738 53564809440589025034441404374281076306080441620574894434640671669027025279635619 32930040224025890501029366745473098499566909006979517670015637706247259588147331 33128036824328968158866344052798404339547924678808456990689679680745278615376077 172084506135550658354134557258306268256535541506355230833116003605301529491580 6807884329226640232776970208196303057741987219525651968059883145048289583605853 13324008698354806985183216376838436292021045233669009787366254734962373162725725 57833391695092662950221754216391581758455171460879064884662263628037 public exponent: 3 Validity: [From: Thu Nov 26 02:07:02 IST 2009,   To: Mon Apr 13 03:07:02 IDT 2037] Issuer: CN=Android, OU=Android, O=Google Inc., L=Mountain View, ST=California, C=US SerialNumber: [ 809ef517 f081c24b]  Certificate Extensions: 3 [1]: ObjectID: 2.5.29.14 Criticality=false SubjectKeyIdentifier [   KeyIdentifier [     0000: 58 80 F9 5E 7F 59 F5 9D EE B9 D8 E4 F9 71 50 9A X..^Y.....qP.     0010: 3A 74 0D 1D .....t..   ] ] -----</pre>	<pre>-----certificate----- Version: V3 Subject: OU=Google Inc., C=US Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5  Key: Sun RSA public key, 1024 bits modulus: 13832379479392432468468036823850134450735468666396055321917745763497 69998021497578060769635059044934988465612553530823721626654156193282568367476472 5572714849958595597723882412042199507681331348506026034692870599627472525919251 71882708116498726230967786535709439533149511334105873775271139465270933070018161 public exponent: 65537 Validity: [From: Wed May 04 06:31:13 IDT 2011,   To: Thu Apr 10 06:31:13 IDT 2110] Issuer: OU=Google Inc., C=US SerialNumber: [ 4dc0c881] -----</pre>

Figure 15 - Legitimate vs. Fake Certificates

This specific malware is requesting the following permissions (figures 16, 17):

This application is available for download from one of Android alternative markets, and disguises as a minesweeper game. You can well see that the developer registered is 'Android Developers' (Mark with red underlined in figures 18,19)

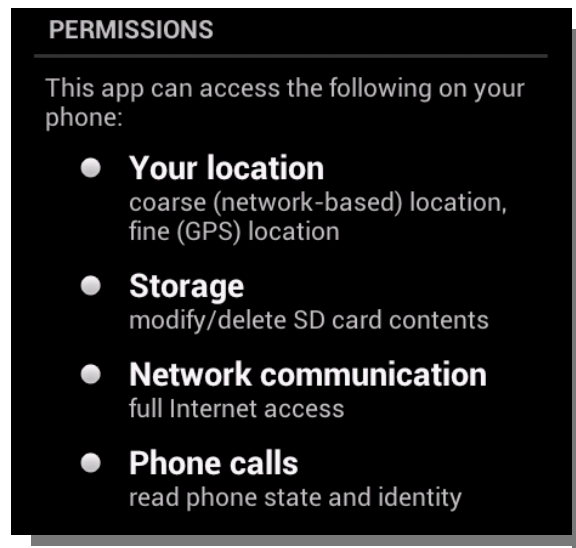


Figure 17 - Permissions as Appear on the GUI

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Figure 18 - Permissions in AndroidManifest.xml



Figure 16 - Original Download Site



Figure 19 - Translated Page

### Malicious Activity

This piece of malware is capable of rooting the vulnerable android phone. When the malware runs, it tries to gain root access on the device and also to contact a remote server to send the collected information.



## Case #2: Lorenz Leaked Certificate Case – Don't leave the (private) key out there, unprotected

In this case, the cyber criminal was impersonating a reputable developer by the name of Lorenz. This developer posted on his blog ([www.londatiga.net](http://www.londatiga.net)), see figure 20), a detailed procedure of how to sign android application (APK). Unfortunately, the developer included in his post also his private key, something that should be kept private as the name suggests.

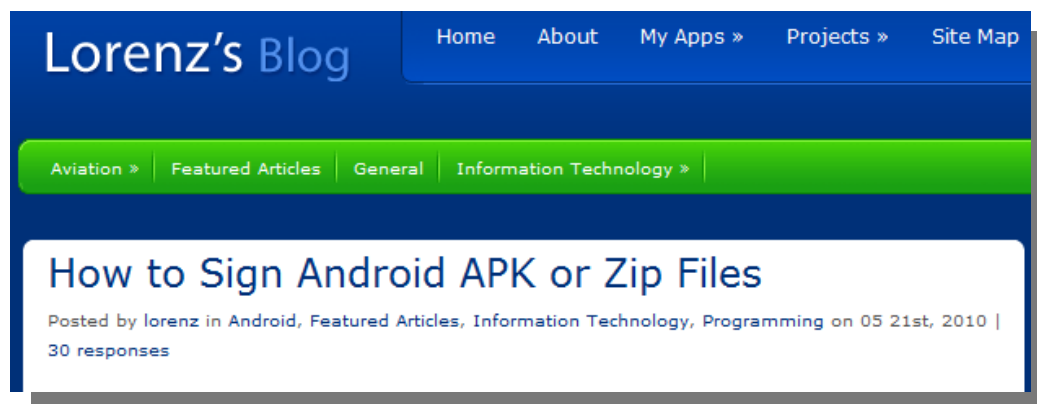


Figure 20 – Lorenz's Blog

Sure enough, cyber criminals were quick in exploiting this opportunity and used his private key to sign their own malicious application (package name ('com.android.vending.sectool.v1'). As said above, using a 'reputable' private key when uploading a malicious application to the market, almost guarantees malware distribution.

The malware impersonates as a security tool released by Google, in figure 21, you can see a long list of permissions (as part of AndroidManifest.xml)

```
<manifest android:versionCode="6" android:versionName="1.5" package="com.android.vending.sectool.v1"
  xmlns:android="http://schemas.android.com/apk/res/android">
  <uses-sdk android:minSdkVersion="4" />
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
  <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
  <uses-permission android:name="android.permission.RECEIVE_SMS" />
  <uses-permission android:name="android.permission.SEND_SMS" />
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
  <uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
  <uses-permission android:name="android.permission.READ_PHONE_STATE" />
  <uses-permission android:name="android.permission.WAKE_LOCK" />
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
  <uses-permission android:name="android.permission.INTERNET" />
```

Figure 21 - 'com.android.vending.sectool.v1' permissions

As seen in Figure 22, the malicious application is using Lorenz's key:

```
[
  Subject: EMAILADDRESS=lorenz@londatiga.net, CN=Lorenz W. L. T, OU=AndroidDev, O=Londatiga, L=Bandung, ST=Java Barat
  C=ID
  Signature Algorithm: sha1WithRSA, OID = 1.2.840.113549.1.1.3
  Key: Sun RSA public key, 1024 bits
  modulus: 1428424557167504313206058412475311856067911625664197707154334934157686641599930908317147388758978219082243102
  372754314185203434139932212181770748802912995954328091316882019210403399849407336780847023281134366689891451441757579709
  03579074634000877134826251997835467445397684730038003052921733614356901636075983
  public exponent: 65537
  Validity: (From: Mon May 05 12:21:38 IDT 2010,
  To: Mon May 05 12:21:38 IDT 2010)
  Issuer: EMAILADDRESS=lorenz@londatiga.net, CN=Lorenz W. L. T, OU=AndroidDev, O=Londatiga, L=Bandung, ST=Java Barat
  C=ID
  Version: 1
  Serial Number: 1
  Signature: 001032A 1E00C077
]
```

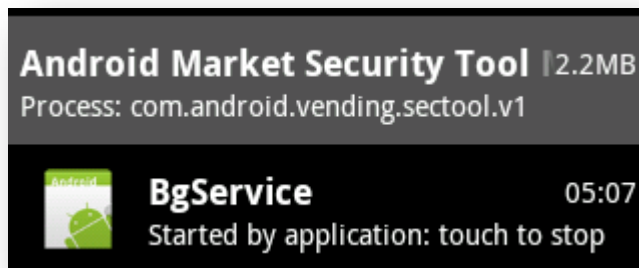
Figure 22 - Application Certificate

Figure 23 shows the icon of the malware looks like:



Figure 23 - Application's Icon

The application is running in the background. No screen is opened when the application is running:



### Case #3: Using AOSP certificate by malware – a home run

Android is an open-source software stack created for mobile phones and other devices. The Android Open Source Project (AOSP), led by Google, is tasked with the maintenance and further development of Android<sup>10</sup>.

Many device manufacturers have released devices running Android, and they are available around the world.

Developers use a publicly available private key that the Android Open Source Project (AOSP) contains.

The problem is that an application signed with the AOSP certificate is trusted and be granted permissions without user intervention when installed on most custom ROMs as it is the same certificate as the system image.

It is good to mention that the Android emulators are using the AOSP certificate (not a specific manufacturer of an Android device).

The following is a certificate from a piece of malware with the package name 'com.hotel' (Figure 24)

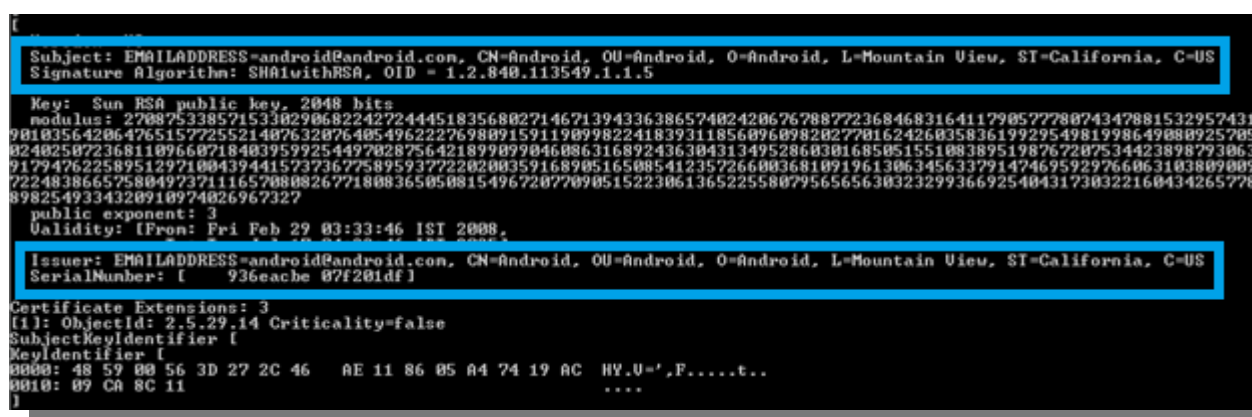


Figure 24 - 'com.hotel' package certificate

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.INTERNET" />
```

Figure 25 - 'com.hotel' permissions

The permissions this malware request are (figure 25):

As it can be seen the details found in the certificate relate to Android project.

<sup>10</sup> [http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))



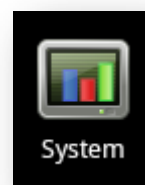


The required permissions (figure 29):

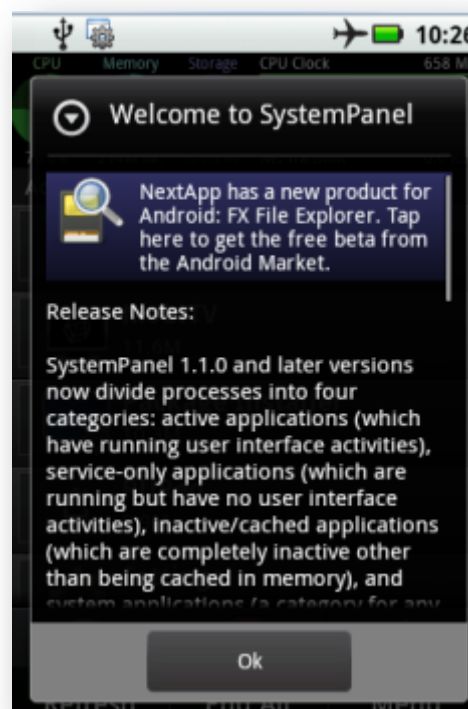
```
<uses-sdk android:minSdkVersion="3" android:targetSdkVersion="4" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.RESTART_PACKAGES" />
<uses-permission android:name="android.permission.GET_TASKS" />
<uses-permission android:name="android.permission.INSTALL_PACKAGES" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Figure 27 - Application's Permissions

The icon of the malware looks like the following:



When the application is running, the following screen is displayed:



## Recommendations

- Private Keys are called private with a reason. Keep them private and protected!
- Pay attention to the list of permission and if it makes sense to have it for this specific application.

## Printed Malware

### Overview

Typing a long text, especially with different symbols in it, using keyboards of mobile devices is challenging. Switching back and forth between the keyboard layouts to access the symbols or to accurately type the text without typos is not as simple as in desktop PC or Mac computers. This has become a major issue for advertisers that are keen to get mobile users access their content.

Finding methods that can simplify this process and make it error free resulted in encoding the text into graphical symbols. Taking advantage of the built-in camera in almost any mobile device and utilizing image processing algorithms, the typing process becomes as non-issue anymore. One frequently used technique is known as QR codes (abbreviated from **Quick Response code**).

Today QR symbols are showing on almost any ad you find on the street, at a conference or even online. Mobile users can simply scan the QR symbol using software on their mobile device and have their device transform it into meaningful information.

Trying to shorten the text, more specifically long URLs, isn't new. Many [Twitter](#) users are familiar with URL shortening services like [bit.ly](#) and many others. A URL such as <http://www.avg.com/cz-en/free-antivirus-download> can be shortened into <http://bit.ly/vJGG2M>.

### QR codes as URL shortening are not 100% risk-free

Both URL shortening and QR codes are hiding behind their patterns a URL or text the users cannot identify until he/she is actively executing or scanning it. **However executing an unknown pattern of symbols on your trusted mobile or computer is something you should be careful with.** This is almost the same as running an unknown executable on your computer.

Exploiting URL shortening to hide malicious URLs is a reality since the very beginning of such services being introduced to users. Now is the time for QR codes to serve malware or direct you to malicious URLs.

Printing QR codes hiding malicious URLs and sticking them on ads, papers or just uploading them online is a very simple and easy way to get people's mobile device infected.

Compromising websites and replacing their legitimate QR codes with malicious ones, will certainly not get the website owner attention fast enough before the sites' mobile visitors get infected.

We expect this infection method to increase in 2012 and beyond.

### QR codes serving malware - Example

Late September 2011, a new malware that spreads through QR was found on a Russian website and forums. The posts about the malicious application redirects the user to a site that contains Trojan embedded in the 'Jimm' mobile ICQ client.

Once the malicious QR code is scanned, the victim is redirected to a URL where the malicious application file can be found and downloaded. The Trojanized application that the user thinks it downloads to his smartphone is an instant messaging client for ICQ network named 'Jimm'.

The package name of the malware is "appinventor.ai\_russ\_support.JimmRussia":

```
package="appinventor.ai_russ_support.JimmRussia"
```

#### Permissions:

These are the permissions requested by the malware:

```
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
```

#### Appearance:

That is what seen after installing the application:



You can spot the 'Russian connection' in the text under the application icon.

After opening the application we can see the following screen:

#### Настройка и установка Jimm

Ждите. Время ожидания от 2 до 5 минут. Не закрывайте это приложение!

Служба поддержки: SMShelp.su

#### Malicious Behavior:

The malware sends SMS messages to premium numbers. The malware also redirects to a URL to download a malicious file.

#### Try it yourself

Scan the following QR code to find what's behind these dots





## Other reports from AVG Technologies

**AVG and Ponemon Institute: 'Smartphone Security - Survey of U.S. consumers'** – March 2011

<http://aa-download.avg.com/filedir/other/Smartphone.pdf>

**Anatomy of a major Blackhole attack** – March 2011

<http://www.avg.com/filedir/other/blackhole.pdf>

**AVG Community Powered Threat Report Q1 2011** – April 2011

<http://www.avg.com/ww-en/press-releases-news.ndi-129>

**AVG Community Powered Threat Report Q2 2011** – June 2011

<http://www.avg.com/ww-en/press-releases-news.ndi-1563>

**AVG and Future Laboratories: 'Cybercrime Futures'** – September 2011

<http://www.avg.com/ww-en/press-releases-news.ndi-1953>

**AVG and GfK: 'AVG SMB Market Landscape Report 2011'** – September 2011

[http://download.avg.com/filedir/news/AVG\\_SMB\\_Market\\_Landscape\\_Report\\_2011.pdf](http://download.avg.com/filedir/news/AVG_SMB_Market_Landscape_Report_2011.pdf)

**AVG Community Powered Threat Report Q3 2011** – October 2011

<http://www.avg.com/ww-en/press-releases-news.ndi-2323>

## About AVG Technologies

AVG Technologies is a global leader in security software, protecting more than 100 million consumers and small business computer users in 170 countries. Headquartered in Amsterdam, AVG is the fourth largest vendor of anti-virus software and employs close to 600 people worldwide with corporate offices in the US, the UK, the Netherlands, the Czech Republic, and Germany.

AVG has nearly two decades of experience in combating cyber crime and operates one of the world's most advanced laboratories for detecting, pre-empting and combating web-borne threats from around the globe for both businesses and home customers.

The company boasts one of the most extensive self-help communities on the Internet, having established its technology credentials early on amongst technically savvy consumers.

[www.avg.com](http://www.avg.com)