At the F-Secure Response Labs in Helsinki, Finland, and Kuala Lumpur, Malaysia, security experts work around the clock to ensure our customers are protected from the latest online threats.

At any given moment, F-Secure Response Labs staff are on top of the worldwide security situation, ensuring that sudden virus and malware outbreaks are dealt with promptly and effectively.

Protection around the clock

Response Labs’ work is assisted by a host of automatic systems that track worldwide threat occurrences in real time, collecting and analyzing hundreds of thousands of data samples per day. Criminals who make use of virus and malware to profit from these attacks are constantly at work on new threats. This situation demands around the clock vigilance on our part to ensure that our customers are protected.
In 2008, a mathematician called Satoshi Nakamoto (a pseudonym) submitted a technical paper for a cryptography conference. The paper described a peer-to-peer network, where participating systems would do complicated mathematical calculations on something called a “blockchain.” This system was designed to create a completely new currency — a crypto currency — which is based on math. The paper was titled “Bitcoin: A Peer-to-Peer Electronic Cash System.”

Since Bitcoin is not linked to any existing currency, its value is purely based on the value people believe it’s worth. And since it can be used to do instant transactions globally, it does have value. Sending Bitcoins around is very much like sending e-mail. If I have your address, I can send you money. I can send it to you instantly, anywhere, bypassing exchanges, banks and the tax man. In fact, crypto currencies make banks unnecessary for moving money around. Which is why banks hate the whole idea.

The beauty of the algorithm behind Bitcoin is solving the two main problems of crypto currencies by joining them — how do you confirm transactions, and how do you inject new units of currency into the system without causing inflation. Since there is no central bank in the system, the transactions need to be confirmed somehow. Otherwise one could fabricate fake money. In Bitcoin, the confirmations are done by other members of the peer-to-peer network. At least six members of the peer-to-peer network have to confirm the transactions before they go through. But why would anybody confirm transactions for others? Because they get rewarded for it: the algorithm issues new Bitcoins as a reward to users who have been participating in confirmations. This is called “mining.”

When Bitcoin was young, mining was easy and you could easily make dozens of Bitcoins on a home computer. However, as Bitcoin value grew, mining became harder since there were more people interested in doing it. Even though the dollar-to-BTC exchange rate has fluctuated, the fact remains that in the beginning of 2013, one Bitcoin was worth USD 8 and by the fall they were worth USD 130. So Bitcoins now have real-world value.

Today, there are massively large networks of computers mining Bitcoins and other competing crypto currencies (such as Litecoin). The basic idea behind mining is easy enough: if you have powerful computers, you can make money. Unfortunately, those computers don’t have to be your own computers. So, you’d have an infected home computer of a grandmother in, say, Philadelphia, running Windows XP at 100% utilization around the clock as it is mining coins for a Russian cyber crime gang.

As detailed in this Threat Report, the second largest botnet in the world is already doing this. We estimate them to make over $50,000 a day by mining Bitcoins on infected computers. If such operations are already happening today, it’s easy to see that such mining botnets will become very popular for online criminals in the future.

Mikko Hyppönen
CRO, F-Secure Labs
http://twitter.com/mikko

“THE BASIC IDEA BEHIND MINING CRYPTO CURRENCIES IS EASY ENOUGH: IF YOU HAVE POWERFUL COMPUTERS, YOU CAN MAKE MONEY.”
EXECUTIVE SUMMARY

Applying security patches can be compared to putting locks on your doors and windows upon hearing the news of burglars lurking around town. But the problem is there are so many burglars out there and you have so many doors and windows to secure - you don’t know which exact way they will enter your house. You can patch the holes one after another, but the burglars will keep on trying. It sounds like a plot of a thriller movie but it is not a farfetched story considering our dependence on computer devices nowadays. The bad guys are always on the lookout for any vulnerabilities - the holes in the wall that can be exploited.

Exploit attacks against known vulnerabilities are everywhere and Java-targeted ones are leading the pack. In the previous half-year, Java-targeted exploits accounted for about one-third of the detections reported to our cloud-based telemetry systems from our protected clients; in the first half of this year, they made up almost half of the detections reported. The Java exploits mostly targeted the CVE-2013-1493 and CVE-2011-3544 vulnerabilities, while an exploit for the Windows TrueType Font vulnerability CVE-2011-3402 makes up 10% of the Top 10 Detections reported globally (a 4% increase for the same exploit from H2 2012). We also look into the exploit kits being used to facilitate most of these attacks, particularly Blackhole, SweetOrange and Cool.

The whole point of carrying out an exploit-based attack is to install malware onto a system - be it ransomware, bots, banking trojans or backdoors. We noted that “Anti Child Porn Spam Protection” ransomware was circulating during March and April this year. The ZeroAccess botnet was quite active as well during the same period, even though our statistics seem to show the number of ZeroAccess infections drastically declining. This is probably due to the effectiveness of our Majava detections, which successfully prevented attacks that would otherwise have delivered ZeroAccess.

With the recent peak in Bitcoin’s exchange rate, we tried making an educated guesstimation of ZeroAccess’s potential monthly profit from Bitcoin mining activity. And the number is huge.

One exploitation method used Advanced Persistent Threat (APT) attacks involves specially crafted ‘bait’ document sent to targeted persons in an organization or field. An analysis of such APT documents reveals, among other things, that the most common subjects are political, corporate, and military in nature.

On the mobile threats side, Android is still the most popular target, as almost all mobile malware that we encountered during this half-year period operate on this platform. A technically interesting mobile threat we cover in this report is Stels, the first Android malware to be distributed via spam e-mails, and a bot that uses Twitter to update its command and control (C&C) server addresses.

On the Mac side of things, during H1 2013 we encountered an interesting malware called Kumar in the Mac (KitM). It is the first Mac malware signed with a valid developer ID. Talk about arrogance.
**CONTENTS**

This threat report highlights trends and new developments seen in the malware threat landscape by analysts in F-Secure Labs during the first half of 2013. Also included are case studies covering selected noteworthy, highly-prevalent threats from this period.

<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FOREWORD</strong></td>
<td>3</td>
</tr>
<tr>
<td><strong>EXECUTIVE SUMMARY</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>CONTENTS</strong></td>
<td>5</td>
</tr>
<tr>
<td><strong>INCIDENTS CALENDAR</strong></td>
<td>6</td>
</tr>
<tr>
<td><strong>IN REVIEW</strong></td>
<td>7</td>
</tr>
<tr>
<td><strong>OF NOTE</strong></td>
<td>10</td>
</tr>
<tr>
<td><strong>WATERING HOLES</strong></td>
<td>11</td>
</tr>
<tr>
<td><strong>CASE STUDIES</strong></td>
<td>13</td>
</tr>
<tr>
<td><strong>STELS</strong></td>
<td>14</td>
</tr>
<tr>
<td><strong>ANDROID</strong></td>
<td>18</td>
</tr>
<tr>
<td><strong>APT ATTACKS</strong></td>
<td>23</td>
</tr>
<tr>
<td><strong>RANSOMWARE</strong></td>
<td>26</td>
</tr>
<tr>
<td><strong>CRYPTO CURRENCY MINING</strong></td>
<td>30</td>
</tr>
<tr>
<td><strong>EXPLOIT KITS</strong></td>
<td>34</td>
</tr>
<tr>
<td><strong>VULNERABILITY EXPLOITATION</strong></td>
<td>36</td>
</tr>
<tr>
<td><strong>KUMAR IN THE MAC (KITM)</strong></td>
<td>40</td>
</tr>
<tr>
<td><strong>PHISHING</strong></td>
<td>44</td>
</tr>
<tr>
<td><strong>SOURCES</strong></td>
<td>47</td>
</tr>
</tbody>
</table>
INCIDENTS CALENDAR

JAN
- Cool/Blackhole exploit kit author purchasing exploits

FEB
- Miniduke malware exploits CVE-2013-0640
- Wiper malware attacks reported in South Korea

MAR
- Red October infrastructure shut down
- Evernote hacked
- Android SMS trojans target South Koreans
- Interpol shuts down ransomware network
- US FCC approves Anti-bot Code for ISPs

APR
- CVE-2013-0634 exploit in the wild
- Spamhaus hit with massive DDoS attack
- Twitter, Apple, Facebook & others hacked
- US jails LulzSec hacker for Sony breach

MAY
- CVE-2013-2423 exploit in the wild
- SafeNet espionage reported
- Syrian Electronic Army hacks The Onion
- Twitter offers multi-factor login

JUN
- Signed Mac malware discovered at conference
- NSA PRISM program leaked
- Op’ NetTraveler espionage reported
- Naikon attacks in Asia reported
- US jails LulzSec hackers for “50 days of lulz”
- Spamhaus DDoS suspect extradited to Netherlands
- Microsoft offers standing bug bounty
- FBI & Microsoft take down Citadel botnet

Sources: see page 47-48
IN REVIEW

UPDATES IN THE THREAT LANDSCAPE

The first half of 2013 mostly saw the continued development of familiar threats we noted in our previous reports. Of special interest this half year: the increasing use of exploit-based attacks facilitated by exploit kits, particularly those targeted against the Java development platform; Bitcoin mining and its ramifications; increasing sophistication in mobile threats; the discovery of the first signed Mac malware; and profiling of phishing sites. Beyond the technical developments, cyber espionage, hacks and privacy concerns became big news during this period, as major news sites and tech companies reported significant infiltration attempts and revelations about the covert data-gathering programs by the United States’ National Security Agency (NSA) raised questions about online privacy.

Based on the statistics from our cloud-based telemetry systems, which monitors detections globally reported from clients protected by our products, the majority of the Top 10 Detections seen in the past six months (see page 9) involved exploits. Geographically, the United States and France saw the most exploit-based attacks. Nearly 60% of the Top 10 Detections involved attacks that used exploits, and 80% of those were targeted against the Java development platform. The statistics illustrate the increasing focus of attackers on exploiting Java to gain access to the user’s machines. Since it’s a hugely popular application, Java is almost certainly found in organizations involved in software development, and an even better real-world illustration of the platform’s ubiquitousness and its role as an entrypoint for attackers is the large-scale attack that took place during February this year, when various tech companies, including Twitter and Facebook, were reportedly hacked via a Java exploit [1].

Almost half of the exploit-based attacks reported by our clients were blocked by our heuristic Majava detections, which prevented the attacks from infiltrating the system and installing malware. Interestingly enough, Majava’s effectiveness had a curious impact on the statistics for ZeroAccess detections reported by our clients, which when compared against the statistics for the same malware in the second half of 2012, saw a noticeable decline. Based on the very active developments we’ve seen for the ZeroAccess family during the last six months however, it would not be a stretch to assume that had those exploit attacks successfully occurred, most of the malware they dropped would have been ZeroAccess-related. Where we did see ZeroAccess-related detections being reported, they mostly occurred in the United States, France, Sweden and Italy.

Most exploit-based attacks we’ve seen are facilitated by a small handful of exploit kits, with 70% of them being attributed to these five: BlackHole, SweetOrange, Crimeboss, Styx and Cool. The exploit kit development space has been highly active in the last few months, as we’ve observed at least one new (or revamped) exploit kit being created each month on average. The top five kits are also undergoing active development, with their authors constantly adding new exploits targeting recently announced vulnerabilities.

Older vulnerabilities, for which security patches have already been issued, also continued to trouble users. Almost five years after the Downdup/Conficker outbreak, we still find it lingering on in Brazil. And even though our specific detection for exploits targeting the CVE-2011-3402 vulnerability was created in 2011, two years on the number of clients reporting it haven’t reduced at all. In fact, it grew from 6% of all reported detections in the last half 2012 to 10% in this period (mostly in France, Germany and Sweden). The continuing detection of exploit threats that were discovered and addressed years ago again brings into question the effectiveness of current patching procedures, which too often leave users at the mercy of attacks against old, known loopholes.

Another field that continues to thrive despite using known tricks is phishing. With construction kits now available that simplify the production of phishing sites, it has become even easier

\[\text{More details on page 34}\]
for phishers to operate, typically by using spam emails to lure victims onto these automatically
created sites, where their personal details or money can be stolen. In a profiling of known phishing
sites, we saw that the most common type of portals these sites mimic involve payment services,
banks and gaming, with fake PayPal sites making up almost 73% of the phishing sites we
analyzed.

**Ransomware** continues to make the rounds and in March and April this year we noted cases of
ransomware particularly targeting corporate clients in countries such as Spain and Italy. Almost all
the ransomware cases we saw during this half-year fell into two broadly different types: encrypting
and police-themed, of which the latter is more common.

As Mikko mentioned in his foreword, **Bitcoin** and Litecoin mining is one of the big money-making
options for cyber-criminals. ZeroAccess’s on-and-off again relationship to Bitcoin has been
interesting. Though the malware originally contained a module for Bitcoin mining, it was removed
from samples we found last September. In April this year, the samples we saw had their Bitcoin
payload reinstated, coincidentally just when Bitcoin’s monetary value went over USD 200 [2]. Since
late April however, the malware’s mining facility has again been pulled. Why that is so is something
of a mystery, though some speculate it is due to the high maintenance cost of private pool servers
and the heavy computing power needed for Bitcoin mining, which makes the malware harder to
hide on infected machines. We tried making an educated guesstimate* of ZeroAccess’ potential
monthly profit from Bitcoin mining and came up with a figure of USD 1.8 million.

On the Mac front, the February attacks were interesting because they also targeted Mac users,
particularly employees of the Apple company, via a compromised mobile developer site [1].
Though attacks against Mac users aren’t unheard of, until now they had mostly been targeted
against political activists—case in point, the **signed Mac spyware** found on an activist’s Mac at
the Oslo Freedom forum in May [3]. Perhaps not coincidentally then, of the 33 new Mac malware
families or variants we saw in H1 2013, 57.6% were backdoors.

Given the scope and scale of the February attacks, and the speculations about its intended
objective, they could arguably be considered **Advanced Persistent Threat (APT) attacks**. More
commonly though, APT attacks involve bait documents, which are carefully crafted to appeal to
users in the targeted organization. Given the nature of these files, most targets are understandably
reticent to share them with outsiders. An analysis of bait documents we were able to obtain
allowed us make generalizations about their contents: for example, that the most common type
of content related to political (65%), corporate (14%) and military (11%) subjects, while the most
common languages were English, Chinese and Arabic. The analysis allows us to build a sketchy
profile of the type of individual the documents were meant to lure.

On the **mobile threats** scene, Google’s Android continues to be the most targeted mobile
operating system, accounting for 96% of all new mobile malware families or variants we saw in H1
2013. Google’s Play Store also officially superseded Apple’s App Store this half-year to become
the largest app market, surpassing the one million apps mark in July. Despite lingering questions about
the Play Store’s security, it remains by far the safest Android app market around, as the majority of
new Android malware we saw were found on non-Play Store sites. In terms of functionality, most
of the mobile threats we’ve seen were either banking-trojans or were involved in malvertising.

**Banking-trojans**, which typically steal Mobile Transaction Authentication Numbers (mTans),
appear to be increasing as more banks shift to using this form of authentication to verify online
transactions. In the last few months we’ve also noticed increased instances of **malvertising**—
advertisements leading to sites that distribute mobile malware—both in-app and on sites accessed
during mobile web browsing sessions. And finally, of all the mobile threats we saw this half year, of
particular technical note is **Stels**, the first (but probably not the last) Android malware to be spread
via United States Internal Revenue Service (IRS)-themed spam distributed by the Cutwail botnet.

**IN REVIEW**

**SOURCES**

1. F-Secure Weblog; Sean Sullivan; **Timeline: Hacks Related to Apple**; published 20 February 2013:
2. Bitcoin Charts; http://bitcoincharts.com/charts/mtgoxUSD
3. F-Secure Weblog; Sean Sullivan; **Mac Spyware Found at Oslo Freedom Forum**; published 16 May 2013;
TOP 10 DETECTIONS AND THEIR TOP 10 COUNTRIES IN H1 2013, BY PERCENTAGE

- Majava, 45%
- Ramnit, 1%
- Sality, 1%
- CVE-2011-3544 exploits, 1%
- CVE-2013-1493 exploits, 2%
- ZeroAccess, 7%
- Autorun, 7%
- CVE-2011-3402 exploit, 10%
- Web-based attacks, 12%
- Downadup / Conficker, 14%

All other countries, 5%
WATERING HOLES

The most notable information security occurrence of early 2013 is undoubtedly the hacking and breach of several Internet giants (Twitter, Facebook, Apple, Microsoft) and of numerous other Silicon Valley companies via a watering hole at iPhone Dev SDK [6,7].

Significant, and yet, the general public appears to have learned few lessons regarding watering hole attacks as a result. Quite possibly because the companies involved kept and have continued to keep important details tightly under wraps.

So what happened?
The story started on February 1st with a blog post by Bob Lord, Twitter’s Director of Information Security, called: Keeping our users secure [2].

“...attackers may have had access to limited user information – usernames, email addresses, session tokens and encrypted/salted versions of passwords – for approximately 250,000 users.”

Lord stated that passwords and tokens were being reset – normal breach operating procedure. But then, at the end of his post, Lord suggested “users to disable Java on their computers in their browsers” and stated the attack was not the work of “amateurs”.

Lord’s post heavily suggested the cause of Twitter’s breach was due to a Java exploit via a browser. Knowing that Twitter has a high concentration of Mac users among its employees – we asked Apple for any related samples used in the attack – and were told that “Twitter hasn’t shared any samples”.

And then...
The story continued on February 15th when Facebook published a blog post called: Protecting People On Facebook [3].

“...Facebook Security discovered that our systems had been targeted in a sophisticated attack. This attack occurred when a handful of employees visited a mobile developer website that was compromised.”

Compromised by a zero-day Java exploit via a mobile developer website – which was later revealed to be iPhone Dev SDK. Facebook security chief, Joe Sullivan, stated in an interview with Ars Technica that several other companies were affected.

On the same day as Facebook’s post, Apple shared back door code (with no context) to an antivirus research mailing list. But the context seemed clear to us and we began to seek confirmation that the back door was related to the Facebook breach.

On February 19th, Reuters broke the news [4] that Apple itself had been hacked which was later followed on the 22nd by Microsoft’s blog post called: Recent Cyberattacks [5].
Criminals – but not crimeware
The key takeaway from all of the breach disclosures should have been this: a dedicated group of criminals had managed to hack numerous Internet companies via a watering hole. The attack was targeted and required human labor – it wasn’t automated crimeware.

But it didn’t need to be. For targets as valuable as Twitter, Facebook, Apple and Microsoft – the attackers were apparently more than willing to put in the man-hours.

What have we learned?
At this point, whenever we learn a website has been hacked – we ask ourselves this – was it just a regular data breach? Or would the site make for a good watering hole? But that doesn’t seem to be a question the public at large is asking.

At least not based on the recent news of NASDAQ’s community forum being hacked[^6] – but then… the news broke during the summer. Perhaps there’s more to that story.

Time will almost certainly tell.

---

**SOURCES**

2. Twitter, Bob Lord; Keeping our users secure; published 1 February 2013; https://blog.twitter.com/2013/keeping-our-users-secure
3. Facebook; Protecting People On Facebook; published 15 February 2013; https://www.facebook.com/notes/facebook-security/protecting-people-on-facebook/10151249208250766
4. Reuters; Jim Finkle and Joseph Menn; Exclusive: Apple, Macs hit by hackers who targeted Facebook; published 19 February 2013; http://www.reuters.com/article/2013/02/19/us-apple-hackers-idUSBRE91I10920130219
5. Microsoft; Recent Cyberattacks; published 22 February 2013; http://blogs.technet.com/b/msrc/archive/2013/02/22/recent-cyberattacks.aspx
6. NYTimes; Nicole Perlroth; Apple Computers Hit by Sophisticated Cyberattack; published 19 February 2013; http://bits.blogs.nytimes.com/2013/02/19/apple-computers-hit-by-sophisticated-cyberattack/
CASE STUDIES

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>STELS</td>
<td>14</td>
</tr>
<tr>
<td>ANDROID</td>
<td>18</td>
</tr>
<tr>
<td>APT ATTACKS</td>
<td>23</td>
</tr>
<tr>
<td>RANSOMWARE</td>
<td>26</td>
</tr>
<tr>
<td>CRYPTO CURRENCY MINING</td>
<td>30</td>
</tr>
<tr>
<td>EXPLOIT KITS</td>
<td>34</td>
</tr>
<tr>
<td>VULNERABILITY EXPLOITATION</td>
<td>36</td>
</tr>
<tr>
<td>KUMAR IN THE MAC (KITM)</td>
<td>40</td>
</tr>
<tr>
<td>PHISHING</td>
<td>44</td>
</tr>
</tbody>
</table>
STELS

Stels is an Android trojan that serves multiple purposes—it can turn an infected device into a bot that becomes a part of a larger botnet, and it can act as a banking trojan that steals mobile Transaction Authentication Numbers (mTANs). Stels made its debut in November 2012 and since then, different major versions of the trojan have been discovered in the wild.

F-Secure has identified over 1,300 unique Stels samples that can be divided into three major variants. The samples of each variant were first seen on 15th November, 28th November and 9th December 2012 respectively. Each variant was identified by looking into the package name used in the original source code: ru.beta, ru.stels2, and ru.stels4. The internal version numbers found in the code and configuration files further verify that the ru.beta variant is version 1 and ru.stels2 is indeed version 2. The newer versions of Stels do not appear to be replacing the old ones as all three variants have been encountered simultaneously.

**Main functionality**

The three Stels variants carry the same primary function, triggered when the trojan fetches commands from a command and control (C&C) server. The commands are communicated via HTTP connection, and delivered in encrypted JavaScript Object Notification (JSON) files, which will be decrypted and parsed by the trojan later. Different commands provide different methods for the bot owner to profit from, e.g., sending premium-rate SMS messages and stealing device information. Stels’ ability to intercept incoming SMS messages allows it to steal the mTANs that banks send to their customers via SMS messages, thus defeating the two-factor authentication method used to validate an online banking transaction. The full list of commands seen in version 1 of Stels can be seen in Table 1.

With the exception of ‘sendSmsLog,’ the same exact commands can be seen in version 2 of Stels, but in version 3 the commands appear to have undergone some changes, as can be seen in Table 2. While commands such as ‘subPref’ and ‘botId’ are not always present, all variants of Stels support commands relating to major functions, such as those that enable SMS theft and SMS sending.

Table 3 shows the list of C&C servers that Stels samples have been found to be communicating with. Each server is only in contact with one version of Stels, with a vast number of them contacting variants in version 2. There are probably several authors behind the different versions and variants of Stels, or at least more than one botnet with different owners.

In addition to contacting the C&C server, some variants of version 2 also use emails to keep in touch with the botnet owner. Once a week, these variants will email the following information to the attacker:

- C&C server’s URL
- Name of the backup Twitter account
- International Mobile Equipment Identity (IMEI) number
- International Mobile Subscriber Identity (IMSI) number

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>FUNCTIONALITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>botId</td>
<td>Change bot identifier</td>
</tr>
<tr>
<td>catchSms</td>
<td>Set which incoming SMS messages will be stolen (based on source phone number and message content)</td>
</tr>
<tr>
<td>deleteSms</td>
<td>Set which incoming SMS messages will be deleted (based on source phone number and message content)</td>
</tr>
<tr>
<td>httpRequest</td>
<td>Make an HTTP request</td>
</tr>
<tr>
<td>makeCall</td>
<td>Make a phone call</td>
</tr>
<tr>
<td>notification</td>
<td>Show a status bar notification</td>
</tr>
<tr>
<td>openUrl</td>
<td>Open a URL in the web browser</td>
</tr>
<tr>
<td>removeAllCatchFilters</td>
<td>Stop intercepting incoming SMS messages</td>
</tr>
<tr>
<td>removeAllSmsFilters</td>
<td>Stop deleting incoming SMS messages</td>
</tr>
<tr>
<td>sendContactList</td>
<td>Collect contacts from the phonebook and send them to the C&amp;C server</td>
</tr>
<tr>
<td>sendPackageList</td>
<td>Collect the list of installed applications from the phone and send it to the C&amp;C server</td>
</tr>
<tr>
<td>sendSMS</td>
<td>Send an SMS message</td>
</tr>
<tr>
<td>sendSMSLog</td>
<td>Get SMS messages from inbox and sent messages folders and send them to the C&amp;C server</td>
</tr>
<tr>
<td>server</td>
<td>Change C&amp;C server address</td>
</tr>
<tr>
<td>subPref</td>
<td>Change the suffix added to SMS messages sent by the trojan</td>
</tr>
<tr>
<td>twitter</td>
<td>Update the name of the Twitter account used to distribute a new C&amp;C server information</td>
</tr>
<tr>
<td>update</td>
<td>Download and install an application</td>
</tr>
<tr>
<td>uninstall</td>
<td>Uninstall an application</td>
</tr>
<tr>
<td>wait</td>
<td>Wait for a specified time before contacting the server again to receive new commands</td>
</tr>
</tbody>
</table>
TABLE 2: BACKDOOR COMMANDS SEEN IN VERSION 3 OF STELS

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>FUNCTIONALITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>call</td>
<td>Make a phone call</td>
</tr>
<tr>
<td>clearDeleteFilters</td>
<td>Stop deleting incoming SMS messages</td>
</tr>
<tr>
<td>clearListenFilters</td>
<td>Stop intercepting incoming SMS messages</td>
</tr>
<tr>
<td>deleteSMS</td>
<td>Set which incoming messages will be stolen (based on the source phone number and message content)</td>
</tr>
<tr>
<td>httpRequest</td>
<td>Send an HTTP request</td>
</tr>
<tr>
<td>install</td>
<td>Download and install an application</td>
</tr>
<tr>
<td>listenSMS</td>
<td>Set which incoming messages will be deleted (based on source phone number and message content)</td>
</tr>
<tr>
<td>message</td>
<td>Show a message dialog to open a URL</td>
</tr>
<tr>
<td>notify</td>
<td>Show a status bar notification</td>
</tr>
<tr>
<td>openURL</td>
<td>Open a URL in the web browser</td>
</tr>
<tr>
<td>run</td>
<td>Start an already installed application</td>
</tr>
<tr>
<td>sendContactList</td>
<td>Collect contacts from the phonebook and send them to C&amp;C server</td>
</tr>
<tr>
<td>sendDeviceInfo</td>
<td>Send device information to the C&amp;C server (phone number, operating system version, phone model and manufacturer, operator, if the device is rooted)</td>
</tr>
<tr>
<td>sendPackageList</td>
<td>Collect the list of installed applications from the phone and send them to the C&amp;C server</td>
</tr>
<tr>
<td>sendSMS</td>
<td>Send an SMS message</td>
</tr>
<tr>
<td>uninstall</td>
<td>Uninstall an application</td>
</tr>
</tbody>
</table>

TABLE 3: C&C SERVER ADDRESSES FOR DIFFERENT STELS VERSIONS

<table>
<thead>
<tr>
<th>VERSION 1</th>
<th>VERSION 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>kitherbin[dot]com</td>
<td>33files[dot]info</td>
</tr>
<tr>
<td></td>
<td>alpaybill[dot]net</td>
</tr>
<tr>
<td></td>
<td>androidfan[dot]name</td>
</tr>
<tr>
<td></td>
<td>bot[dot]mobiportal[dot]net</td>
</tr>
<tr>
<td>droidad[dot]net</td>
<td>bot[dot]mobile[dot]org</td>
</tr>
<tr>
<td>steamads[dot]info</td>
<td>i storhol[dot]ru</td>
</tr>
<tr>
<td></td>
<td>mach-nikolay[dot]info</td>
</tr>
<tr>
<td></td>
<td>mobiportal[dot]net</td>
</tr>
<tr>
<td></td>
<td>play-google[dot]mobi</td>
</tr>
<tr>
<td></td>
<td>pondelnet[dot]info</td>
</tr>
<tr>
<td></td>
<td>serviseru[dot]ru</td>
</tr>
<tr>
<td></td>
<td>skladchik[dot]in</td>
</tr>
<tr>
<td></td>
<td>ynfdbdybd1[dot]freeiz[dot]com</td>
</tr>
</tbody>
</table>

Social media as C&C backup

A problem with centralized botnets is that if the owner loses control of his C&C server, he also loses the ability to control the bots. The Stels author(s) attempt to combat this issue by setting up a few Twitter accounts for the bots check with to obtain a new C&C server address if the old one is no longer available.

In the first implementation, the Stels trojan parses the Twitter page, looking for encrypted messages that might contain the new C&C server address. In a later implementation, the encrypted messages are searched from the account description. The usernames ‘Vaberg1’ and ‘app36005565’ have been found to be used for the Twitter accounts used as part of the backup C&C system. However, the accounts do not appear to have been used before.

In May 2013, the botnet owner lost control of one of his C&C domains, droidad.net. He was forced to resort to the backup channel of a Juick account called ‘droidad.’ Figure 2 shows the account sending an encrypted message that relays a new C&C server address for version 3 of Stels trojan on May 24th. Once decrypted, the new domain was revealed to be steamads[dot]info.
Distribution

Android malware has thus far been mostly spread through third-party marketplaces, and Stels’ approach is no different. It has been found distributed through spaces.ru, a popular social network service that targets mobile device users in Russia, where it was repackaged with clean applications to lure victims. Bait applications that Stels author(s) use are mostly games. An interesting discovery was made during Q1 where Stels was found distributed via IRS-themed spam sent by the Cutwail botnet (http://www.secureworks.com/cyber-threat-intelligence/threats/stels-android-trojan-malware-analysis/). A user who clicked on the link on an Android device was directed to a web page asking him to update the Flash Player application. The ‘update’ which the user ended up with is actually the Stels trojan. In addition to using Flash Player’s name, Stels may also use a fake update under the name ‘Google updater.’ Figure 3 shows an example of a Stels infection when looking at the installed application list.

Conclusion

Stels is a flexible bot that offers the owner a fleet of different monetization avenues such as SMS messages, phone calls, software installation, or even acting as a banking trojan component that intercepts mTANs. The trojan is designed to last a long time, since the author has included a backup method for changing C&C server addresses in case the old servers become unavailable. We have seen instances where malware uses spam as a distribution method and uses social media as part of the C&C infrastructure, but those are Windows malware. The fact that Stels is using the same method is evidence that Android malware is advancing closer to reach the level of highly developed Windows threats.

Figure 2: Stels botmaster sending a new C&C server location via a backup Juick account

Figure 3: Stels trojan masquerading as a Google updater
MOBILE MALWARE

Since late 2011, we’ve seen a sharp increase in the total number of mobile device-targeted files or programs being distributed on all platforms. This is particularly true for Android applications (above), which saw a significant spike in the total number of APKs detected from March onwards, indicating an accelerating rate of new APKs being created for this platform.

Though simply counting the total number of programs we encounter produces impressive numbers, most of the files seen are simply replicas of a handful of unique families (programs that are grouped together based on similar codes or behaviors and detected using these shared characteristics). As such, a more meaningful gauge of significant development is to count the number of unique new families found on a platform, or new variants for known families, found within a year (left and below). These figures then allow us to, among other things, determine where malware developers are concentrating their efforts.
**ANDROID**

The Android platform continues to dominate the attention of mobile malware authors, with increasing interest in finding ways to target Android users while bypassing the Google Play Store. In this article, we take a look at trends we saw in malware developments on this platform in the second quarter of 2013.

**Android continues to dominate**

Android continues to maintain its dominant position in the smartphone operating system market, with the International Data Corporation (IDC) reporting that the platform accounted for 79.3% of all smartphone shipments in the second quarter of this year [1]. The next most popular platform is, unsurprisingly, iOS, which accounted for 13.2% of all new phones shipped in the same period. More surprisingly, this past quarter saw the Windows Phone platform overtake BlackBerry (3.7% to 2.9%) for the first time to become the third most popular smartphone operating system. All other platforms were unable to gain even as much as a 1% toehold in the market this quarter.

Given these numbers, little wonder then that malware development has focused on the Android operating system, with 96% of new malware families or variants we saw in the first half of 2013 found on that platform. In the second quarter of the year, 99% of new mobile threats we saw were on Android and on pages 20 to 21, we profile 6 of the most interesting malware we encountered during that period.

**Google Play Store**

While Apple’s App Store continues to be the strongest app market in terms of revenue generation [2] for its developers, it lost the title of the largest app market in July 2013, when Google announced that its Play Store had approved its 1 millionth app, surpassing Apple’s 900,000 total apps count at the time. That number doesn’t include the multitude of apps available on various third-party sites catering to Android users. Unfortunately, while the sheer volume of apps and markets available offers a range of choices for Android users, the dark side is that it also offers attackers an abundance of legitimate apps to compromise and trojanize, as well as varied ways to distribute their malwares.

In 2012, the Play Store introduced security measures that, among other things, identified malicious apps uploaded to the market. Though this has reportedly been effective, it doesn’t entirely eradicate risk - for instance, it doesn’t eliminate apps we would classify as Adware or Riskware, and is ineffective in blocking advertisements (either in the Play Store [3] or in the apps themselves) that users may consider unwanted, risque or outrightly malicious.

The intersection of advertising, privacy and security remains a grey area for Android, given the open ecosystem of the platform, the de facto use of advertising to monetize the mainly free apps in the market and the lack of a viable reputation-checking system for the advertising modules used in most of the programs. Despite the occasional hiccups however, thus far the Play Store is still considered the safest source for finding clean Android applications.

**Beyond the Play Store**

We believe the increased hardening of the Play Store has led to an increase in Android malware being distributed using other attack vectors. During Q2 2013, we discovered 205 new Android families or new variants of existing families, the majority of them from non-Play Store sources. Beyond the boundaries of the policed app market, malware developments of particular interest at the moment are: malvertising; drive-by downloads and mobile banking-trojans.
“Malvertising”: advertising malware
Malvertising seems to be increasingly used to distribute mobile malware, likely due to its ease of implementation and the wide reach it offers. This distribution method so far has typically involved either a compromised site injected with a malicious link or code, or simpler still, use of a third-party advertisement network to promote links to the malicious products, where the advertisements themselves may be taken and displayed in good faith.

One advertising strategy that leverages the reputation of the Play Store while avoiding the actual market’s security measures involves the use of fake Play Store advertisements. This social engineering technique uses a link or advertisement on a website that appears to be for an app in the market, but when clicked instead opens a non-Play Store site in the web browser.

We’ve also seen advertisements displayed during web surfing sessions on mobile devices that promote fake ‘mobile antivirus’ scams [4] - basically the mobile equivalent of similar efforts to distribute PC-based rogues. We also saw another ad-based technique used by Trojan:Android/Badnews.A, where in-app advertisements promote what turned out to be a fraudulent app for a premium SMS subscription.

Mobile drive-by downloads
We’ve also encountered drive-by downloads of mobile malware while browsing compromised sites on an Android device [5]. Unlike PC-based drive-by downloads, these attacks are still relatively visible to the user, as they automatically trigger a notification message asking whether the user wants to install the app (and providing an opportunity for the user to delete the unwanted program).

Again, since this attack vector involves either compromising a legitimate site or redirecting users to a malicious site (both activities likely to be familiar to any malware distributor with prior experience spreading PC-based malware), we expect to continue seeing malware distribution using this method in the future.

Mobile banking trojans
A more direct threat to users comes in the form of mobile banking-trojans - typically malware that attempt to steal SMS messages containing Mobile Transaction Authentication Numbers (mTans). The proliferation in this type of mobile malware has been in response to a general shift by banks providing online shopping facilities to take two-factor security authentication into use. This ties in with the discovery this quarter of Trojan:Android/Pincer.A, which we believe to be related to fake banking security apps that were reported in July targeting users of the Commonwealth Bank [6], as well as Trojan:Android/FakeKRBank.A, which we saw targeting users in Korea.

And finally, in Q2 2013, we observed more mobile banking-trojan toolkits - that is, programs that automate the process of creating these malware - are now being sold in the underground markets. As such, we expect to see more of this kind of threat in the near future and perhaps evolve further.

SOURCES
5. FS Labs Youtube Channel; Drive-by Android Malware; published 3 May 2013; [http://www.youtube.com/watch?v=aYFX8V7ExbA](http://www.youtube.com/watch?v=aYFX8V7ExbA)
The following 6 Android malware were selected as representative of the mobile threats discovered during Q2 2013.

In addition, where available, the Protection Network Count counts the times a device protected by F-Secure’s Mobile Security reported blocking an attempted malware installation to our cloud-based telemetry systems during Q2 and Q3 of 2013. Comparing the low rate of blocked mobile infections to the vastly greater rate of blocked infections for PC-based threats can arguably indicate both the currently rare nature of mobile infections and the effectiveness of mobile antivirus solutions in protecting a device from infection.

**TROJAN:ANDROID/BADNEWS.A**

- **Count of Known Unique Samples:** 113
- **Oldest Known Sample Date:** 2013-03-03

**Distribution**
Badnews variants have been found on the following online app markets: Google Play, Opera, Baidu, Anzhi and Liqucn

**Summary**
Badnews variants are fraudulent premium SMS-trojans. On installation, the app displays advertisements for a premium-SMS subscription.

**TROJAN-DROPPER:ANDROID/FAKEKR_BANK.A, TROJAN:ANDROID/FAKEKR_BANK.A**

- **Count of Known Unique Samples:** 64
- **Oldest Known Sample Date:** 2013-05-20

**Distribution**
Unknown

**Summary**
Once installed, Fakekrbank intercepts SMS messages and forwards the messages via URL POST to a remote server, together with the device’s name and number.

**TROJAN:ANDROID/VMVOL.A**

- **Count of Known Unique Samples:** 43
- **Oldest Known Sample Date:** 2013-03-28

**Distribution**
Unconfirmed in Google Play, records shows it was removed.

**Summary**
Once installed, VMVOL.A sends the device’s phone number and International Mobile Equipment Identity (IMEI) number to a remote server. It also sends a copy of the messages sent, including the message content and the receiver’s number.
**TROJAN:ANDROID/FAKEDEFENDER.A**

**Count of Known Unique Samples:** 14  
**Oldest Known Sample Date:** 2013-03-24

**Distribution:**  
Seen advertised in third-party advertisements displayed on mobile devices.

**Summary**  
Similar to rogue antispyware programs found on PCs, Fakedefender is a rogue anti-spyware program for the mobile device. The program does not provide the scanning or malware removal functionalities claimed.

**Protection Network Count**

![Protection Network Count](image1)

---

**TROJAN:ANDROID/PINCER.A**

**Count of Known Unique Samples:** 11  
**Oldest Known Sample Date:** 2013-03-19

**Distribution**  
This app is known to be distributed either on phishing sites (typically imitating a bank site) or via malicious links in SMS messages. It may also be distributed as part of the payload of a PC malware.

**Summary**  
Once installed, Pincer forwards SMS messages to a remote Command and Control (C&C) server. It is also able to function as an mTan-stealing banking-trojan, as well as carrying out instruction given remotely by the C&C server.

---

**TROJAN:ANDROID/OBAD.A**

**Count of Known Unique Samples:** 6  
**Oldest Known Sample Date:** 2013-05-16

**Distribution**  
Obad variants were observed being advertised on a malicious website while browsing on an Android device and is likely to arrive on a client device via a mobile drive-by download.

**Summary**  
Once installed on the device, Obad variants gain administrator privileges and uses a exploit to break through the Android operating system’s security layer. Obad collects and sends the following details about the device to a remote C&C server: the Media Access Control (MAC) address and IMEI, the operator name, the time and root access. The C&C server is also able to issue the commands to the installed application, including to send SMS messages, make the device act as a proxy or a remote shell, launch a URL in the mobile browser, download and install additional components, get the Contacts list as well as further details of a specific installed app and send a file via Bluetooth.

![Protection Network Count](image2)
**Securing the device**

Today most people have their email accounts (personal and/or work) and other critical services on their mobile devices. This convenience also means that if your device is lost or stolen, your losses could involve more than just the physical device.

And despite concern about online-based attacks, the easiest way for malware to get on a device is still for someone to manually install it while the device is in its possession. In other words, protect your device’s physical security first.

1. **Lock the device.**
   - Locking your device prevents anyone else from meddling with its settings and installing an app (such as a monitoring-tool or spyware) while it is out of your possession. For the lock to be effective, make sure the password/passcode/pattern is unique, and preferably memorable for you without being easy for someone else to guess.

2. **Set up anti-theft protection.**
   - Anti-theft protection typically provides you the ability to remotely wipe the data on your phone, including on any memory cards installed, if you decide your phone is irretrievable.
   
   Some anti-theft solutions also include features such as location mapping or sounding the alarm, to help when attempting to locate the device.

**Blocking unwanted services**

Lucrative profit-generating mechanisms for mobile malware are to silently send premium rate SMS messages, subscribe the user to continuous premium services, or to force the device to call premium-rate numbers. Blocking premium calls or messages is one way to minimize financial losses, even if malware does get installed. This also provides protection against non-malware billing fraud by “operators” who silently subscribe users to premium services and forward billing requests to the user’s mobile operator, hoping to have the charges quietly added to the user’s bill.

3. **Set up call or message barring.**
   - Most operators allow users to set up a call or SMS barring service to block the device from sending unwanted calls or messages. Also known as ‘premium-rate blocking’, this is particularly useful for parents who want to prevent their children’s devices from inadvertently incurring unnecessary charges. To set up this service, contact your phone operator for more details. Some services also provide a PIN number or other method that allows the user to selectively remove the barring, if they desire.

**When downloading apps**

Once your device’s physical security has been addressed, you can also take the following steps when downloading an app.

4. **Download apps only from the Play Store.**
   - By default, Android devices block installation of apps from any source other than the Play Store. You can check to make sure your device only allows Play Store apps by looking in **Setting > Applications > Unknown sources**. If the checkbox is checked, non-Play Store apps can be installed. Uncheck this.

5. **Check the apps’ permission requests.**
   - Whether you’re downloading from the Play Store or other sources, make sure to read the app’s list of requested permissions (the ones that typically raise eyebrows due to security or privacy concerns are listed below right).
   
   If the permissions requested seem excessive or unrelated to the app’s purpose—for example, a casual game asks to send SMS messages—you can check the developer’s references for more details, as reputable developers usually explain why the permissions are needed. If the use appears justified to you, then you may elect to download the app.

   Incidentally, apps such as PocketPermissions, LBE Privacy or PermissionDog can be useful guides for explaining the sometimes-obscure permissions. Some also include features to restrict permissions used by installed apps, though such functionality is often intended for advanced users.

6. **Scan apps with a mobile antivirus.**
   - Once downloaded onto your device, use a reputable mobile antivirus to scan the app. You can think of this as a check on the app’s ‘silent’ behavior—permissible actions that are implied in the app’s permissions list (for example, sending the device’s details to a remote server) but may cause users concern. If the verdict from the mobile antivirus is acceptable to you, then you may proceed to install the app.

**While online**

As websites have evolved to cater for visitors browsing from mobile devices, we’ve also seen malicious sites follow suit(1).

7. **Use web browsing protection.**
   - To avoid stumbling onto a malicious site while surfing on a mobile device, use web browsing protection (available from most antivirus solutions) to block known harmful sites.

---

**Sources**

APT ATTACKS

Advanced Persistent Threat (APT) attacks typically involve a carefully crafted exploit document being delivered (usually through some form of social engineering) to a user, or users, in a targeted organization or industry. During 2012 and 2013 there was a lot of talk about various individual APT campaigns and significant individual cases. While these individual cases are interesting, we felt that it would be useful to gain an overall picture of the kind of victims various APT attackers are targeting.

In order to gain insight into the larger picture, we randomly selected and analyzed 100 APT documents from our sample collection. The sample documents were categorized by language, topic of the document and if identifiable, the national focus - that is, the country it appeared to be addressing. While the contents of an APT document do not directly identify the attacker, the information gathered from its analysis gives a good indication of the kind of victims being targeted, as the document must be sufficiently interesting bait for the potential victim in order for the attack to succeed.

Languages other than English were translated with computer translation, resulting in a less than perfect understanding of the documents; this was however good enough for classification purposes. We also extracted text from the first pages of all the sample documents and visualized the text in a word cloud (see page 24) to give an overview of the most common words used in the ATP documents.

Based on the statistics generated from our analysis, we can see that English is the most common language in APT documents, which makes sense both from the attacker’s and victim’s perspective, as English is commonly used in international business and politics, so that the attacker need not produce lure documents in the victim’s own, potentially non-English language.

Political attacks
As can be seen from both the word cloud and statistics derived from our analysis, the most common type of content in ATP documents is political, usually in the form of a report of some politically interesting event, development or longer term review. All in all, the content of APT bait documents look like the type of content one would expect to be interesting to an embassy officer, government employee or otherwise politically active or aware figure.

The most common topic in the political documents deal either with China or India; USA-related topics are also very

### Language Used in APT Documents

<table>
<thead>
<tr>
<th>Language</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>75</td>
</tr>
<tr>
<td>Chinese</td>
<td>12</td>
</tr>
<tr>
<td>Arabic</td>
<td>5</td>
</tr>
<tr>
<td>Persian</td>
<td>3</td>
</tr>
<tr>
<td>Russian</td>
<td>2</td>
</tr>
<tr>
<td>French</td>
<td>2</td>
</tr>
<tr>
<td>Vietnamese</td>
<td>1</td>
</tr>
</tbody>
</table>

### National Focus of APT Documents, Mapped *

* Only nations that were targeted by more than one APT document in the sample set are mapped.
prominent. When reading the documents, it seems that many of them mostly cover relations between USA and China, or India and China. Extrapolating from that, it seems that the most common target for a politically-motivated APT attack is someone who is from or has some form of affiliation to the USA or India, and is interested in China.

**Corporate attacks**
Corporate users are mostly targeted with bait documents that look like conference proceedings or reports. This does make sense as conference proceedings are normally propagated by e-mail as part of standard business practices anyway, making them easy for the attackers to obtain, modify and pass on as ‘revised’ editions. The second most common type of corporate-targeted ATP documents were reports, which are also relatively easy to obtain and are easily passable as credible business material. As can be seen from the corporate word cloud, the aerospace and energy sectors seem to be of most interest to the spies. As with political documents, the corporate-targeted documents commonly contain references to China or other Asian countries, which would indicate that attackers are interested about targets that operate in the aerospace or energy sector and have some form of contact with Asian countries.

**Military attacks**
As with other victim categories, a large portion of APT documents targeted to the military were reports on topics that would be of interest to military personnel. What was interesting however was that in addition to strictly military-related topics, a major portion of these documents related to the personal finances of service members, such as tax information or military pension details. In these cases, the
attackers attempted to attack victims using documents of personal rather than professional interest, which is an angle that we have not seen with any other category.

**Academic attacks**
The number of ATP documents in our sample set targeted to academics was too small and scattered to draw any major conclusions. Based on the handful of samples however, it does seem that either the attackers are trying to lure victims with either material related to a particular academic field or administrative topics such as curriculum vitae, memos or forms.

**Conclusion**
In order to be able to publish results for this report, we used APT documents that were obtained from public sources, which means that our sampling is most likely biased as it is very likely that many corporate targets consider any APT material they encountered to be so sensitive that they either do not share the documents or share them confidentially, thus preventing their use for a public study.

Even so, based on the material we are able to share, we can construct a basic profile for the people targeted for APT attacks: primarily those working with political material, especially those whose work includes contact with China or other major players in the Southeast Asian region. While corporations, the military and academia receive less attention, it is still true that as long as you possess information of interest, it is likely that you may be a potential target for APT attack.

**TOPICS APPEARING IN MORE THAN ONE APT DOCUMENT**

<table>
<thead>
<tr>
<th>Report</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Letter</td>
<td>7</td>
</tr>
<tr>
<td>Conference</td>
<td>6</td>
</tr>
<tr>
<td>Service related finances</td>
<td>4</td>
</tr>
<tr>
<td>Aerospace</td>
<td>4</td>
</tr>
<tr>
<td>Activist</td>
<td>3</td>
</tr>
<tr>
<td>Sex scandal</td>
<td>3</td>
</tr>
<tr>
<td>Curriculum Vitae</td>
<td>3</td>
</tr>
<tr>
<td>Diplomatic car ad</td>
<td>2</td>
</tr>
<tr>
<td>Banking</td>
<td>2</td>
</tr>
<tr>
<td>List of contacts</td>
<td>2</td>
</tr>
</tbody>
</table>
Ransomware is a type of malware that uses extortion to get money from victims whose computers have been infected. The malware typically takes control of the victims’ data or computer and demands a payment for the victims to reclaim their possession. We last wrote about ransomware problem in our H1 2012 Threat Report. Since then and during the first half of 2013, this threat has remained very active.

Two types of ransomware

It is useful to compare the two different types of ransomware that are currently infecting users. The only common factor between these two types is the concept of extorting money from victims by holding something valuable — data or the computer itself — hostage. Everything else is very different, as we will soon discover.

The first type is called the encrypting ransomware. When this malware infects a computer, it encrypts the user’s data and leaves a ransom note that asks for money in exchange for a cryptographic decryption key needed to restore the files.

The second type is a police-themed ransomware, often simply referred to as police ransomware. On an infected computer, this malware shows a “lock screen” which requests for a ransom payment for the computer to be unlocked. The lock screen is simply a dialog box that fills the whole computer screen. It prevents the user from closing the box and quitting the malware.

Encrypting ransomware

Encrypting ransomware’s infection happens in an unusual way. Most malware infect a computer through a drive-by exploit, where they infiltrate the machine by abusing a vulnerable application like Java. Their target is often a personal computer, especially the users with poor browsing habits.

Encrypting ransomware, on the other hand, often targets corporate Windows servers, where it encrypts all data files and asks the company to make a payment in order to decrypt them. These servers are not meant to be used for browsing and reading e-mails. In many infection cases, the server was found to be running a Remote Desktop (RDP) service and the user accounts were guarded by weak passwords. The attackers employed a password guessing attack and managed to log in using the RDP service.

PayLoad

The payload of encrypting ransomware is usually very simple. Once the attacker has gained access to the corporate server, he manually executes the malware. The malware will enumerate files on the server, encrypting all data files but skipping any system files to avoid making the system unbootable. The encryption is done using strong cryptography, and once the operation is completed, the encryption key will be securely erased. Every file on the network shares is also encrypted; if backups are accessible to the server, these are also encrypted.

Payment mechanism

To decrypt the files, a decryption key is needed. The victims will receive information on how to deliver the payment to the attacker in exchange for the key. In many cases, the payment is made using online payment systems such as Moneypak, Paysafecard or Ukash. The amounts go from USD 500 onwards.
In a typical situation, the attacker will ask the victim to send the encrypted file, which will be returned in a decrypted format. The victim is also required to send a proof of payment over e-mail. Since the attacker has to keep track of the victims and the keys used to decrypt their files (each victim is assigned a different key), the manual work involved in this correspondence limits the amount of victims that can be targeted simultaneously. Attacking a million targets at once is not possible. This limitation may explain the “waves” of attacks — the attackers target a limited set of servers in one country and then move on to the next one.

**Police ransomware**

Police ransomware gets its name from the “lock screen” that appears on an infected computer. The lock screen displays a message allegedly coming from the local law enforcement, claiming the computer’s involvement in illegal activities such as copyright violations and child pornography distribution.

A handful of families are behind most of police ransomware infections. Two of the most notable ones during the past six months are Reveton and Urausy. Both families infect users via exploit kits when they are browsing the web. The users will be redirected to a web server that automatically infects the computer using an exploit.

There have been cases where malicious advertisements (malvertising) were used on popular websites such as MSN Italy to redirect users to exploit kits serving ransomware. The most common method still seems to be malicious advertisements on adult websites.

<table>
<thead>
<tr>
<th>Encrypting ransomware</th>
<th>Police ransomware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypt user data and demand a ‘ransom’ in exchange of the decrypting key.</td>
<td>Display a ‘lock screen,’ pretending to be a local law enforcement. Demand a ‘fine’ to unlock.</td>
</tr>
<tr>
<td>Corporate Windows servers running a Remote Desktop (RDP) service.</td>
<td>General users’ computers.</td>
</tr>
<tr>
<td>Perform password guessing attacks to break in and log in using RDP.</td>
<td>Redirect users to servers hosting exploit kits; users’ computers will be automatically infected.</td>
</tr>
<tr>
<td>Enumerates all files on the server, encrypting all data files except for system files. Encryption key is securely erased after the operation is completed.</td>
<td>Has a launch point in registry that starts malware automatically after system restart, usually in ‘safe mode.’ The lock screen shows up immediately.</td>
</tr>
</tbody>
</table>

**Geographical focus**

Police ransomware uses localized messages to convince infected users. Both Reveton and Urausy target several countries such as the US, Germany, France, Saudi Arabia and Italy (see Figure 2).

Unlike encrypting ransomware, the lock screen on police ransomware is localized with the national law enforcement logo and often translated to the local language. The current list of localization that Urausy supports include:
Argentina, Australia, Austria, Belgium, Bolivia, Canada, Croatia, Cyprus, Czech Republic, Denmark, Ecuador, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Jordan, Latvia, Lebanon, Luxembourg, Malta, Mexico, Morocco, Netherlands, New Zealand, Norway, Palestine, Poland, Portugal, Romania, Saudi Arabia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey, United Arab Emirates, United Kingdom, United States and Uruguay

The cost involved in localizing the malware may have caused the ransomware families to steal the design of the lock screens from each other, thus, making it difficult to tell one family’s infection apart from the other by looking at the screen.

Payload

Similar to encrypting malware, the payload for police ransomware is quite simple too. After infecting a computer, the malware will wait for several minutes before displaying the lock screen. This delay makes it more difficult for the victims to identify which visited website might have been responsible for the infection.

The malware has a launch point in the registry to make sure that the malware starts its payload automatically when the computer is restarted, usually in “safe mode.” The locked screen will show up immediately after restart, making removal difficult.

The Reveton ransomware also includes a second payload that allows the attacker to benefit from victims who refuse to fall prey to paying for the ransom. As an insurance, the second payload automatically steals any username and password that has been saved in the web browser, FTP clients and other places. The attackers may then use or resell these credentials for financial benefit.

Payment mechanism

Unlike encrypting ransomware, nearly all police ransomware families have a fully automated payment collection. On the lock screen, the victim will be asked to input a PIN code for an online payment system like Moneypak, Paysafecard or Ukash. This PIN code is then automatically sent to a command and control (C&C) server for verification. The C&C infrastructure consists of several proxy servers in an attempt to hide the criminals’ track. Because of the level of automation, criminals using this type of ransomware are able to target thousands of victims and handle the payment without any difficulty.

How to protect against ransomware:

- Enable Browsing Protection and DeepGuard in your F-Secure product to stay safe from malicious websites
- Keep software up-to-date to stay safe from exploit kits
- Uninstall Java if not needed, or keep it installed in a secondary browser that is not used for general web browsing
- If infected with police ransomware, change all passwords that were saved on the computer
- Make regular backups to recover from encrypting ransomware. Make sure the backups are not accessible from the server itself

Encrypting Ransomware VS. Police Ransomware Operation Flowchart
Ransomware and BitCoin in Asia
For the most part, most of the ransomware cases we detected were in Europe, the Middle East and most parts of the United States.

However, around the Asian region, there have not been too many cases of exploits utilizing ransomware to extort money from victims, most likely since the region has a lack of convenient and anonymous payment systems such as Ukash, MoneyPak and Paysafecard. In Europe, these systems make it easy for a victim to obtain a prepaid card from a convenient store to pay the ransom (while not informing the authorities of the extortion, especially in cases of police-themed ransomware). The receiving party on the other end can also be easily trained without much personal or unique information being given out, and with automation, is able to collect the ransom from hundreds and even thousands of victims.

In Asia however, most countries have not been set up with this sort of electronic payment system (probably due to the fact that much fraud occurs in this part of the world) and most victims so far have relied on online banking methods to transfer funds electronically.

In future however, with Bitcoin on the rise and its acceptance in Asia due to the pseudo-anonymous nature of the currency, could it possibly drive malware writers to conquer this part of the world with ransomware attacks tailored for collection of the ransom using Bitcoin?

China is currently the second biggest and fastest growing Bitcoin market in the world today, second only to the United States. China is no stranger to virtual currencies, as they were exposed to the Q coin in the mid-2000s by Tencent (which all started with avatar purchases online) and online gaming. Because China’s strict cross-border financial controls make it difficult to spend money in stores abroad, Bitcoin could be the perfect virtual currency for Chinese Bitcoin owners, as they can convert them into currencies that sites like Paypal and Amazon would accept.

Since Bitcoin is so decentralized, it would be extremely difficult for the government to handle any issues related to Bitcoin use, as they cannot track it, block it or crack down on it. It would be the perfect way for cybercriminals to safely move and launder money collected from ransomware and evade authorities.

The PC isn’t the only platform likely to suffer from ransomware in the future. The increase in Android users, especially in Asia, has lured malware writers into branching out from the PC to the mobile platform, leading to increasing sophistication and maturity in the malware targeting Android users. Most relevant is the increase in fake antivirus and ransomware attacks making their way onto the mobile platform. Once rogue antiviruses are installed on your mobile device, they display a false malware detection in an effort to con the user into paying for software that would supposedly remove the infection, making them at least spiritually similar to ransomware.

China is currently the second biggest and fastest growing Bitcoin market in the world today, second only to the United States.

Given that Asia is one of the largest markets of smartphone users and a major market for free apps downloaded through third-party stores, it is likely that in future the region will see an increase in malware writers finding more creative ways to introduce fake antivirus or ransomware onto a user’s mobile device.
Crypto currency mining operations have been running for years. The top two currencies, Bitcoin and Litecoin, have become a holy grail to cyber-criminal organizations that need a stream of hard-to-trace income. As a decentralized/peer-to-peer currency, Bitcoin is exactly what they have been waiting for.

**Bitcoin Malware Timeline, 2008–2012**

Back in 2008, a mathematician named Satoshi Nakamoto (a pseudonym) submitted a technical paper for a cryptography conference. In the paper titled “Bitcoin: A Peer-to-Peer Electronic Cash System,” he described a peer-to-peer network where participating systems would perform complex mathematical calculations on something called a “blockchain.” The goal is to create a completely new currency — crypto currency — that is based on math.

Bitcoin is not linked to any existing currency; therefore, its value is purely based on how much people believe it is worth. Some vendors do accept Bitcoin as a form of payment. To send Bitcoin to another party is similar to sending an e-mail. All it needs to transfer money instantly, bypassing exchanges, banks and taxes, is an address linked to somebody’s wallet file. In a way, crypto currency would make banks completely unnecessary for moving money around, which is why banks hate the whole idea of Bitcoin.

In 2010, when Bitcoin had just started to get adopted by enthusiasts, its initial price was listed at merely several dollars. At that price, many doubted that it would make a good investment. But over time, as the media started to pick up on the subject, its price began to shoot up. This undoubtedly caught the attention of the bad guys.

The first batch of Bitcoin malware, which utilized infected computers’ CPU for proof-of-work calculations, was discovered in 2011. The calculations, also referred to as “mining,” rewarded the malware authors with coins. However, such tactic did not provide enough income as the authors had originally hoped. By Q2 2011, those malware had lessened in number. The authors figured out that it would be more profitable to simply sell access to infected machines rather than mine the coins themselves.

Within a short period of time, the malware authors managed to adapt the code to enable a more efficient way of mining. In Q4 2011, some malware families, such as the Trojan.Badminer reported by Symantec[1], began to utilize third-party mining programs that specialize in GPU-accelerated hash calculations. Using the GPU is 50 times more profitable than using a top-notch CPU, and it soon became the most common mining method amongst cyber-criminals.

Commonly used mining programs include bfgminer/cgminer, cudaminer and Ufasoft’s Coin. These programs are all open source and the malware authors stick to using the public builds as much as possible to evade targeted signature detections.

Some cyber-criminal gangs opt to use a custom mining tool, often delivered by a third-party developer who prefers to stay in the “grey zone.” Examples of these tools are Silent Miner, Chrome Miner and Insidious Miner. Silent Miner still has an up and running website, where it is being sold for money through PayPal or Bitcoins via Mt. Gox.

**Bitcoin Malware Timeline, H1 2013**

In 2013, different Bitcoin malware began to emerge, as reported by various anti-virus companies. The most remarkable ones are as follows:

- **April 2013** - A scam message containing a malicious link spread out amongst Skype users. The link led to a malicious executable that would download additional components from a command and control server. One of the identified components was a Bitcoin miner tool that would run silently in the background.
- **May 2013** - Bitcoin malware leveraged a Proxy Auto-Configuration (PAC) file to redirect infected victims to a fake Mt. Gox website. The victim’s Internet browser would contact the criminal’s proxy server as defined in the PAC. So, when visiting the original Mt. Gox website, the victims would instead be directed to the fake website hosted on the criminal’s server[2].
June 2013 - Cyber-criminals distributed spear-phishing spam that led to a fake Mt. Gox website. The URL of the fake site contained a similar second-level domain name “mtgox” but different top-level domain names - “.org,” “.net,” “.co.uk” and “.de.” This trick was easier to recognize compared to a malicious PAC.

Late June 2013 - Researchers from Webroot uncovered the latest DIY Bitcoin mining tool in an underground market. The Bitcoin miner generated by this DIY tool was claimed to be stealthy and invisible. Based on our analysis of the sample, the criminals were using the Ufasoft Bitcoin miner, which was freely available on the official website.

ZeroAccess Botnet Estimated Profits

Up to this day, the largest botnet mining operation was carried out by the ZeroAccess family. The malware consists of a powerful rootkit to hide its presence and so-called “plugins.” While slower computers are useful enough for fraudulent pay-per-click operation, the faster ones will occasionally receive an additional plugin to perform mining tasks.

To calculate how much profit the mining operation can bring, let’s consider these few assumptions:

- According to a survey conducted by Valve Corporation in May 2013, the most common graphic cards among gamers are NVIDIA 400/500/600 series and AMD 5000/6000 series, which produce ~225 Mhash/s out of a typical gamer’s PC.
- Close to 100,000 PCs are suspected of being infected by ZeroAccess daily. Monthly, the botnet consists of ~3 to 5 million active installations; there could be ~2 million machines directly involved in the mining operation (half of the botnet) on a certain time frame.
- If 5% of these (probably more) are gamers’ machines, a simple calculation gives us 22,500,000 Mhash/s

\[(2 \times 10^6) \times 0.05 \times 225 \text{ Mhash/s} = 22.5 \times 10^6 \text{ Mhash/s}\]

With a monthly profit of USD 1.7 million out of 5% top victims, ZeroAccess-powered mining operation is undoubtedly a very lucrative option to cyber-criminals.

ZeroAccess Bitcoin Plugins Back-to-Back Update

ZeroAccess’ botnet operator has recently been performing multiple updates on the plugins responsible for carrying out Bitcoin mining payloads. Interestingly, even though the Bitcoin plugin is still actively updated and pushed to infected peers, the Bitcoin mining payloads do not exist in the binary file.

Based on the samples that exist in our backend systems, we have come up with an update history of the Bitcoin plugin binary file. The release date of the binary is presumably determined from the timestamp of the binary compilation update. An update is usually released every month, but there are a few instances where the botnet operator only releases a new binary after two months. And it uses its own private pool server to distribute a cryptography block for its workers—infected PCs running the mining plugin—to solve. The pool server used by the Bitcoin plugin released in August 2012 was identified as “google-update[dot]com.” From September 2012 to April 2013, the botnet operator released a plugin binary that does not contain the mining payloads.
On 2nd April 2013, the botnet operator released a new Bitcoin plugin binary and reintroduced the Bitcoin mining payloads using a different private pool server, “ooyohrmebh9qfof[dot]com.” This plugin binary, with the mining payloads, did not last long. The screenshot in Figure 3 shows the comparison between the older and the latest version of the Bitcoin plugin built in the same month. The file on the right shows the latest plugin, which was built on 23rd April 2013. The yellow highlighted rectangle shows that the embedded UPX packed Ufasoft’s Coin miner binary has disappeared. In other words, starting from 23rd April 2013 until today (at the time this article is written), ZeroAccess does not carry Bitcoin mining payloads.

The absence of the mining payloads however, does not mean that the botnet operator has stopped generating revenue from its botnets. The click fraud payloads that reside in the same plugin binary file are still functioning. The reason why the botnet operator stopped Bitcoin mining payload despite the lucrative BTC exchange rate is a mystery. But here are some
assumptions that can be made:

- It is too expensive to setup and maintain its own private pool server
- It is too invasive and noisy to run a Bitcoin mining operation on an infected machine
- It is not easy to mine a Bitcoin without using a dedicated machine, especially when the botnet operator turns off GPU mining by default

Conclusion
It is hard to estimate the worldwide infection rate due to the use of commonly available mining tools, which may or may not be installed with the users’ consent. But we have managed to take a total count of incoming Bitcoin mining-related samples for every month in the first half of 2013.

Not surprisingly, the graph matches (with slight delay) with the BTC trade prices reported by Mt. Gox exchange.

Since the introduction of Bitcoin in 2009, there has been two types of Bitcoin malware attacks. The first one involves malware variants that steal Bitcoin wallet files from infected victims. The second type involves variants that run legitimate Bitcoin mining tools on infected machines, allowing cyber criminals to collect Bitcoin without paying for hardware or electricity. Apparently, the cyber criminals gave up the first type of Bitcoin malware attack because of the extra security feature implemented in the Bitcoin client. The feature encrypts the "wallet.dat" file, making it valueless to cyber criminals that did not "key-log" the password key.

Based on the current trend, it is not surprising to see more malware leveraging the type-2 method in the future to earn a penny. However, it is hard to estimate the profits as they spread differently and often utilize “p2pool mining” where each infected PC is kind of “solo-mining” to a specified wallet address. Typical infection vectors that may be utilized by the malware include Skype or other instant messaging services, e-mail, drive-by-downloads and exploits.

BITCOIN MALWARE GROWTH

Source: virustotal.com

Source: mtgox.com
EXPLOIT KITS

In the last couple years, exploit kits have become a popular, convenient tool for attackers and malware distributors to efficiently find and compromise new victims. In H1 2013, we saw increasing development and competition among exploit kits, as new contenders emerged and older, established toolkits expanded their capabilities with new vulnerability targets and new tricks to avoid detection.

Top-5 exploit kits
According to F-Secure’s telemetry data, 70% of all exploit-kit related detections encountered by our clients in the first half of 2013 are the work of 5 exploit kits: Blackhole, SweetOrange, Crimeboss, Styx and Cool. 28 other exploit kits account for the remaining 30% of exploit-kit detections.

Of the top five kits, Blackhole by itself was responsible for 31% of all exploit-kit related detections globally reported to our cloud-based systems during this period. This exploit kit’s behavior was first detailed in our H1 2012 Threat Report and one year on, has changed little except for incorporating exploits for recently announced vulnerabilities, such as CVE-2013-2423 and CVE-2013-0422 (see Vulnerability Exploitation article on page 36), and tweaks to the URL patterns used for its exploit sites.

Of the other exploit kits in the top five, SweetOrange is the closest competitor to Blackhole, accounting for a comparatively low 11% of all exploit-kit related detections seen worldwide. The geographic distribution of the clients that saw these detections (chart at right) indicates that though most of the top five kits had some presence in every region (particularly Blackhole, which is practically omnipresent), the Crimeboss exploit kit focused almost exclusively on the Americas, where it accounted for 31% of all exploit-kit detections there.

New entrance
In the span of six months, we saw seven new exploit kits emerge (see timeline at bottom), as well as the updating of an older kit, CritXPack aka SafePack aka FlashPack, with new names. Though these kits have had relatively modest success compared to Blackhole or even SweetOrange, the increasing number of active kits in play has simply given more options for malware distributors to carry out their work. Of particular note among the new batch is Whitehole, which focuses on exploiting the Java development platform.

In H1 2013, 31% of our clients worldwide reported a Blackhole-related detection, making it the most globally prevalent exploit kit.
Targeting recently publicized vulnerabilities

It’s been noted before that exploit kit authors have been very prompt in including new exploits for recently announced vulnerabilities into their products, and H1 2013 has been no different. Case in point involved the Java vulnerability CVE-2013-2423; a Metasploit module targeting this was first published on April 20th, and a day later we noticed in-the-wild attacks against it had already gotten underway by the CrimeBoss exploit kit.

We also saw a new exploit kit, Private, become the first to target the CVE-2013-1347 vulnerability in version 8 of Internet Explorer. This particular vulnerability gained public notice due to a US Department of Labor website compromise in early May 2013; a patch was released by Microsoft just days later. The following month, we encountered the Private Pack exploit kit on June 11, already busy targeting this vulnerability.

As of July 2013, at least 5 vulnerabilities which were only publicly announced during the past half year have been incorporated into various exploit kits. Four of the five vulnerabilities—CVE-2013-0422, CVE-2013-0431, CVE-2013-1493 and CVE-2013-2423—have been related to the Java development platform. These exploits are identified by various detections in our security products, including the specific detections Exploit:Java/CVE-2013-2423.A and Exploit:Java/CVE-2013-2423.B, and a powerful generic detection, Exploit:Java/Majava.C.

New tricks employed

Given the consistent focus of multiple kits on exploiting Java, during this six month period Oracle has unsurprisingly been busy releasing updates to close the vulnerabilities under attack and to generally increase the security of the product. Of particular note was the Java 7 Update 11, which not only patched the CVE-2013-0422 vulnerability but also changed the default Java Security Level from ‘Medium’ to ‘High’. This change means that a security warning dialogue appears requiring the user to click ‘Run’ before an applet is able to execute. In response, attackers are now changing their methods and bypassing this feature by loading either serialized applets or Java Network Launch Protocol files (jnlp).
Another new trick we saw introduced in H1 2013 is that a number of exploit kits, such as Neutrino, Sakura and Redkit, have started encrypting their binary payloads using XOR or AES encryption to evade detection.

Exploit site hosting
Users typically encounter exploit kits during online browsing, either by stumbling onto a compromised site injected with an exploit kit, or by being hijacked or redirected to a malicious attack site. Most of the websites distributing exploit kits are hosted in the United States and Europe. 3 countries in particular—the US, Germany and the Netherlands—host a significant portion of all exploit sites.

Some exploit kits favor hosting their attack sites in specific countries. For Crimeboss, 84% of its exploit sites are hosted in just 3 countries—Germany, Luxembourg and the Netherlands. Meanwhile, the US hosted 54% of the sites known to distribute the SweetOrange exploit kit, as well as 40% of both Blackhole and Styx sites. Tracking the IP addresses associated with the kits also turned up a handful of ASNs involved in hosting the exploit sites. For example, 14% of Blackhole’s IP addresses, 12% of Styx’s and 7% of Cool’s were all tracked back to AS16276 Ovh Systems.

Protecting against exploit kits
The most commonly cited advice for protecting against exploit kits involves speeding patching of affected systems as soon as updates are issued by vendors. When zero-day exploits are involved however, the most effective defense would be to reduce the attack surface of any system connected to the Internet. This requires a multi-faceted, ‘defense in depth’ approach to security, and includes such elements as web browser security, web traffic scanning, email filtering, well-implemented intrusion detection/prevention systems and more. For suggestions on how to protect a system against exploit kits, see Recommendations on page 39.

VULNERABILITY EXPLOITATION

Following on the trend from 2012, in H1 2013 vulnerability exploitation continues to be a popular way for malware distributors and attackers to gain access to a machine, whether it’s to plant malware on the system or to perform some other nefarious action. In this article, we look at some of the interesting trends we noted in exploiting vulnerabilities during the first six months of 2013.

Focusing on zero-days
Throughout most of 2012, we observed attacks mainly being targeted against vulnerabilities that had been publicly known for some time, often years. For example, throughout 2012, one of the most commonly attacked vulnerability was CVE-2010-0288—a vulnerability in Adobe Reader and Acrobat that had been discovered fully two years earlier.

In 2013 however, we noticed a shift and an increasing focus on exploiting zero-day vulnerabilities, which had not yet had patches released by the affected application’s vendor. During this period, our telemetry data indicated that an astonishing 95% of detections reported to our systems identifying attacks against specific vulnerabilities were targeted against only five vulnerabilities; and of the five most targeted, three were vulnerabilities that had only been publicly announced within the last six months. Not coincidentally, all the top five vulnerabilities are known targets for various exploit kits (see Exploit Kits article on page 34).

CVE-2011-3402 — the most targeted vulnerability
Far and away the most commonly targeted vulnerability in H1 2013 was the CVE-2011-3402 Truetype font vulnerability in Windows. This vulnerability first came to prominence when it was used by the Duqu malware in a targeted attack campaign in early 2012; by late 2012, the exploit for the vulnerability had been added to various exploit kits. Since then, attacks against this single vulnerability have skyrocketed and in the past six months, CVE-2011-3402 alone accounted for an amazing 69% of all exploit-related detections reported.

Java — second most targeted program
Of the top five most targeted vulnerabilities, four are found in the Java development platform, either the Runtime Environment (JRE) or the browser plug-in. This is hardly surprising since next to the Windows operating system (also a popular target for exploits), Java is probably the second-most ubiquitous program in an organization’s IT setup. In H1 2013 however, the security issues related to Java received unwelcome attention from a series of widespread and
Unfortunately, removing either the runtime or plug-in may not be a feasible option for companies that use Java in business-critical instances. For organizations faced with such a dilemma, the alternative involves hardening their system’s or network’s attack surface against exploitation — a more effective, but also more complicated solution than just simply uninstalling a program. Possible mitigation strategies for these companies would likely involve some combination of tweaking Java’s security settings, configuring web browser settings to minimize unwanted applet execution (or installing other third-party plug-ins to do so) and monitoring network traffic.

Geographical distribution of vulnerability attacks
Perhaps unsurprisingly, the country that reported the most vulnerability-based attacks was the United States, where according to our telemetry data, approximately 78 out of every 1,000 users saw a detection identifying an exploit of a specific vulnerability within the last six months.

Out with the old
3 of the top 5 targeted vulnerabilities (marked in blue) were discovered within the last 6 months.

Top 5 Exploited Vulnerabilities
95% of all vulnerability-related attacks reported by our clients in the first half of 2013 involved 5 vulnerabilities, 4 of them in Java.
Germany also saw a relatively high amount of vulnerability attacks, with approximately 60 out of every 1,000 users reporting a hit in the same time span. Of the top 10 most targeted countries, all except the United States were in Europe.

### Market for exploits expanding

For years now, a largely overlooked aspect of security has involved the way vulnerability disclosures are handled, especially for zero-days in popular applications. Security researchers and application vendors have long debated whether the best way to handle such issues was to keep the information secret (with or without payment), or publicly disclose it to force developers to fix the security problems.

This has been of particular concern with the increasing activity in the rapidly growing ‘exploit sales’ market \[1\], which essentially monetizes vulnerability research by selling the results to the highest bidder. In addition to the application vendor concerned, other players (government contractors, other researchers and malware authors) become potential buyers of information that was once released for free, or for a token sum.

Though legal, exploit sales have typically been clandestine in nature. As vulnerability exploitation has become more widely used in both targeted attacks and malware distribution however, it seems to be becoming more prominent, if not exactly open. In early 2013, there was a report \[2\] of the Blackhole/Cool exploit kit author posting a notice in a Russian cybercrime forum, offering to purchase exploits. According to the reported translation of the notice, the ‘exclusive program’ offered by the author also included payment for “improvements to existing public exploits and also any good solutions for improving the rate of exploitation”.

Government agencies and contractors have also reportedly been involved in purchasing software exploits, purportedly for use in offensive cyberattacks \[3\], despite burgeoning concerns that this activity would ultimately damage the security and civil liberty of their own citizens \[4\]. Though no government has openly stated they are actively engaging in exploit purchasing, most reports implicate agencies in the United States and China of doing so.

### Countering exploits sales

In H1 2013, Microsoft launched its own ‘bug bounty’ program to encourage researchers to responsibly disclose vulnerabilities in its products. The company joins a small handful of technology entities (Facebook and the HP-affiliated Zero Day Initiative among them) trying to counter the lure of the ‘exploits-for-sale’ market by offering payment for vulnerability information. These initiatives faces stiff competition however; reports indicate that some government players are willing to pay six-figure sums for both the critical information and an agreement to keep the vulnerability private, essentially guaranteeing the buyer a secret access point to the program of interest.

Government interest in exploit sales has also extended to tentative talk of introducing legislation to regulate the market \[5\], at least in the European Union. Until that becomes a reality though, the market in exploit sales is likely to keep growing. What this means for the future state of cybersecurity is anyone’s guess, but at a minimum, organizations should certainly pay more attention to defending their networks and systems from vulnerability exploitation.

**SOURCES**

2. Krebs on Security; Brian Krebs; *Crimeware Author Funds Exploit Buying Spree*; published 7 Jan 2013; [https://krebsonsecurity.com/2013/01/crimeware-author-funds-exploit-buying-spree/](https://krebsonsecurity.com/2013/01/crimeware-author-funds-exploit-buying-spree/)
4. Reuters; Joseph Menn; *Special Report: U.S. cyberwar strategy stokes fear of blowback*; published 10 May 2013; [http://www.reuters.com/article/2013/05/10/us-usa-cyberweapons-specialreport-idUSBRE9490EL20130510](http://www.reuters.com/article/2013/05/10/us-usa-cyberweapons-specialreport-idUSBRE9490EL20130510)
5. Slate; Ryan Gallagher; *Cyberwar’s Gray Market: Should the secretive hacker zero-day exploit market be regulated?*; published 16 January 2013; [http://www.slate.com/articles/technology/future_tense/2013/01/zero_day_exploits_should_the_hacker_gray_market_be_regulated.html](http://www.slate.com/articles/technology/future_tense/2013/01/zero_day_exploits_should_the_hacker_gray_market_be_regulated.html)
RECOMMENDATIONS: Protecting against exploits

1. Apply security updates

Apply security patches for affected programs as soon as they are released by the application vendors. It would go a long way towards preventing a wide range of exploit kits from successfully compromising a machine.

In addition to a consistent patch cycle (particularly if patches are problematic to push out), companies can protect their systems against vulnerability exploitation with an Intrusion Detection and Prevention System (IDS/IPS) running up-to-date signatures.

<table>
<thead>
<tr>
<th>CVE</th>
<th>AFFECTS</th>
<th>PATCH RELEASED</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-2423</td>
<td>Java</td>
<td>16 April 2013</td>
</tr>
<tr>
<td>2013-1493</td>
<td>Java</td>
<td>4 March 2013</td>
</tr>
<tr>
<td>2013-1331</td>
<td>Microsoft Office</td>
<td>11 June 2013</td>
</tr>
<tr>
<td>2013-0809</td>
<td>Java</td>
<td>4 March 2013</td>
</tr>
<tr>
<td>2013-0422</td>
<td>Java</td>
<td>13 January 2013</td>
</tr>
<tr>
<td>2012-5076</td>
<td>Java</td>
<td>16 October 2012</td>
</tr>
<tr>
<td>2012-4681</td>
<td>Java</td>
<td>30 Aug 2012</td>
</tr>
<tr>
<td>2012-1723</td>
<td>Java</td>
<td>12 June 2012</td>
</tr>
<tr>
<td>2012-0507</td>
<td>Java</td>
<td>17 May 2012</td>
</tr>
<tr>
<td>2011-3544</td>
<td>Java</td>
<td>18 October 2011</td>
</tr>
<tr>
<td>2011-3402</td>
<td>Windows OS</td>
<td>8 May 2012</td>
</tr>
<tr>
<td>2010-0188</td>
<td>Adobe Reader &amp; Acrobat</td>
<td>23 February 2010</td>
</tr>
<tr>
<td>2010-0840</td>
<td>Java</td>
<td>30 March 2010</td>
</tr>
<tr>
<td>2010-1885</td>
<td>Windows OS</td>
<td>10 June 2010</td>
</tr>
</tbody>
</table>

2. Minimize attack surface

Security updates and intrusion detection are effective against known threats to already-patched vulnerabilities. But for zero-day exploits which do not yet have fixes, the most effective protection is to minimize the attack surface (the various vectors through which malicious code can be run) of any system connected to the Internet.

For most end users, ’reducing the attack surface’ translates simply enough to securing the web browser. This is particularly true in cases where uninstalling the Java web browser plug-in, currently a favored attack vector for multiple exploit kits, is not an option. In such cases, the user can instead increase the security of the web browser by:

(i) Updating Java

In the Java 7 update 11 release, the default security level setting for Java was increased to High. This configuration means that users need to expressly authorize an applet execute (whether they are unsigned or self-signed).

(ii) Disabling the Java browser plug-in

If updating Java isn’t an option, user can focus on managing the Java browser plug-in by disabling the plug-in and only enabling it when needed. This can be done via a handy, one-click option in the Control Panel (available in the Java 7 update 10 release) or via the web browser’s settings. The instructions for disabling Java in various web browsers are available at: http://www.java.com/en/download/help/disable_browser.xml

(iii) Using two browsers

Rather than fiddling with security settings, the user may opt for a two-browser strategy, in which one browser with the Java plug-in enabled is dedicated solely to using the website or program that demands it. All other web browsing is done on a separate browser without the plug-in.

(iv) Enabling Click to Play

For Java-enabled web browsers, an additional touch of security comes from the plug-in blocking feature built into most browsers. In Firefox and Opera, it’s known as ‘Click to Play’ while Chrome has a ‘Block all’ option for plug-ins in its Contents Settings page. This functionality prevents automatic execution of plug-ins (not just Java) and requires the user to click on the plug-in of interest before it will run.

(v) Using third-party apps

Another possibility is to use third-party programs to block plug-ins from automatically running on page load, unless the user chooses otherwise. The most popular of such programs is NoScript, which blocks multiple types of active content in Mozilla-based browsers, though there are a handful of other applications available that perform a similar function.

TOP 15 MOST TARGETED VULNERABILITIES IN H1 2013, LISTED BY CVE NUMBER

- Even with fully patched programs and a locked down web browser, companies concerned about vulnerability-based attacks against their systems may also need to consider and counter the possibility of an undisclosed vulnerability being used. In such cases, the last line of defense is a well-implemented IDS/IPS setup, or an equivalent system capable of identifying suspicious activity on the machines in the protected network.

- Though general recommendations are less useful here, ideally, such a protection system would be a unique combination of network monitoring, firewall, white- and blacklisting solutions that suits the normal needs and production environment of the end users.

- Zero-day exploits

- For Java-enabled web browsers, an additional touch of security comes from the plug-in blocking feature built into most browsers. In Firefox and Opera, it’s known as ‘Click to Play’ while Chrome has a ‘Block all’ option for plug-ins in its Contents Settings page. This functionality prevents automatic execution of plug-ins (not just Java) and requires the user to click on the plug-in of interest before it will run.

- Another possibility is to use third-party programs to block plug-ins from automatically running on page load, unless the user chooses otherwise. The most popular of such programs is NoScript, which blocks multiple types of active content in Mozilla-based browsers, though there are a handful of other applications available that perform a similar function.
KUMAR IN THE MAC (KitM)

During the first half of 2013, we saw the first Mac malware signed with a valid Apple Developer ID. For those unfamiliar with code signing, the Apple Developer ID is like a signature in digital form. Just like a handwritten signature, which identifies a document to its author, a digital signature identifies a code to its developer.

Last year, Apple introduced a security feature called Gatekeeper to OS X. Gatekeeper restricts running of applications to only those from identified developers as a default protection. This helps prevent users from unwittingly running malwares since malwares are generally not signed. Malware authors would not want their malicious work linked back to them, wouldn’t they?

Ironically, this is exactly what a malware author did. To pass through the defense introduced by Apple, the malware was signed using an Apple Developer ID in the name of Rajinder Kumar (see Figure 1), hence the name KitM or “Kumar in the Mac.” KitM is our designation for samples signed by the affected Apple Developer ID.

The malware was discovered by an independent researcher, Jacob Appelbaum. He found two spyware variants on the Mac machine of an Angolan activist during a workshop in the Oslo Freedom Forum.

The first variant was an application bundle having a file called “FileBackup”. After the discovery, we started mining our backend system for samples having the same filename, and found one that dated back to April 21, 2012 [2,3]. The sample not only share the same filename but also its internal structures such as classes, methods, etc., which led us to conclude that it was indeed an earlier variant. This sample also contained build paths that revealed to us that the malware is actually a repurposed version of an open source online backup utility called FileBackup (see Figure 3).

The utility uses a configuration file called “FileBackup.ini” to determine which files to backup (based on extensions) and where to upload the backup (based on a drop URL). Naturally, we tried to locate the corresponding configuration file. We found one that was shared to us by an AV partner at the same time as the sample mentioned earlier[3]. We found that the configuration file specified a domain (see Figure 4) similar to that in newer variants, which you will see later in Table 1. This confirmed that the sample is not just some benign utility but is indeed used maliciously. This also confirmed that the APT has been going on since much earlier than anyone has expected.

It is not clear whether Rajinder Kumar is the actual person behind the malware. A quick search of the name in LinkedIn returned at least 468 hits (see Figure 2). The name might as well be the “John Smith” equivalent of India. It might be just a disposable name cleverly chosen by the attackers to get an Apple Developer ID[3]. Regardless, Apple was swift in revoking the said ID to prevent any future infection of the malware.
The other variant discovered by Appelbaum is something we have never seen before. This variant takes screenshots instead of collecting files. Analysis of the sample revealed that it shared a lot of codes with the file-stealing variant, some of which were not even used. From this we deduced that the screen-grabbing variant is a development branch of the file-stealing variant.

Since then, other cases have been reported which lead to the discovery of other earlier variants\[5\]. Some of the file-stealing variants have a downloader capability. In most cases, only the file stealer was used. In one case however, the file stealer downloaded a companion application, which turned out to be identical to the screen grabber discovered by Appelbaum. We therefore believe that most likely this is also the case for the Angolan activist.

To summarize the attack (see Figure 5), the victim receives a spear phishing email containing the file stealer as an attachment. Take note that some of the file stealer variants have a downloader ability. The attachment, which is an application, poses as a document, image or other types of media. After the victim falls for the social engineering trick and get himself infected, the attackers may choose to introduce additional malware depending on the purpose of the attack. For now, only the screen grabber is known to be used but do not be surprised when a key logger or backdoor variant pops out in the future. It is not uncommon for APT attacks to have such components.

Security researchers have linked the cases to a larger espionage effort known as Operation Hangover\[6\]. The operation is known to have been employing freelance developers\[7\]. Interestingly, a path containing the name “Elance” (see Figure 6) was found on an unused remnant file (login.scpt—see Table 1) of some of the file-stealing variants. Elance here may be a reference to the online freelance recruitment site\[8\].

![Figure 5: KitM attack summary](image)

![Figure 6: Project for Elance](image)

<table>
<thead>
<tr>
<th>Table 1: Sample Development Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ATTACK PERIOD</strong></td>
</tr>
<tr>
<td>Signed by Rajinder Kumar (KitM)</td>
</tr>
<tr>
<td>Designated malware family</td>
</tr>
<tr>
<td>Filename / AppDelegate-prefix</td>
</tr>
<tr>
<td>Functionality</td>
</tr>
<tr>
<td>Uses “FileBackup.ini”</td>
</tr>
<tr>
<td>Uses “login.scpt”</td>
</tr>
<tr>
<td>Comes with “login.scpt”</td>
</tr>
<tr>
<td>Domain of secondary payload URL</td>
</tr>
<tr>
<td>Remnant domain found in sample (used as fallback)</td>
</tr>
</tbody>
</table>
We have summarized the major sample developments of the attacks in Table 1. You may notice that during the majority of the operation, the attackers used “liveapple[dot]eu” as the command and control (C&C), which lasted until early H1 2013 (February). However the remnant domain “researcherzone[dot]net” was found on the April 2012 sample. This may indicate that there are yet undiscovered earlier samples using that domain as C&C.

During later H1 2013 (April), the C&C changed to “securitytable[dot]org” for the drop URL and “torqspot[dot]org” for the secondary payload URL. We all know that the secondary payload hosted in “torqspot[dot]org” was the screen grabber discovered by Appelbaum. However no one has found a sample yet that is supposed to be the secondary payload hosted in “liveapple[dot]eu” when it was live. The authors have implemented the downloader capability long before the “securitytable[dot]org”/“torqspot[dot]org” variant, which almost guaranteed that such sample exists. This sample should be something that is not a file stealer, perhaps an earlier variant of the screen grabber or another type of malware like a backdoor or key logger that uses “liveapple[dot]eu” as its drop URL domain.

Finally the remnant domain “docsforum[dot]info” found in the screen grabber may indicate that there are undiscovered secondary payload samples using that domain as C&C.

Whichever is the case, users can be assured that their AV will most likely be able to detect if not protect infection from those possible in-between variants. This is because it is known that the attackers have always used a file-stealing malware that belongs to an already well-known family, at least by now, called Hackback as the initial payload. For those missing secondary payloads, we know that the attackers have been signing their work before they started using secondary payloads, which means these missing variants will also be signed. For F-Secure users, these variants will be identified as Kumar in the Mac (KitM).

**SOURCES**

1. Twitter; Sean Sullivan; published 16 May 2013; https://twitter.com/SeanSullivan/status/33498732556054528
2. Twitter; Sean Sullivan; published 16 May 2013; https://twitter.com/SeanSullivan/status/33505549859980290
3. Sample; sha1 hash: 93decbdc89e52d5c5bde5d443f1f00e2b49d7
4. Sample; sha1 hash: 55374f3a281bd81e70ae1be8857085906cc87204
6. F-Secure Weblog; Sean Sullivan; Big Hangover; published 21 May 2013; http://www.f-secure.com/weblog/archives/00002557.html
8. Elance; http://www.elance.com/
TOTAL COUNT (JAN-JUN 2013) = 33 families + variants of Mac malware

NEW FAMILIES AND VARIANTS
- BACKDOOR: 57.6% (19/33)
- TROJAN: 36.4% (12/33)
- OTHERS: 6% (2/33)
- ROOTKIT: 1/2
- VIRUS: 1/2

** The number shown is the count of unique variants detected. This means repackaged installers are not counted and multiple component malware only have 1 count.
When the word phishing comes to mind, we usually think of our bank accounts and the credentials needed to access them. But in the age of one-time passwords and multiple-factor authentications, a lowly phishing webpage rarely stands a chance against such sophisticated protection. As such, more advanced banking trojans have emerged to take on the challenge of breaking into online-accessible bank accounts.

This does not mean that phishing has been rendered obsolete however; it only means that phishing has to be less discriminating regarding its catch. Nowadays, every person who spends time online has something that phishers can steal and attempt to sell. In an age where all online information is fair game, the melody has become: if it has a username and password, then it can be a phishing target—if it’s not already.

This means everything from Google, Yahoo, AOL or Live accounts used for e-mails and online storage, to social networking and media accounts such as Facebook and Twitter, used to keep abreast with the latest updates from various connections. Even dating websites are not immune. Match.com and SeniorPeopleMeet are only a few of the names targeted in the dating scene. And how about gaming sites and gaming portals? Battle.net, Steam, Runescape, Habbo, anyone?

Sourcing and reaching targets

The most visible place where users run into phishing URLs are spam e-mails. The recipients for these unsolicited mails can be bought in bulk from underground markets or even various websites that offer e-mail addresses for sale (see Figure 1). After the e-mail addresses are obtained, the spamming begins. Despite the low click-through rate associated with spam, the sheer volume of spam sent out worldwide results in enough traffic going to the phishing sites to make this method a mainstay of the phisher’s business model.

Once the users click the link in the spam e-mail, they are taken to the phishing site, which is usually crafted with a particular target in mind.

Types of phishing sites

We profiled 71% of all the phishing URLs we gathered during the first half of this year and found that they are of two types: direct phishes and indirect phishes.
Direct phishing straightforwardly involves displaying a fake webpage mimicking the target site, which is used to steal login credentials and possibly other information for that site (see Figure 2, top left). This type of phishing has been around since phishing has existed and has not really changed much, except perhaps for the improving ‘look and feel’ of the phishing sites. When it comes to the overall coverage though, direct phishes still trump indirect phishes 99 to 1.

Indirect phishes are pretty recent and involve displaying a fake webpage mimicking a secondary ‘bait’ site, which is used to steal the credentials for another site, which is the real target. For instance, the ‘bait’ site may be a tax filing site linked to various banks, and the phishing site copying the bait may include a new ‘feature’ that requires the users of the associated banks to enter their username and password for “account verification” (see Figure 2, bottom). Indirect phishes are mostly, though not exclusively, targeting sites in the real estate, taxation and logistics categories.

Phishing Sites Distribution
When it comes to sorting out what is the most valuable information that phishers are aiming for, PayPal credentials top the charts—the online payment service is the target of 73% of all phishing sites we profiled, with all other organizations sharing the remaining 27%. It’s pretty obvious why that information is easy to sell, it’s essentially free money. Also, since usage of the service is so widespread, phishers can get victims from everywhere.

Phishing, being less discriminating nowadays, necessarily covers a wide range of categories (the type of targets they attack), though the top 10 most targeted categories are, unsurprisingly, usually in some way related to online commerce. The phishing sites are also expanding by targeting brands in various regions and countries (quite aside from the fact that many of these brands also have an international presence). The geographical distribution of the phishing sites and their categories are listed on the next page.

Automated Phishing and Scattershot Phishing
Since phishing kits (toolkits that automate the production of phishing sites) have become quite common, setting up phishing pages for anything that will take the bait has become easier, and everyone is now a potential victim. These kits make it easier for the miscreants to generate different pages for potentially different targets. Although this article does not go into detail dissecting individual phishing kits, one point worth noting is the similarity in the format many phishing links, which points to the pages being automatically generated, which in turn indicates the phishing is being facilitated by the use of a phishing kit.

Another notable trend in the phishing sites we’ve seen is scattershot phishing, which involves multiple financial institutions, as well as other targets, getting phished from a single domain. This technique allows the phisher to target potential victims at a fraction of the cost, as multiple domains don’t need to be created to address multiple targets. Above is an example of scattershot phishing.

Selling the Harvest
Once the phishers have gained their victim’s information, they can sell their ‘wares’ in underground markets or websites. The details can be sold on a per-item basis or in batches, depending on the site and the sellers’ preferences. Below are examples of two sites that cater to this underground market in stolen credentials.
### Categories & Brands Targeted for Indirect Phishes, by Percentage

- **Real estate agencies**: 80%
- **Tax agencies**: 19%
- **Logistics companies**: 1%

### Top 10 Categories for Phished Sites, by Percentage

- Payment services
- Banks
- Gaming
- Credit cards
- Online shopping
- Online portals
- Real estate
- Social media
- Utilities
- All other types

### Number of Targeted Brands per Category, Top 10

- **Banks**: 82
- **ISPs**: 11
- **Online shopping**: 13
- **Credit cards**: 7
- **Real estate**: 6
- **Online portals**: 6
- **Dating sites**: 5
- **Gaming**: 4
- **Social media**: 4
- **Tax agencies**: 4

### Targeted Brands per Region, by Percentage

- **Europe**: 28%
- **Global**: 22%
- **North America**: 23%
- **South America**: 13%
- **Oceania**: 6%
- **Africa**: 4%
- **Asia**: 4%
H1 2013 INCIDENTS CALENDAR (PAGE 6)

MALWARE DEVELOPMENT


[6] The Register; Thomson, Iain; Mac malware found with valid developer ID at freedom conference; published 17 May 2013; http://www.theregister.co.uk/2013/05/17/mac_malware_steals_screenshots/

HACKS & ESPIONAGE
[1] Ars Technica; Goodin, Dan; Red October espionage platform unplugged hours after its discovery; published 19 January 2013; http://arstechnica.com/security/2013/01/red-october-espionage-platform-unplugged-hours-after-its-discovery/


[5] Guardian UK; Branigan, Tania; South Korea on alert for cyber-attacks after major network goes down; published 20 March 2013; http://www.guardian.co.uk/world/2013/mar/20/south-korea-under-cyber-attack


[12] Trend Micro; Aquino, Marhalito; RARSTONE found in Targeted Attacks; published 13 June 2013; http://www.guardian.co.uk/technology/2013/jun/07/uk-gathering-secret-intelligence-nsa-prism
MOBILE MALWARE DEVELOPMENT


[4] The Register; McAllister, Neil; Sneaky new Android Trojan is WORST yet discovered; published 7 June 2013; http://www.theregister.co.uk/2013/06/07/android_obad_trojan/


SECURITY & ENFORCEMENT


[3] Reuters; Accused LulzSec hacker gets year in prison over Sony breach; published 18 April 2013; http://uk.reuters.com/article/2013/04/18/usa-lulzsec-hacker-idUKL2N0D521220130418

[4] The Register; Leyden, John; British LulzSec hackers hear jail doors slam shut for years; published 16 May 2013; http://www.theregister.co.uk/2013/05/16/lulzsec_sentencing/


[6] Ars Technica; Gallagher, Sean; Twitter launches two-factor authentication, too late to save The Onion; published 23 May 2013; http://arstechnica.com/security/2013/05/twitter-launches-two-factor-authentication-too-late-to-save-the-onion/

[7] BBC; FBI and Microsoft take down $500m-theft botnet Citadel; published 6 June 2013; http://www.bbc.co.uk/news/technology-22795074


F-Secure has been protecting the digital lives of consumers and businesses for over 20 years. Our Internet security and content cloud services are available through over 200 operators in more than 40 countries around the world and are trusted in millions of homes and businesses.

In 2012, the company’s revenues were EUR 157 million and it has over 900 employees in more than 20 offices worldwide. F-Secure Corporation is listed on the NASDAQ OMX Helsinki Ltd. since 1999.
Protecting the Irreplaceable