

The Quarterly Magazine for Digital Forensics Practitioners

WITH!  
AN IPOD NANO IN THIS  
ISSUE'S COMPETITION

# DIGITAL VIRUS FORENSICS

ISSUE 16  
AUGUST 2013

MAGAZINE

## INSIDE

- / USING GOOGLE EARTH
- / UTILISING REP DATA
- / SOCIAL NETWORK STEGANOGRAPHY
- / THE HISTORY OF MALWARE



# VM INTROSPECTION

Unearthing and profiling sophisticated x64 bit kernel mode "bootkits" that continue to leverage holes on Windows 7



Issue 16 / £14.99 TR Media

/ **REGULARS**  
NEWS, 360, IRO,  
LEGAL & MORE...

/ **INTRODUCING**  
A FRESH LOOK AT  
CRYPTOGRAPHY

/ **FROM THE LAB**  
CREATING NEW FRONTIERS  
FOR LIVE FORENSICS



/ **BOOK REVIEWS**  
CUDA PROGRAMMING &  
SILENCE ON THE WIRE

# HISTORY OF MALWARE

*In the past three decades almost everything has changed in the field of malware and malware analysis. From malware created as a proof of some security concept and malware created for financial gain to malware created to sabotage infrastructure. Nikola Milosevic looks back on the history and evolution of malware and describes the most important malware from the mid '80s to today.*

## INTERMEDIATE

Malware, short for malicious (or malevolent) software, is software used or created by attackers to disrupt computer operations, gather sensitive information, or gain access to private computer systems. Malware may appear in the form of code, scripts, active content, and other software. 'Malware' is a general term used to refer to a variety of forms of hostile or intrusive software. Malware includes computer viruses, ransomware, worms, trojan horses, rootkits, keyloggers, dialers, spyware, adware, malicious Browser Helper Objects (BHOs) and other malicious programs; the majority of active malware threats are usually worms or trojans rather than viruses [1].

The history of malware can be split to five categories that will also represent a timeframe in which events from that category happened. These categories are:

- The first category is the early phase of malware development. This is time when the first malware started to appear.
- The second category is the early Windows phase. In this category we will describe the first Windows malware, along with the first mail worms and macro worms.
- The third category is the evolution of network worms. These threats became popular when the Internet became more widespread.
- The fourth category is rootkits and ransomware. These were the most dangerous evolution of malware before 2010.
- To bring us up to date we look at malware that was made for virtual espionage and sabotage. This malware was created by intelligence organisations of some countries. This is the latest phase of malware evolution that we are now facing.

In this article we will not describe all kinds of malware (there are just too many), we will, however, look at malware that could be considered a game changer, and introduce new aspects in the world of malware.

## BEGINNINGS OF MALWARE

There was malware for other platforms before 1986 but in 1986 the first malware for the Personal Computer (PC) appeared. It was a virus called Brain.A. Two brothers; Basit and Amjad developed Brain.A in Pakistan. They wanted to prove that the PC was not a secure platform, so they created a virus that was capable of replicating using floppy disks. It infected the boot sector of the floppy drive and the boot sector of every inserted floppy disk. Anytime an infected floppy was inserted into the PC, it would infect its drive, so the drive would then infect every disk inserted. This virus did no harm, and the authors signed the code, with their phone numbers and address [2]. The intention of early malware writers was to point out problems, rather than do some harm or damage. It was later that malware became more destructive.

After Brain.A, other viruses followed. One of the more interesting was the Omega virus. It was called Omega because

Fizzer (2003) was the first malware whose only purpose was to generate revenue and money. It came in an infected attachment, and turned the infected machine into a spam sender. Before Fizzer, enthusiasts who would like to prove something or to demonstrate a technique often wrote malware. From Fizzer onwards the main focus for malware writers was profit.

of the omega sign that it wrote in certain conditions in the console. It was infecting the boot sector, but was not doing much damage unless it was Friday 13th. On that day the PC would not boot.

The Michelangelo virus would, on Michelangelo's birthday in 1992, rewrite the first 100 sectors of a hard disk[3]. By doing this, the File Allocation Table (FAT) was destroyed and the PC could not boot. V-sign is another virus that also infected the boot sector and wrote a V sign on the screen every month. The Walker virus is the next virus that was quite visual and it appeared in 1992. It produced an animated walker, walking from one side of the screen to the other. The Ambulance virus was quite similar to Walker, however, in this case an animated ambulance would drive from one side of screen to the other, complete with the sound effects of an ambulance.

One of the most interesting viruses to appear at the beginning of 1990s was the Casino virus. Casino would copy the file allocation table to memory and then delete the original table. The virus then offered a slot game to user. The user had to get 3 '£' signs if he wanted to use his PC and he had three tries. If the user restarted the machine the FAT would be gone,

and the machine would not be able to boot. The same would happen if the user lost the gamble, the file allocation table would be deleted from the memory. If the user won the game, the virus would copy back the FAT from memory, and the PC could be used normally.

The next big step in malware evolution was the introduction of Mutation Engine (MtE). A Bulgarian hacker who called himself Dark Avenger created Mutation Engine. It was a tool that could add mutation functionality to viruses, so they would be harder to detect by anti-virus software. This was essentially the first polymorphism module that could take any virus and make it far more invisible. Until mutation engine, anti-virus software was able to find viruses on PCs using file signatures and changes in file signatures. The introduction of polymorphism made this method ineffective [5].

Virus creation laboratory was the first User Interface (UI) tool for creating viruses. The user could select the features of the virus and create it. This made virus creation easy. It has some disadvantages, but almost anyone using this tool could create a virus [6].

## ✓ TYPES OF MALWARE

Malware can be split into the following categories:

- Virus – a self-replicating piece of code that attaches itself to other programs and usually requires human interaction to propagate.
- Worm – a self-replicating piece of code that spreads via networks and usually doesn't require human interaction to propagate.
- Malicious mobile code – a lightweight program that is downloaded from a remote system and executed locally with minimal or no user intervention.
- Backdoors – a program that allows attackers to bypass normal security controls on a system, gaining access on the attackers own terms.
- Trojan horses – a program that appears to have some useful or benign purpose, but really masks some hidden malicious functionality.
- Rootkits – Trojan horse backdoor tools that modify existing operating system software so that an attacker can keep access to and hide on a machine.
- Combination malware – Combination of two or more from above.

## ✓ THE FIRST WINDOWS MALWARE

When Windows was released, it was interesting for many users since it provided a powerful user interface. The simplicity of use attracted many users and anything that has many users in the computing world soon becomes interesting for both attackers and malware creators.

WinVir was the first Microsoft Windows virus. It was not doing much harm and its main feature was that it was self replicating, and that it was the first virus that had the ability to infect Windows Portable Executable (PE) files. WinVir made little changes to infected files. When the infected file was executed, WinVir looked for other PE files to infect. While WinVir was infecting other files the original executed file was "rolled back" to its original state; to put it simply, WinVir was deleting itself.

Monkey was a virus that infected the Master Boot Record (MBR) of hard drives and floppies. Monkey was moving the first block of the MBR to the third block and inserting its own code into the first block. When the infected computer was booted, it ran normally, unless it was booted from a floppy. In this case, an "Invalid drive specification" message appeared.



One-half or Slovak bomber was interesting and may be considered a quite destructive virus. It infected MBR, EXE and COM files, but did not infect files that contained words like SCAN, CLEAN, FINDVIRU, GUARD, NOD, VSAFE, MSAV or CHKDSK in the filename. These files were not infected because they might belong to some antivirus software, so auto-checking algorithms might catch the virus. It was encrypting parts of a user's hard drive using XOR function with some key known to the virus. But if a user tried to access an encrypted file, the file was decrypted and the user wouldn't notice anything. The problem with this virus was that if it was cleared inappropriately, the encrypted files couldn't be retrieved [7]. The virus displayed a message on the 4th, 8th, 10th, 14th, 18th, 20th, 24th, 28th and 30th of every month under particular circumstances

Concept (WM.Concept) was the first macro virus and it was detected in 1995. It was written in Microsoft Word macro language, and sharing documents spread it. It worked on PCs and Macintosh computers if Microsoft Word was installed. When documents infected with Concept were opened on some PCs, the virus would copy its malicious template over the master template, so every new document created on that computer would be infected [8].

Laroux (X97M/Laroux) was the first Microsoft Excel macro virus. It was written in Visual Basic for Application (VBA), macro language for Office documents which were based on Visual Basic. It worked on Excel 5.x and Excel 7.x. It could also be run on Windows 3.x, Windows 95 and Windows NT. It did not do any harm, it just replicated.

Boza was the first virus that was written specifically for Windows 95. It infected Portable EXE files; files that were using Windows 95 and Windows NT, but it did not attack Windows NT itself. There had not been a virus detected that was written particularly for Windows NT until this virus was detected in January 1996. It had Australian origins, but it was detected all over the world. When a file infected with Boza was executed, it would infect other files in that directory. One to three files would be infected on each run after which Boza would run the original program. The virus would not be active in the memory anymore.

Boza spread quite slowly, but the spreading algorithm was fast enough that the user could not detect it. Boza had no destructive routines, but it did have one error that under

some circumstances caused infected files to become several megabytes in size. This was a problem on machines where hard disks were less than 100Mb capacity. The virus had an activation routine that displayed a message window on every 31st of any month. The Messages were: "The taste of fame just got tastier!" and "From the old school to the new".

Marburg (Win95/Marburg) is a virus that started to circulate in August 1998 when it infected the master CD of the MGM/EA PC game called Wargames. The Publisher, MGM on 12th of August 1998 published an apology to users:

*From: "K.Egan (MGM)" <kegan@mgm.com>  
Subject: MGM WarGames Statement  
Date: Wed, 12 Aug 1998 18:03:39 -0700*

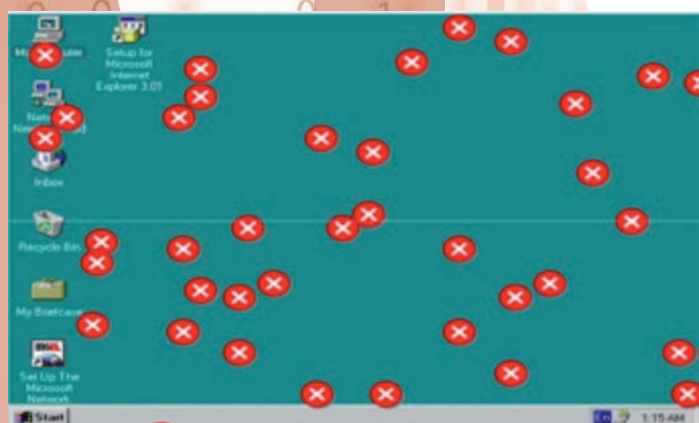
*MGM Interactive recently learned that its WarGames PC game was shipped with the Win32/Marburg.a virus contained in the electronic registration program. The company is working as fast as it can to resolve the problem... MGM Interactive is committed to delivering top quality products to consumers. This is an unfortunate circumstance and we sincerely apologize for any convenience this has caused you. ... If you have any questions or if you would like to receive a replacement disc, please contact MGM Interactive.*

This same virus was found on the CD that covered the Austrian PC Magazine Power Play, in August 1998.

Maburg is a polymorphic virus that infected Win32 and SCR (screen saver) files and encrypted its code with a polymorphic variable layer of encryption. The polymorphic engine of the virus was quite advanced since it was an encrypting virus with 8, 16 and 32 bit keys using several different methods. The virus used a slow polymorphism, which means that it changed its decrypt slowly. Maburg deleted the integrity database of several antivirus programmes. It also avoided infecting files that belonged to antivirus software or files that contained a V in the filename. This was done to prevent auto checking of antivirus software. Maburg activated 3 months after the infection if the infected file was run at the same hour as the hour of infection it displayed a standard MS Windows error icon (a white cross in a red circle) all over the desktop [9].



Boza



Maburg

Happy99 was the first mail virus. It spread as an executable attachment to an e-mail and was detected in 1998. At that time spam filters barely existed, and the sending of executable files was still acceptable. If a user clicked and ran the attachment, it would show them a screen with fireworks; the virus would replicate the attachment and send mail to the user's contacts.

Melissa was a virus that combined the techniques of a macro virus and an email virus. It was sent via email with an attached infected MS Word file. When the infected file was opened it would replicate randomly chosen documents from the user's hard disk and subsequently send the infected files to all of the user's contacts. This was very problematic due to the potential for information leakage. In addition, certain variants added quotes from The Simpsons to the infected documents [3].

LoveLetter was one of the most successful social engineering viruses. It was using the premise of love to influence the user to open the attachment. The attached file would subsequently execute the virus and would rewrite some important files on the victim's system. Using the premise of love, the virus convinced millions of people to open the attachment causing financial damage estimated to be in the region of \$5.5Bn dollars worldwide. Anakurnikova was a similar virus that sent an executable file, socially engineering victims to believe that there were sexy photos of the tennis player Ana Kurnikova attached. Many were convinced to open the attachment, even after the antivirus companies had resolved the detection and blocked the malicious attachment. In fact, many victims asked the AV company support teams on how they might view the pictures.

## WORMS

At the end of the 1980s the first PC worm was accidentally created. In 1988 Robert Tappan Morris, who was at the time a student at MIT wrote a program that would become a significant event in malware history. As part of his project, Morris wanted to count the number of computers connected to Internet. So he wrote a little program that would replicate from one connected computer to another and count incrementally. But Morris unfortunately created a bug in his code and the worm consequently revisited computers that it had previously visited. In actuality, the worm was replicating from an infected computer to all other connected computers continuously generating a significant amount of network traffic and almost crashing the

Internet of that time. Morris was arrested and convicted under the Computer Fraud and Abuse Act from 1986[10]. This was the first case where someone was convicted using this statute. At the time, computers had open ports and connections and replications could be carried out without the use of exploits.

Internet worms work using scanning algorithms that scan networks for computers and in most cases try accessing public or both public and private IP addresses. If the IP address is unassigned, assigned to a device that could not be attacked (wrong platform) or meet a patched and protected computer; the worm would not attack. However, if a computer with an IP address was running on an unpatched platform, the worm would use the exploit to gain access to that computer. Once access had been achieved, it would add a payload that could trigger, at some point in time, and once deployed, the worm would again start scanning the network in an attempt to propagate from that computer.

Code Red was the first Internet worm that did not need any user interaction. It was also the first intentionally written worm (the Morris worm was malicious by accident not design). Code Red was released in 2000 and spread all over the world in a couple of hours. The worm was successful in hiding from defending mechanisms and had several capabilities that were triggered in cycles. The worm attacked the Internet Information Service (IIS) web servers and for the first 19 days it only spread over the network using vulnerability in the IIS, from day 20 to day 27 the worm launched Denial of Service (DoS) attacks on selected websites (i.e. The Whitehouse) and for the last 3-4 days of the month it would just rest.

Nimda was discovered on September 18th 2001 and spread quickly all over the world. If the "Nimda" letters are reversed we get adm1N. Nimda used a code similar to Code Red by scanning networks and self-propagating, but it included additional features. The scanning algorithm of Nimda scanned all IP addresses while Code Red was scanning just the public IP range. Because of this feature Nimda was able to reach further and infect private networks[3]. Nimda also had the ability to change the hosted website in such a way that they would offer the downloading of infected files. Using this technique the spreading of Nimda was faster and more dangerous, because with user interaction, Nimda could overcome firewalls and be spread from private computer

## EVOLUTION OF ANTI-MALWARE TOOLS

As malware evolved, anti-malware software evolved as well. In the '90s, the first anti-virus applications used a signature-based detection, holding a database of all known viruses and malicious software, and comparing database signatures with files on a system. Later anti-malware application started using integrity databases for detected suspicious changes. When worms became a serious threat, the first firewall applications were built. Now anti-malware combines techniques from signature-based search, integrity checks, firewalls, machine learning, artificial intelligence and detection intrusion systems to detect suspicious activity on the defended machine. In addition, all new anti-malware tools send all suspicious applications to a lab for automated or expert analysis in a sandbox environment.

hosts. It could spread to Windows 95, 98, Me, NT 4 and Windows 2000. Nimda had one error in its code that resulted in the code crashing under certain circumstances.

Fizzer is a mail worm found in 2003. This was not an Internet worm per sé, however, we have included it here because of the timeframe when it was found. Fizzer was the first malware designed to generate a revenue stream. It spread as an infected attachment and turned the infected machine into a spam sender.

In this period we can see the changes to the structure of malware writers. Before Fizzer, enthusiasts wrote malware that would like to prove something or to demonstrate a technique. From Fizzer onwards, the main focus for malware writers was profit! Subsequent to Fizzer a lot of malware appeared that sent spam or was designed to blackmail computer users. In addition, the malware writers were not predominantly from developed countries as was seen in the 1980s and 1990s. The main sources of malware in 2000s were being developed by people from less developed countries technology wise, including Russia, China, Pakistan, India etc.

Slammer appeared on the 13th September 2003 and demonstrated some new attributes. It was an Internet worm that used vulnerability in OpenSSL and was one of the first malware that attacked Linux machines and Apache servers. It also had a backdoor, allowing the attacker to access the infected machine at will and upload to it some additional tools or malware. The backdoor was created using a UDP socket with the attacker listening on the UDP port 2002 for the attacker's connection.

In 2003 and 2004, the three most destructive Internet worms were discovered and were a catalyst for the review of security of real time systems (e.g. factories, power plants, airports and other transportation systems) and the idea of virtual sabotage.

Slammer was an Internet worm that was spread in 2003 using vulnerability in Microsoft SQL Server and Microsoft Data Engine 2000. Every application that used some of these two services was a potential target and entrance point for Slammer. Some of the applications that Slammer used to gain access to the system were:

- Microsoft Biztalk Server
- Microsoft Office XP Developer Edition
- Microsoft Project
- Microsoft SharePoint Portal Server

- Microsoft Visio 2000
- Microsoft Visual FoxPro
- Microsoft Visual Studio.NET
- Microsoft .NET Framework SDK
- Compaq Insight Manager
- Crystal Reports Enterprise
- Dell OpenManage
- HP Openview Internet Services Monitor
- McAfee Centralized Virus Admin
- McAfee Epolicy Orchestrator
- Trend Micro Damage Cleanup Server
- Websense Reporter
- Veritas Backup Exec
- WebBoard Conferencing Server[11]

Slammer was spreading as a memory process, it never wrote anything onto the hard disk so when the PC was restarted the infection would disappear. But since the PC was connected to other PCs from where it originally got the infection, or where it replicated the infection too, it was not long before the infection would be back. Slammer was creating great network traffic causing significant damage; e.g. the ATM network of the Bank of America was brought down, the 911 service in Seattle was down for couple of days, flight control systems at a couple of airports were infected and some flights were delayed, and, more worryingly, a problem in a nuclear power plant in Ohio was attributed to Slammer.

Blaster was detected in August 2003 and used buffer overflow vulnerability in Distributed Component Object Model Remote Procedure Call (DCOM RPC). Blaster was used to create a SYN flood to the windowsupdate.com website, but since it was the wrong website (the real one was windowsupdate.microsoft.com), it did not cause much damage to Microsoft which was the target. It did, however, create significant traffic and subsequently slow down and disable several systems like Air Canada, forcing planes to land, and the US train company CSX stopping movement of its trains.

Sasser in 2004 used a buffer overflow in Local Security Authority Subsystem Service (LSAS). It spread over the network and was often crashing the LSAS service resulting in a restart after one minute. When Microsoft released a patch, it was quite large to download and install, longer than the time the

## ✓ ZERO-DAY EXPLOITS

A zero-day (or zero-hour or day zero) attack or exploit is an attack that exploits a previously unknown vulnerability in a computer application, meaning that the attack occurs on "day zero" of awareness of the vulnerability. Malware, that use zero-day exploits for gaining control over some host, are the most dangerous because by their nature, no known patch or detection exists.

malware needed to crash the LSAS service. This caused a lot of frustration for users, resulting in a new model of automatic updates being developed. The Sasser worm caused Railcop trains to stop in Australia, Delta Airlines to have a problem and delays on British Airways flights, the Hong Kong government department of energy was infected, two hospitals in Sweden were infected and could not run scanners, the EU commission was infected, Heathrow airport had problems as well as the UK Coastguard and several Banks who ended up closing their offices for couple of days because of the infection.

### ROOTKITS & RANSOMWARE

RootKits are malware tools that modify existing operating system software so that an attacker can keep access to and hide on a machine. RootKits operate at two different levels, depending on which software they replace or alter on the target system. They may alter existing binary executables or libraries on the system; in other words, altering the very programs that users and administrators run (e.g. ls, cd, ps or other programs). We'll call such tools "User-Mode RootKits" because they manipulate the user-level operating system elements. Alternatively, a RootKit could go for the jugular, the centrepiece of the operating system, the kernel itself. We'll call this type of RootKit a "Kernel-Mode RootKit" [3].

The first RootKit seen was made by SONY Entertainment, and had quite a bad impact on SONY's reputation.

SONY BMG RootKit was created in 2005 in an attempt by SONY to protect the copyright of their publications. The idea was to detect and disable copying of their publications to other media using the RootKit. Sony BMG RootKit was included in 52 publications of Sony amongst them albums by Ricky Martin and Kylie Minogue. When a CD was inserted in a normal CD player nothing would happen, however, when a CD was inserted into a PC the RootKit would be installed, hide itself and all files starting with \$sys\$. It would control how the user accessed the music. If the user tried to copy the CD, the RootKit would prevent it. The functionality used to hide all files starting with \$sys\$ was subsequently used by other malware writers to hide their files on a system by naming malware files starting \$sys\$. When the RootKit was detected, it caused a storm when Thomas Hesse the Director of Global Sales in Sony BMG made a statement in which he said, "Most people, I think, don't even know what a rootkit is,

so why should they care about it?" This caused significant public reaction and resulted in an impact to SONY's image. This was a good demonstration on how to mismanage public relations and a lawsuit resulted in SONY offering customers a refund and free music downloads from their website.

StormWorm was an email worm that came 7 years after LoveLetter and like LoveLetter used social engineering to spread, except in this case it used fear and horror instead of love. StormWorm started spreading using an email with the subject "230 dead as storm batters Europe". Other manifestations have been seen using this same technique, namely:

- A killer at 11, he's free at 21 and kills again!
- U.S. Secretary of State Condoleezza Rice has kicked German Chancellor Angela Merkel
- British Muslims Genocide
- Naked teens attack home director.
- 230 dead as storm batters Europe.
- Re: Your text
- Radical Muslim drinking enemies' blood.
- Chinese/Russian missile shot down Russian/Chinese satellite/aircraft
- Saddam Husain safe and sound!
- Saddam Hussein alive!
- Venezuelan leader: "Let's the War beginning".
- Fidel Castro dead.
- If I Knew
- FBI vs. Facebook

Infected machines were creating what could be considered a botnet. However, since most botnets are controlled by one central Command & Control (C&C) server this was not the case with StormWorm, which was acting more like a peer-to-peer network, so the controlling node could change from host to host. StormWorm was also installing a RootKit that it used to hide itself. Later variants, starting around July 2007, loaded the rootkit component by patching existing Windows drivers such as tcpip.sys and cdrom.sys with a stub of code that loads the rootkit driver module without requiring it to have an entry in the Windows driver list.

Mebroot from 2008 brought one new thing that changed the game; just surfing the Internet using a browser could infect

the victim. It used an exploit in the browser to gain access to the system, one of the first websites used to spread this malware was the official website of Monica Belluci. When Mebroot gained access to the victim's PC, it would install a rootkit that could hide from RootKit detectors, which became part of many antivirus solutions. Mebroot spied on what the victim was typing and sent this data to the attacker. The malware had been significantly debugged, so it almost never caused a crash of the system. Even if it caused a crash, it could still collect and send traces back to the attacker allowing him to further debug and fix the problem.

Conficker is one of the greatest mysteries in malware history. The intention of the malware creator is not known. It uses vulnerability in Windows and the cracking of weak passwords to spread; it installs a backdoor, a rootkit and creates a botnet node on the infected machine. It infected about 10 million host PCs but the great mystery is that whilst it had a very complex botnet it was never used for any attack.

Interestingly, ransomware is malware that encrypts victims' hard disks, changes the desktop background with a message and demands \$120 for the decryption key. The interesting aspect is that the attackers were giving away the keys if they were paid. To spread, it uses browser vulnerability and an infected PDF file with a script that downloads and installs the malware. It would change the desktop background and placed a "how-to-decrypt.txt" file on the desktop, in which the following text appeared:

*Attention!!!*

*All your personal files (photo, documents, texts, databases, certificates, kwm-files, video) have been encrypted by a very strong cypher RSA-1024. The original files are deleted. You can check this by yourself – just look for files in all folders.*

*There is no possibility to decrypt these files without a special decrypt program! Nobody can help you – even don't try to find another method or tell anybody. Also after n days all encrypted files will be completely deleted and you will have no chance to get it back.*

*We can help to solve this task for \$120 via wire transfer (bank transfer SWIFT/IBAN). And remember: any harmful or bad words to our side will be a reason for ignoring your message and nothing will be done.*

*For details you have to send your request on this e-mail (attach to message a full serial key shown below in this 'how to..' file on desktop): [email address]*

Files that were encrypted on the hard disk had extensions: .jpg, .jpeg, .psd, .cdr, .dwg, .max, .mov, .m2v, .3gp, .doc, .docx, .xls, .xlsx, .ppt, .pptx, .rar, .zip, .mdb, .mp3, .cer, .p12, .pfx, .kwm, .pwm, .txt, .pdf, .avi, .flv, .lnk, .bmp, .icd, .md, .mdf, .dbf, .mdb, .odt, .vob, .ifo, .mpeg, .mpg, .doc, .docx, .xls, and .xlsx.

## /// VIRTUAL SABOTAGE & ESPIONAGE

In 2010 a huge step in malware evolution occurred. Malware is no more seen just as a threat for businesses, personal finances or files. Military, police forces and intelligence agencies of several countries started to get involved in malware creation. Malware is now considered just like any other weapon. The US government declared that the US Army maintains the right to respond to a cyber attack with a physical attack. Dropping bombs and cyber attacks using malware are considered to be on equal footing, in addition, malware has become capable of doing almost the same damage as a bomb, but without risking human lives. The best example for this type of malware is Stuxnet, which was discovered in summer 2010.

Stuxnet is the first so-called super malware. When found it was soon realized that it had been spreading undetected for about a year. When Stuxnet was eventually detected, it had already done what it was built for. It is believed that Stuxnet was created to destroy or at least slow down the Iranian nuclear program. Stuxnet physically sabotaged turbines for uranium enrichment by changing rotation frequencies. This was done in a way that had not been seen before. Stuxnet was spread using a USB stick, when inserted any disabled "autorun" or "autoplay" feature would not help. If a USB stick was inserted into an infected PC it would be subsequently become infected and if an infected USB stick was inserted into a PC, then the PC would become infected and no anti-virus was able to detect it. Stuxnet used a rootkit to hide itself on the infected machine and it would do nothing else but replicate to other inserted USB sticks. For gaining control over the PC it used 5 exploits of which 4 were 0-day exploits found on the day when Stuxnet was first detected. Stuxnet would activate its routines just in case the PC was attached to

## /// DID YOU KNOW?

It is believed that Stuxnet was created to destroy or at least slow down the Iranian nuclear program. Stuxnet physically sabotaged turbines for uranium enrichment by changing the rotation frequencies. Stuxnet was spread using USB. For gaining control over the PC it used 5 exploits from which 4 had not been seen until the day when Stuxnet was first detected. It would activate its routines just in case the PC was attached to a particular Siemens Step 7 controller, and the PC would be used for programming of the controller.

a Siemens Step 7 controller, and the PC would then be used for the programming of that controller. In this case, it would change the frequencies of the turbine rotation system as well as reprogram the tools for automatic response, so it would act as if the system were working correctly. Stuxnet contained a valid certificate, and when it was blacklisted within one-day it changed its certificate. It had death date set on June 24th 2012 when all instances of Stuxnet would destroy itself. It is believed that this malware was created by the intelligence services of USA and Israel [12].

DoQu is malware that had a similar code base to Stuxnet, and it is believed that Stuxnet and DoQu have the same origin and the same authors. DoQu used same exploits as Stuxnet, but with a different purpose. Its purpose was to gather information about victims, in other words its purpose was to spy on infected PCs. DoQu was written using higher programming languages, which is unusual for malware because most malware is written either in assembler, C or eventually in C++, or using some scripting languages such as Python or Lua. DoQu was written in object oriented C, and it is believed that it was compiled using Microsoft Visual Studio 2008.

Flame is the most complex malware that has been seen. It was found in 2012 and most computers infected were located in the Middle East. It is also believed that Flame was created by Israel and the US intelligence services. This is modular malware that can be controlled by an attacker and has the ability to add new modules remotely. With all its modules it can be 20MB in size and is spread over the USB port or by the network. It uses rootkit capability to hide itself on the infected system and has the capability to record audio, video, Skype calls, network activity, to steal files from the hard disk and send to the attacker. At the time, when antivirus companies were gathering samples of Flame for analysis, the attacker sent a kill command, which destroyed all instances of the Flame malware. Flame is written in Lua and C++, and, as with Stuxnet and DoQu, it had a valid stolen certificate.

## CONCLUSION

More than 25 years have passed since the first malware for PC appeared and malware has evolved, however, some of the underlying principles remain the same. The first malware Brain.A spread over floppy disks, Stuxnet, one of

the most complex malware is spread using USB drives. The purpose and motives for malware creation have changed from exhibitionism, through revenge and profit to espionage and sabotage. Profit is still a great motivator for malware creation and it will continue to be so in the future. Military and intelligence gathering motives such as espionage and sabotage have been proven successful for malware creators, and we can expect more military malware and cyber warfare in the future.

It has to be seen how antivirus companies will deal with the kind of attackers who have almost limitless resources for malware creation on the one hand, and those whose motives are profit on the other. Still, we might see some other purposes of malware creation in the future with some game-changing events such as Stuxnet when we are talking about the military use of malware. /

## REFERENCES

- [1] Wikipedia, Malware, Internet: <http://en.wikipedia.org/wiki/Malware>, 03.02.2013.
- [2] Brain: Searching for first PC Virus, Mikko Hypponen, F-Secure, 2011.
- [3] Malware: Fighting Malicious Code, Ed Skoudis, Lenny Zeltser, Prentice Hall PTR, 2003
- [4] Wikipedia, Storm Worm, Internet: [http://en.wikipedia.org/wiki/Storm\\_Worm](http://en.wikipedia.org/wiki/Storm_Worm), 10.02.2013.
- [5] Virus Wikia, Dark Avenger's Mutation Engine, [http://virus.wikia.com/wiki/Dark\\_Avenger\\_Mutation\\_Engine](http://virus.wikia.com/wiki/Dark_Avenger_Mutation_Engine), 17.02.2013.
- [6] Virus creation laboratory documentation, Internet, <http://www.textfiles.com/virus/DOCUMENTATION/vcl.txt>
- [7] One\_half, ESET Threat encyclopedia, Internet <http://go.eset.com/us/threat-center/encyclopedia/threats/onehalf/>
- [8] Concept.A, FSecure Threat description, Internet, <http://www.f-secure.com/v-descs/concept.shtml>
- [9] Maburg, FSecure Threat description, Internet, <http://www.f-secure.com/v-descs/maburg.shtml>
- [10] Dressler, J. (2007). "United States v. Morris". Cases and Materials on Criminal Law. St. Paul, MN: Thomson/West
- [11] Slammer, FSecure Threat description, Internet, <http://www.f-secure.com/v-descs/mssqlm.shtml>
- [12] Stuxnet dossier, Symantec, [http://www.symantec.com/content/en/us/enterprise/media/security\\_response/whitepapers/w32\\_stuxnet\\_dossier.pdf](http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf)

## AUTHOR BIO

Nikola Milosevic was born in Bratislava, Slovakia in 1986. He finished his bachelor and master studies at the School of Electrical Engineering, Department of Computer Science, University of Belgrade (Serbia). Nikola is a contributor to OWASP (Open Web Application Security Project), Anti-Malware project and is the founder and leader of the OWASP local chapter in Serbia. Nikola is currently working as a software developer in Belgrade, Serbia.

