A Survey of USB Exploit Mechanisms, profiling Stuxnet and the possible adaptive measures that could have made it more effective.

Kevin Orrey, MSc

Abstract

Universal Serial Bus, (USB), is today's ubiquitous mechanism for allowing plug and play functionality, enabling quick, fast and easy data transfer between removable and other associated hardware devices. USB provides the flexibility many of us require to carry out day to day functions, however, with every advantage there is a respective downside. USB is also an extremely efficient and extensible exploit mechanism able to deliver, install and run malware, malicious payloads and programs sometimes with extremely limited user and system interaction. Stuxnet was the ideal example of such a mechanism, employing previously unseen techniques to silently install, infect, propagate and execute its malicious payload potentially able to cause significant damage to its intended target. Microsoft Windows is the platform of choice for business, corporations and users alike but consequently has also become the target operating system of choice for would be attackers. This platform was one of the requirements for the Stuxnet worm to be able to carry out its intended function. As a consequence of this and varied other exploit techniques using this functionality, Microsoft after 15 years have finally issued a patch which deals with this long standing vulnerability. This notably adds further restrictions to the use of the autorun functionality and also its integration and use within autoplay. The use of autorun and exploits utilising it was strictly limited to Microsoft products, however, recent research now makes this facility though directly exploitable on the dissimilar Linux platform. Technology and capability is forever changing and the defences employed to mitigate and restrict vulnerabilities opened up by these advances must also. Coupled alongside this though attackers are agile, changing their targets and vectors alike to take into account such changes. Stuxnet was detected potentially a full year after having initially been released, with hindsight potential changes to the code base could have extended this period or for that matter prolonged its life whilst the Incident response process was ongoing.

Key Words: Universal Serial Bus, USB based Exploit, Stuxnet, History of USB, USB Attack Countermeasures.

1 Introduction

Gone are the days whereby there were limited if any portable devices that could be plugged into a computer network and work straight out of the box without any extra drivers installed. Today we have plug and play, meaning any number of devices, using a vast array of services such as Bluetooth, Firewire, Universe Serial Bus (USB) etc. can be utilised on our home and corporate network. This gives us choice in our ways of working, freedom and the flexibility to fulfil our IT and working needs in a changing work environment. With this increase in technology though, comes with it different threat vectors which must be identified and addressed to ensure adequate security can still be provided to protect our assets.

Early adopters of such technology lay themselves more open to such threats, security researchers and attackers alike find holes in new products only after they have been initially released, able to study the hardware itself, its standards and protocols and the software utilised for accessing it. Vulnerabilities once discovered, are exploited, patched and closed, further down the software and hardware product lifecycle as products mature and technology progresses other security holes may be discovered or previously closed ones reinvented through different attack avenue and the exploit, patch remediate cycle continues. Early adopters have to go through the pain, exposure and cleanup operations from such initial

attacks until the technology employed reaches a predominantly more secure stage. It is only after this indeterminate period of time that adoption for main stream users is a safer option.

This paper will look at removable USB devices and will draw upon the history of vulnerabilities associated with them, concentrating on an overview of Stuxnet, the latest attack to utilise this physical security issue. It will survey whether other techniques utilised by other exploit mechanisms could have been employed which could have prolonged its life; proving means to escape detection but once identified, provide possible avenues that it may have restricted the defenders ability to limit its effect, spread and the carrying out of effective remedial clean up action.

2 History of USB

Attacks using USB removable devices have been around for many years, it is just the way they implemented that has changed over time. Throughout the history involving such attacks, the weak link in the chain that can be exploited is yet again the user utilising their naivety and other social engineering techniques to achieve the attackers aim.

The initial specification for USB 1.0 came out in 1996 and has moved along rapidly since this time with version 3.0 released in 2008, although USB 3.0 products were not seen on the market until 2010. In 1996, regular transfer speeds of 12 Mb/s were supported but today this has potentially risen to in excess of 3 GB/s such is the increase in technology and the thirst for ever quicker data transfer mechanisms.

Automatically being able to run programs from removable devices started with CD/DVD drives, whereby placing the file autorun.inf in the root of the media. If the autorun.inf command contained what is known as an OPEN command pointing to a specified program this would execute once the disk had been inserted. This, with a few alterations in the normal configuration settings on windows, would also allow the same thing to happen on USB devices, against the normal behaviour of having the autoplay menu displayed.

Integrated technology that works in conjunction with USB such as U3 was co-developed by SanDisk and M-Systems in 2005. U3 technology, in essence, uses two partitions on a USB device, one which is read-only and which Windows interprets as a CD drive partition. This contains the autorun.inf (autorun) file and associated LaunchPad software. The LaunchPad software then uses the second partition, which is file allocation table (FAT) formatted, which contains a hidden “system” folder from which installed applications can be run from. Thus when a U3 enabled USB device is plugged into a computer it will automatically launch associated applications installed.

Before discussing the latest Stuxnet attack utilising USB, it is best to go through the varied history of attacks using this medium. Before technologies such as U3 were created the way to execute anything from a drive be it USB or for that matter CD/ DVD was to use the autorun feature. This generally worked out of the box to try and aid the user and speed up access to data but unfortunately with the side effect of allowing things to execute with limited user interaction, in essence a boon to the potential attacker.

3 USB Attacks

The evolution of attacks utilising USB as the physical delivery mechanism according to Anderson, (2010), Crenshaw, (2011) and Larimer, (2011) can be broken down into the following attacks:

a. Autorun

Creating an autorun file in the root of the USB drive with the following parameters could potentially be used to exploit a user. Autorun will not execute the program by default and the autoplay windows dialog box will be displayed but getting the user to open folders to view files from this normal windows pop-up is a trivial manner as they expect such things to happen. In addition having suitable icons representing programs to make them look innocuous will add to the credibility of a program/ application.

```
[autorun]
action=Open Files On Folder
icon=icons\drive.ico
shellexecute=badthingshappen.exe
```

b. USB Dumper

USB Dumper, developed by Secuobs, (2011) and released in 2006, was the starting point from which attacks using USB got more and more sophisticated. USB Dumper created a background process on the system and once a USB was plugged into it, it started to copy the contents to a directory created based on the current date. An attacker could then read through the contents of the data at a later time. As you can see this is good but not very useful to a remote attacker who needs to logon to retrieve data.

c. USB Hacksaw

USB Hacksaw from hak5 is an extended version of USB Dumper which addresses the need for a remote attacker to revisit the machine the USB is plugged into. The tool is installed on the system in a hidden folder. Dependent on a user's rights the tool will survive a reboot and starts either from a registry run command or from being placed in the startup folder. Once a USB is plugged in, USB Dumper will copy the files to disk. A batch file is then run (send.bat) which compresses these files using WinRAR. The tool will then utilise stunnel, which allows a user to encrypt TCP connections even when non-SSL aware daemons and protocols are being utilised as the forwarding transport mechanism, to initiate a SSL connection. Blat is then used which allows mail delivery to be carried out using Simple Mail Transport Protocol (SMTP), to deliver the files to a specified mail address. All associated documents and compressed archives are then removed from disk.

d. USB Switchblade

USB Switchblade is an evolution of Hacksaw and although requiring administrative access to the machine to be attacked, it does offer an awful lot of functionality. Different variants and installation methods for this tool exist, notably Amish, Kapowdude etc. These do not rely on U3 technology to work. By far the most favoured version though is the U3 enabled GonZor Switchblade. This version combines all the functionality of USB Hacksaw, Dumper et al but offers an awful lot more including the ability to kill anti-virus software, dump system information, network and varied windows and application user passwords together with installing Virtual Network Computing (VNC). This which would allow a remote attacker to connect to the machine and remotely control it. Plugging in a Switchblade configured USB to a target computer allows pre-configured programs to be executed and their output saved to the USB, enabling a local attacker to quickly acquire sensitive information. In addition hacksaw can also be installed which will then enable an attacker to dump and exfiltrate data from every USB device inserted afterwards.

e. USB-Based Virus/Malicious Code Launch

USB based viruses and malicious code usually use the aforementioned autorun, autoplay or U3 technology to infect hosts. Examples of which include Worm:AutoIt/Renocide.gen!A and Worm:Win32/Nuj.A, etc. These once installed will infect any USB device utilised on the system, creating custom autorun files on the device which will then execute if plugged into other hosts.

f. USB Device Overflow

There have been a couple of occasions whereby the act of actually inserting the USB device into a computer has allowed an attacker to execute their own code. These attacks were presented at BlackHat, by SPI Dynamics, (2005) and MWR Labs, (2009) at Defcon respectively and although used different hardware solutions to initially create there attack both their end goals achieved the same aim, causing a buffer overflow in a driver allowing the ability to run their own code. Once a USB device is inserted, vendor identification (VID) and product identification (PID) take place and the associated driver is loaded into memory. Multiple PIDS can be designated so the attacker need only alter the PID to match that of a vulnerable driver to enable the exploit to occur. This may sound simplistic but an extensive knowledge of hardware and software is required for this to succeed.

g. Social Engineering and USB Come Together for a Brutal Attack.

BinarySec, (2009) define social engineering as the "*art of manipulating persons in order to bypass security measures and tools*", Anderson states that users have many traits that can be exploited and one of the main ones is their naivety. Baiting, pretexting, phishing, whaling and quid pro quo attacks can be utilised as a means of delivery or enticement for users. Users welcome gifts or things that are free, or if asked like to assist/ help out if someone is in need. Each of these foibles can be exploited in conjunction with the humble USB device to attack users. Examples of such attacks are common, a user finding a USB device on the floor or is given one at a conference who consequently plugs it into their system and a payload is executed. This technique has been used on several occasions as a successful malware delivery mechanism, DarkReading, (2006).

h. Programmable HID USB Keyboard, Mouse and Dongle Devices.

Crenshaw, (2010), Pisani et al (2010) and SecManiac, (2010) all discuss the use of hardware peripheral devices that can be used for nefarious purposes, either as keyloggers, as delivers of malware or exploit code direct to the underlying operating system (OS). Any innocent peripheral device can be modified to carry out a number of malicious functions by replacing the USB microcontrollers with those from third party suppliers, (Teensy by PJRC et al).

i. Hardware key loggers

PS2 used to be the hardware mechanism of choice for keyloggers, these have, as technology has advanced, been replaced by their more extensible USB cousins as the favoured means to record the keystrokes of unaware users according to Crenshaw, (2010) and SANS, (2007). These devices are small, easily and cheaply purchased and can be left deployed for long periods of time due to their large storage capabilities as was highlighted recently by Sophos, (2011).

j. USB Autorun attacks against Linux

Whilst not totally dissimilar to autorun attacks on the Microsoft Windows platform, Larimer, (2011) recently provided an insight into vulnerabilities that could exist on the Linux platform. Larimer demonstrated that code could be executed when a USB device is connected utilising a combination of USB storage subsystem, file system, user mode or Kernel level drivers, together with vulnerabilities within Desktop applications.

k. Tainted USB devices

A recent report by The Enterprise Strategy Group, (2010) when they assessed Cyber Supply Chain Security Vulnerabilities from within critical U.S. infrastructure revealed a lack of effective security procedures which may leave them open to attack. Manufacturers that do not safeguard their supply chain and internal build programs have sometimes left themselves open to their product range being manipulated with associated malware being installed upon them. As reported by rationallyparanoid.com (2010) and Zdnet, (2008), there have been many instances of aforementioned USB products being shipped to customers pre-infected with malware. Customers in turn utilise these products and unbeknownst to them become infected through no particular fault of their own. This self same attack mechanism has been seen in the past on a number of other mediums, notably, CD/ DVD and floppy drive and this is just the next iteration of such an attack vector being utilised.

4 USB Attack Countermeasures

4.1 Introduction

Defence in depth when employed correctly employs multiple defensive layers and controls within a network to provide an effective shield and protection from a variety of attack vectors. SANS, (2007) recommends a number of strategies adopting this, be they from a vector orientated, information centric or other associated stance.

One of the major defences any network can have is ensuring that it has undergone Accreditation. This should ideally ensure that effective controls, processes and policies have been put in place to risk manage any perceived vulnerabilities within the network together with identifying likely threat sources and actors. This is documented in a Risk Management and Accredited Document Set (RMADS). A RMADS should document interconnections with other systems, what controls have been put in place. The Communications and Electronics Security Group, (CESG), (2009) Information Assurance Standard No.1 - Technical Risk Assessment and other CESG Good Practice and Readiness guides alongside a number of industry best practices provide a sound starting point for its compilation.

For UK HMG systems, a number of Information Assurance requirements are mandatory for organisations as stipulated by the Cabinet Office, (2011) and these respective policies and controls that relate to IT networks should ideally be referenced in the RMADS also. These policies require an organisation to adopt certain "*mandatory security requirements and management arrangements*" notably in the areas of:

a. Governance, Risk Management and Compliance.

b. Protective Marking and Asset Control.

c. Personnel Security.

d. Information Security and Assurance.

e. Physical Security.

f. Counter-Terrorism.

g. Business Continuity.

For civilian organisations, the International Organization for Standardization, (2005) utilises ISO 27001:2005 to specify the requirements for Information Security Management Systems (ISMS).

Regular independent or in-house Audits, Vulnerability Assessments and Penetration Tests should also be carried out to ensure that vulnerabilities and non-compliance with security policies are identified and procedures put in place to mitigate them at the earliest opportunity.

After the Stuxnet attack which used USB as one of its major forms of propagation, the Cyber Security Forum Initiative, (2010) recommended a comprehensive list of countermeasures that could be employed to mitigate against similar attacks. A survey of these in conjunction with recommendations from other resources broke down these countermeasures into the following areas:

a. Hardware Security.

b. Personal Security.

c. Physical Security.

d. Software Security.

4.2 Hardware Security Countermeasures

Software applications can be deployed as load balancers, proxies, content filters, firewalls etc. to provide security for the network, however, hardware appliances are becoming more and more extensible and prevalent within the DMZ forming a protective shield around the internal network. The following technologies could be adopted:

a. Use of Data Leakage Protection (DLP), Cisco IronPort etc. can provide a mechanism to interrogate email, instant message chat sessions, webmail, file transfer and other forms of communications that attempt to egress from the network. This is based around verifying the data against a specific rule set and filters to determine if a breach has occurred.

b. Host Intrusion Prevention Systems (HIPS) can utilise rule and behavioural monitoring that notes any changes to the file system on the installed machine. HIPS utilise a system whereby a cryptographic checksum is taken of files and any associated changes to them are flagged. This can then be alerted to a central reporting repository for further action when used in conjunction with a Network Intrusion Prevention System (NIPS).

c. Host Intrusion Detection Systems (HIDS) can identify changes to the file system and the creation of new services; Stuxnet used DLL injection, in conjunction with installing a rootkit and these actions may have been detected. Any alerts could then be sent to

a central reporting repository for further action when used in conjunction with a Network Intrusion Detection System (NIDS).

d. Corporate Firewalls and Filtering Appliances should be able to inspect traffic traversing into and out of the corporate network. This would include implementing stateful packet inspection for layer 3 traffic complimented with application inspection to afford maximum protection.

4.3 Personal Security Countermeasures

KPMG, (2010) in their annual Dataloss Barometer report, noted that losses due to insider threats are steadily increasing year on year and now make up almost 20% of the total losses being reported, figure 1. In addition USB media has accounted for 7% of such losses requiring means to be put in place to reduce the losses using this medium:

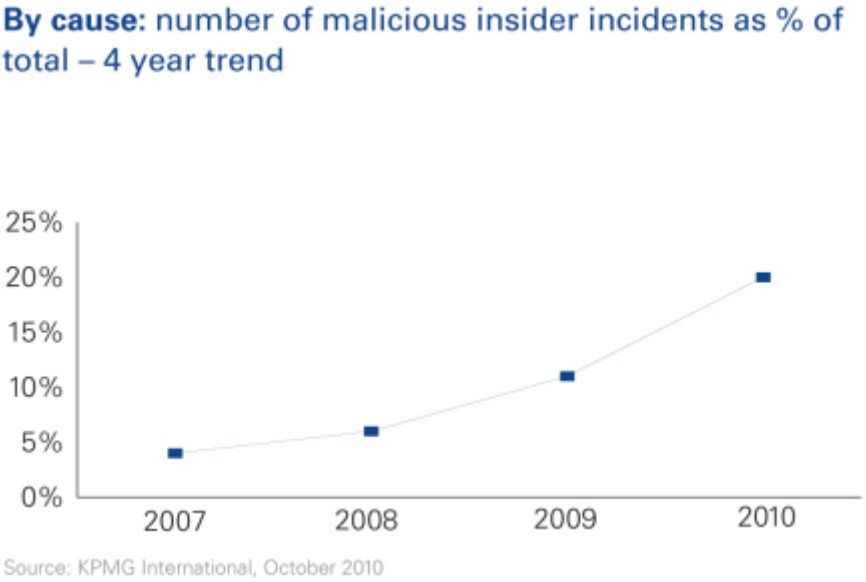

Figure 1 - Malicious Insider Threats (KPMG, 2010)

Noonan and Archuleta, (2008) previously conducted a review of the Insider threat and also identified a number of concerns. As a consequence of this a number of recommendations relating to personal security were identified, most notably that effective employee screening and vetting, education and awareness training and information sharing should be carried out as a way to provide countermeasures to mitigate this threat both from the insider and USB medium attack vector as a whole.

4.3.1 User Education

In a recent report, GTISC, (2008) noted that "*Technology is one piece of the puzzle, regulation is another and user education is the final hurdle*". When all hardware and software security mechanisms fail, the knowledge of the user could be the one thing that could prevent a network being compromised and as such an effective user education and awareness policy should be adopted. This would ideally include:

a. How to deal with USB devices, ensuring they come from trusted sources and the use of anti-virus boundary/ sheep-dip devices.

b. Implementation of robust and achievable Acceptable Use Policy (AUP).

c. Training in Incident Response and Reporting procedures.

d. Training on the varied social engineering techniques that may be utilised by attackers to enable the exploitation of hosts via USB or by other mechanisms.

e. User education training relating to the threats realised from Internet usage, the sharing and transfer of files, this will hopefully ensure a safer browsing experience and that all software and data is obtained from reliable trusted sources.

f. User awareness training on unusual activity; is USB activity continuing for longer than expected, does network traffic start unexpectedly, are any pop-ups experienced asking to initiate a connection or allow an application etc.

4.4 Physical Security Countermeasures

One method to protect against attacks from tainted USB devices and programmable USB peripherals is to instigate effective controls as prescribed by the ISO 28000 series of standards for supply chain security management systems, (International Organization for Standardization, 2007). This may not fully protect the user from all threats but adds to the overarching defence in depth security boundary that is designed to protect the system. Enforcement of the use of encrypted USB drives.

Physical security measures with regards to the prohibition of the use and introduction of USB devices into a site may also provide another form of protection. This would need to be backed up by appropriate security policies and user acceptance of these terms and conditions coupled with the ability to carry out searches etc. if required.

Disabling USB devices in the BIOS is another way to prevent the use of such devices, additionally ensuring that access to the BIOS is protected by a suitably complex password. As with all protective policies that can be applied, various mean exist, according to computersnetworking.info, (2010) and other sources to bypass these countermeasures, notably using manufacturer's default BIOS passwords, resetting the jumpers on the mortherboard etc.

Further hardware methods include the potential use of USB port blockers[1] and locks.

4.5 Software Security Countermeasures

Software security countermeasures can encompass varied lockdowns which can be applied to both the OS and applications alike. Varied lockdown guides exist to assist administrators to secure systems and applications, most notably guides from the National Security Agency, (NSA), (2011), Homeland Security (2011) and the National Institute of Standards and Technology, (NIST), (2011). For Industrial Control Systems, (ICS) the following gives guidance on specific security practices that should be adopted.

a. NIST 800-82: Guide to Industrial Control Systems (ICS) Security.

b. Department of Homeland Security: Catalogue of Control Systems - Security Recommendations for Standards Developers.

4.5.1 Autorun and U3 Attacks

[1] http://www.lindy.co.uk/usb-firewire/usb-light-fan-security-locks/

Autorun can be disabled in many ways, Microsoft, (2010, 2011), provides a number of support guides:

a. Install the relevant security updates MS08-038 (kb953252) and then utilise the Group Policy Editor Tool selecting to “Turn off Autoplay” from within Computer Configuration.

b. Utilise Microsoft’s Fix it for me facilities to auto-fix the issue.

c. Alter the following Windows Registry key modifying the Value data box to 0xFF:

HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Explorer\NoDriveTypeAutorun

d. US-CERT, (2008) suggests also to stop the OS parsing autorun.inf files on the system; this can be achieved by creating the following .reg file:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\
IniFileMapping\Autorun.inf]
@="@SYS:DoesNotExist"
```

Autorun.inf files will then be treated as if they were a pre-Windows 95 application configuration files. This is due to the “IniFileMapping” key instructing the OS to read its sub keys upon encountering autorun.inf files. The “DoesNotExist” value ensures that the autorun.inf file is treated as if it were empty so any command syntax within is not run.

e. Install the relevant security update (KB971029) which extends disabling autorun functionality with regards to the autoplay facility.

f. U3 technology Hacksaw and Switchblade attacks can be mitigated and the level of threat reduced by reducing the privileges of logged on users which may reduce the functionality and impact experienced from these tools.

g. The authors of the Hacksaw and Switchblade tools have released a tool entitled USB Antidote which automatically carries out a number of the above, however, this contains a number of scripts and registry key changes and it may be more prudent to rely on more “proven” software vendors advice due to the possibility that this may harm a user’s system or install “extra” functionality which may be used for nefarious means.

4.5.2 Anti-virus

Virus Bulletin, (2011) carry out a number of comparative tests on a plethora of anti-virus (AV) vendors on a yearly basis to try and gauge the effectiveness of current AV products against four distinct sets of malware samples. A network should have an effective AV product solution installed, which has been appropriately configured and is regularly updated. TrendMicro, (2010) reported that the ZBOT and SALITY family of malware utilised the .LNK vulnerability not long after Stuxnet worm was identified, up to date AV would therefore detect an attempt to install on the system via this attack vector. This is common practice within malware writing taking advantage of previously used exploit mechanisms in the hope that user and corporate networks would not have installed the respective security patches and updated their AV signatures.

Many standalones products exist together with more scalable client server models for corporate environments, (TrendMicro[2], Symantec Endpoint[3] et al).

4.5.3 Installation Restrictions

Crenshaw, (2011) and Microsoft (2007) lists comprehensive group policy and registry tweaks to restrict the installation of removable devices together with ways to restrict devices to certain devices identifiers.

A further way to restrict the use of certain software contained on USB devices is to employ Software Restriction Policies (SRP); this can either be restricted via the creation of Hash, Certificate, Path based or Internet zone rules according to Microsoft, (2011).

Varied security applications exist that can control the use of USB devices, ensuring they are restricted to certain devices and are allowed to be installed and utilised by appropriately authorised personnel only, i.e. Lumension Device Control[4], DeviceLock, GFI Endpoint Security[5] etc.

4.5.4 Patching

A plethora of literary sources, (Microsoft, Krebs on Security and MalwareIntelligence, (2010)) point to the fact that exploitation is becoming more application than OS vulnerability centric and predominantly concentrating on the Oracle Java and Adobe product range, figures 2 and 3 refer. This may be due to the fact that OS vendors are making their platforms more secure, or when a vulnerability has been identified their update mechanism is much more robust with automated Microsoft Update and Windows Service Update Services (WSUS) for the Enterprise to protect users. Certain 3rd party applications have manually configured update programs and are not as extensible as OS variants, the frequency and complexity of needing to manage many different update mechanisms will most probably mean that users are not fully covered. This all combined with inadequate user awareness to the threat posed by not patching 3rd party applications is a boon to attackers which they are actively exploiting. This is backed up by Secunia, (2010) in its half yearly report. A user who has 50 programs installed which will have 3.5 times more vulnerabilities in 3rd party programs than in the Microsoft programs installed. This ratio is expected to rise to 4.4 before the end of 2010.

[2] http://uk.trendmicro.com/uk/products/enterprise/client-server-suite/
[3] http://www.symantec.com/business/endpoint-protection
[4] http://www.lumension.com/device-control-software/usb-security-protection.aspx
[5] http://www.gfi.com/endpointsecurity

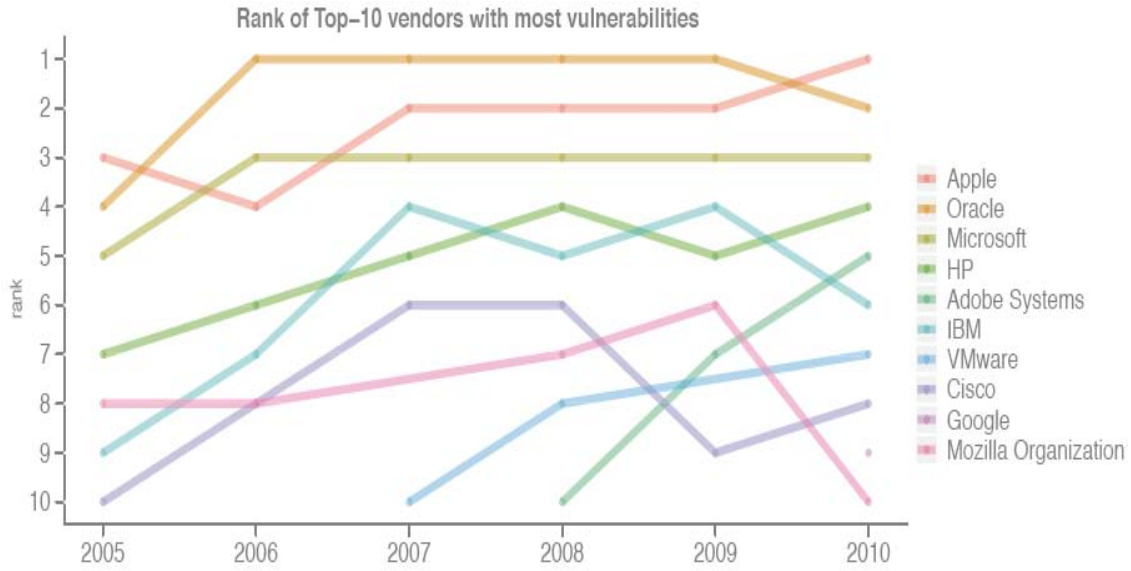

Figure 2 - Ranking of the Top-10 vendors with most vulnerabilities per year (Secunia, 2010)

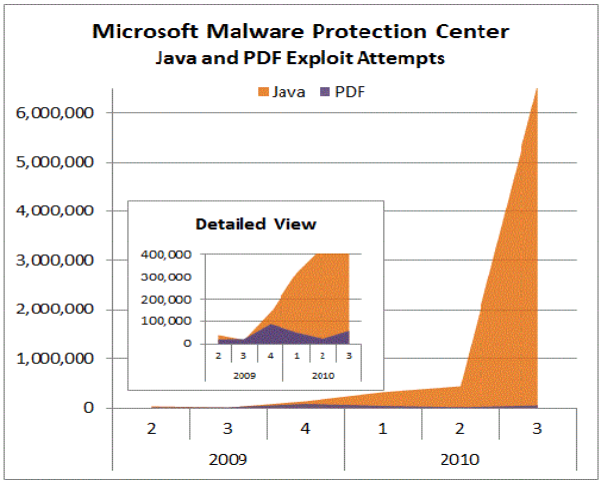

Figure 3 – Java and PDF Exploit Attempts (Microsoft, 2010)

Protection against 0-day attacks is difficult, bordering on impossible, but a common theme within the plethora attack vectors utilised today, especially within corporate networks is that they for, the most part, use tried and tested exploits for previously released vulnerabilities to propagate as was seen with one of the propagation vectors for the Stuxnet worm, Hakin9 magazine, (2009). Installing MS08-67, the patch which fixes the Windows Server Service vulnerability would have stopped propagation via SMB so may have reduced the spread of the Stuxnet worm by this vector.

Keeping systems up to date with the latest patches for the users' OS, web browser of choice and applications, notably flash player, java, adobe etc. could potentially stop an exploit attempt. OS specific updates from Microsoft or any other OS vendor just focus on the OS, browser, and office programs etc. it is the users' responsibility to update the add-ons applications they have installed. Given the lack of user awareness and education, this is normally a weak point attackers can exploit.

Various tools exist that assist with identifying missing patches, some are OS specific, i.e. Microsoft Baseline Security Analyser, (MBSA), GFI Languard, Tenable Nessus etc. others

though will cover 3rd party applications also making them much more extensible, i.e. Secunia Personal Software Inspector (PSI). Users can then ensure that missing patches are updated.

4.5.5 Sandboxing/ Virtual Environments

Running virtual machines or thin clients in a network would potentially not stop infection and potentially propagation but those organisations implementing such solutions benefit from the fact when a user logs off they potentially get an untarnished new image which has no infection associated from it from a centralised server. In evaluating the security benefits afforded from the use of thin clients in an organisations security environment, Intel, (2010) identified a number of areas that provided added security, notably; "*prevention of physical data loss, removal of administrative privileges, limitations on installed applications, client integrity, and ability to roll back to a known good state".* Vassilev, (2007) and Principled Technologies, (2007) provide further evidence that utilising such environments reduce the security risk to the host and consequently network has a whole. All three sources and others note the one big drawback with this mechanism. This is the need to preserve the sanctity of the central "known" good images that each user or server instance runs or is supplied with; should this be infected it may possibly lead to a catastrophic spread of the worm. NIST, (2011), provides comprehensive coverage of virtualisation and security in the safe deployment of such technologies which may alleviate the perceived drawback identified. This would allow for thorough control of build states, segregation and updating mechanisms and the employment of a comprehensive and effective defence in depth security policy.

4.5.6 Disable Unnecessary services

Disabling unnecessary services provides an extra security mechanism and performance gain for any system according to Birkholz, (2003), Dubrawsky, (2009) and Krutz et al (2010). Resources are utilised by services and associated TCP and UDP ports may be left open listening for connections with associated protocol service related traffic being generated. When conducting varied Penetration Tests, services that are not being utilised on the system are one particular attack vector that could potentially be exploited. In addition, vulnerabilities are continually being researched and identified over a wide range of services and platforms and a minimalistic approach to what services are left enabled could thwart a would-be future attack or limit the available attack surface for onward ingress into the network.

4.5.7 Default Usernames and Passwords

The majority of software and hardware appliances and applications come with default usernames and passwords to provide initial setup and administrative access. Numerous online resources exist that document the existence of these, Cirt Inc, (2011), Phenoelit, (2010) and Orrey, (2010) amongst others. An attacker with remote access to a hardware device or software application front-end will no doubt try these default username/ password combinations to gain initial access. Mansourov et al (2010), Salomon, (2010) and numerous other resources have continued to identify this vulnerability and recommend that all default username and passwords should thus be changed. The Stuxnet worm, as an example, took advantage of this, utilising a default username and password to connect to the backend WinCC database and execute out SQL commands.

4.5.8 Audit Logs

Bayuk, (2010), Shiller, (2010), Vacca, (2010) et al, all point to the forensic and investigative value of audit logs. Prevention, detection and responses that are required pre, during and post attack can be greatly enhanced by enabling software auditing on all systems at both the

OS and application layers. As a consequence of carrying out regular and comprehensive reviews of the logs nefarious actions may be identified. This is recommended by

5 Stuxnet Overview

The Stuxnet worm's[6] end goal according to Symantec, (2011), was to sabotage and reprogram industrial control systems (ICS) utilised in gas pipelines and power plants. This was to be achieved by modifying code within specific types of programmable logic controllers (PLC) which controlled frequency converter drives that maintained the speed of varied motors. Stuxnet would ensure that these motors would speed up and slow down at varied intervals, thus causing damage to the system as a whole as the system was not designed to withstand such changes in motor speeds. Four variants of the worm were identified:

a. Variant 1 – Compiled on Mon Jun 22 16:31:47 2009
b. Variant 2 - Compiled on Mon Mar 01 05:52:35 2010
c. Variant 3 - Compiled on Wed Apr 14 10:56:22 2010
d. Variant 4 - Is likely to exist but has yet to have been recovered

Three waves incorporating five attacks were carried out by the worm against 5 specific organisations in Jun, Jul 2009 and Mar, Apr and May 10. Based on retrieved information Symantec were able to partially provide a pictorial representation detailing the spread and success of each particular campaign, (figure 4). These attacks amounted to 12000 separate infections alone from the 100,000 total recorded, the remaining 88,000 infection could be down to collateral damage caused by the many ways the worm was able to propagate.

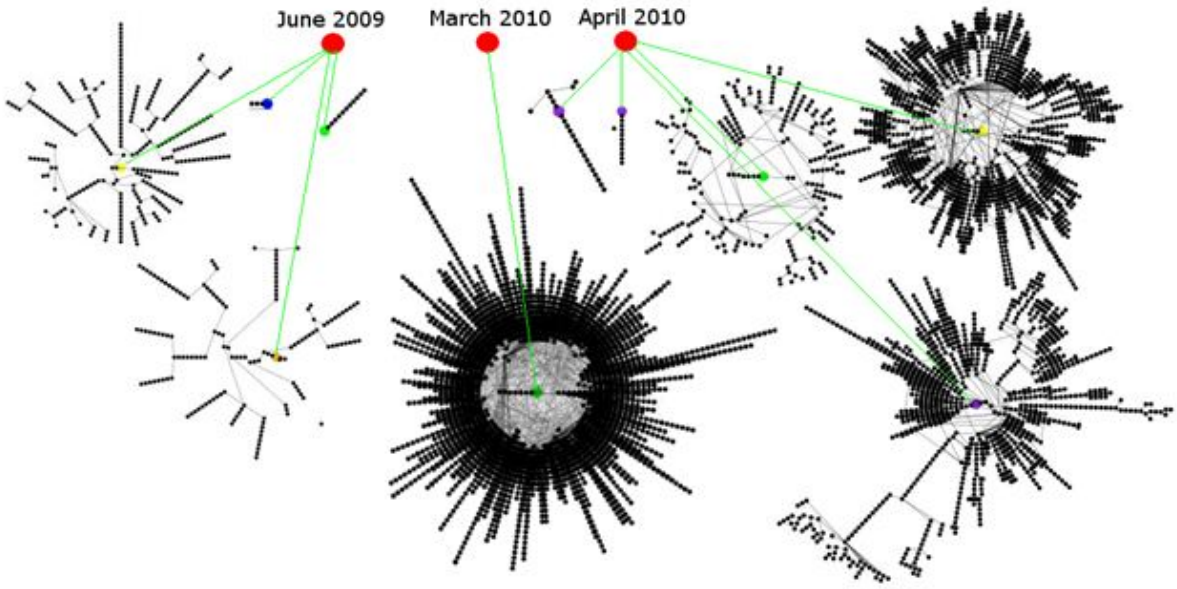

Figure 4 Stuxnet Cluster of Infections (Symantec, 2011)

USB devices were to be utilised within the attack as a means of propagation, enabling the worm to spread quickly and effectively. These devices also enabled the possibility of jumping air-gaps between Internet connected and closed networks. Stuxnet was programmed to try and identify Field Programmable Gateway (PG) devices, which usually take the form of a Windows based laptop, used to program PLC's via proprietary Step 7 and

[6] A worm is a piece of malware that self-replicates

WinCC software. The latter is used in Supervisory Control and Data Acquisition (SCADA) systems as a Human-Machine Interface (HMI) and allows interaction with Step 7 projects and files. The Dynamic Link Library (DLL) S7OTBXDX.DLL used by Siemens WinCC systems was replaced by Stuxnet which allowed it to read/ write and control the PLC's. Stuxnets' need to find the Field PG's made propagation via USB a key element in the whole process.

In order to achieve this and ensure that user's were not aware of the attack, the worm needed to be built in such a way to:

a. Defeat Antivirus Products so that its actions would not be flagged as suspicious.

b. Propagate to other machines via user interaction; utilisation of network shares etc. ensuring a check is first conducted to ensure the machine is not already infected.

c. Hide within plain site, i.e. install visible files to disk but using rootkit technology to disguise itself from system and user defences by integrating itself within valid system processes. This would allow it to function unhindered both on the PLC and also the base OS.

d. Be controlled by the attacker via a Command and Control system to allow:

    i. An encrypted auto update mechanism,

    ii. Carry out a survey of infected machine identifying OS details, installed software (including Step7/ AV variants), IP addressing etc.

    iii. Provide the ability to remotely execute commands sent from the attacker.

e. Have an inbuilt payload and the commands to execute it when the correct PLC's have been infected (for attacking those system not connected to the Internet).

f. Utilise driver files that were digitally signed to ensure its underlying code base was verified[7].

g. Reduce collateral damage and the spread of the worm by limiting its propagation to three machines only.

Stuxnet used a number of 0-Day vulnerabilities, (which will be discussed), to propagate and also used previously unseen privilege escalation techniques to gain the right amount of privileges to be able to initially install itself, remain resident in memory and survive a reboot.

5.1 Stuxnet Installation Routine

Installation

Stuxnet carried out a number of checks before it installed itself on a target machine or removable drive, to ensure that:

a. It was not already installed[8] [9].

[7] This would be achieved by using digital certificates stolen from Realtek Semiconductor Corporation and JMicron Technology Corporation.
[8] Ensure the Windows Registry Key HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\MS-DOS Emulation was not present and if present did not have the value 19790509 set.

b. The operating system was of a specific type[10].

c. The date has to be before 24 Jun 12, the date that Stuxnet has been programmed to stop spreading, although no evidence has been found to date why this date is important.

d. A suitable Antivirus product was installed and could be utilised.

e. It had suitable installation privileges i.e. Admin or could acquire them via a Privilege Escalation Attack. Stuxnet utilised two such attacks dependant on the OS targeted:

    i. Windows 2000/XP used the Win32k.sys (MS10-073 refers[11]) windows kernel-mode driver vulnerability which loaded a specially crafted keyboard layout allowing code to be run with SYSTEM privileges.

    ii. Windows Vista+ used the Task Scheduler (MS10-092 refers[12]) vulnerability whereby scheduled tasks can be run without the OS properly validating the request, allowing commands to be run with SYSTEM privileges.

f. It could communicate with Command and Control servers (although not required to execute its planned payload to attack PLC's).

The installation process is summarised in figure 5.

[9] A certain amount of debate over this value continues, as potentially it denotes the date Habib Elghanian was executed by firing squad in Iran, this may be a ruse or give an indication of political motivation behind the attack. (Hack In the Box, 2010)
[10] Must not be 64-bit and be Windows 2000 or higher.
[11] http://www.microsoft.com/technet/security/bulletin/ms10-073.mspx
[12] http://www.microsoft.com/technet/security/bulletin/ms10-092.mspx

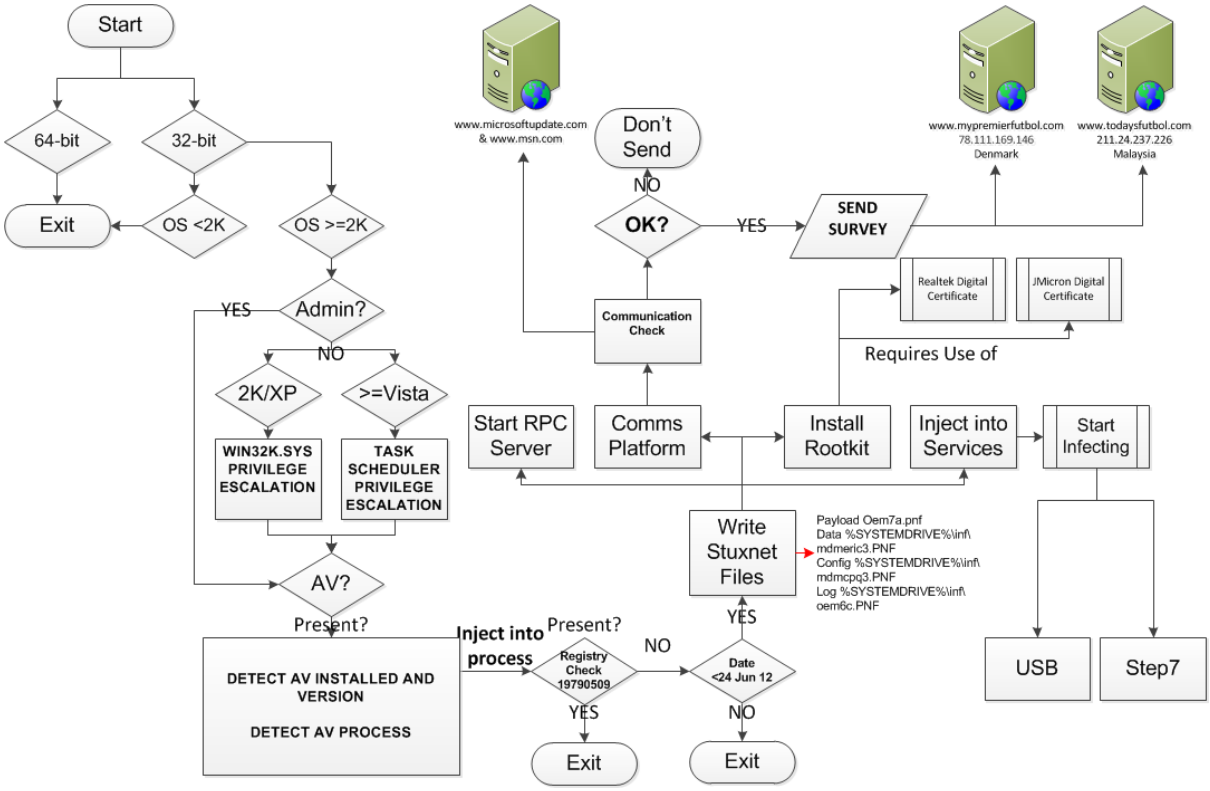

Figure 5 - Stuxnet Install Process

5.2 Stuxnet Propagation Routine

Stuxnet propagated, according to Symantec, (2011), Kaspersky, (2010) and TrnedMicro, (2010) in a number of ways; via the use of removable USB media, and across the network in multiple ways. The following methods were utilised:

a. Removable Media (USB) – The MRNet.sys file which forms part of the rootkit intercepts access to all I/O requests from USB devices to the base OS. As such Stuxnet is able to intercept read and write requests and copy itself to the device (figure 6). The .LNK files are the actual exploits that load and execute the .tmp files which then drop Stuxnet to disk when the USB device is re-inserted into a further machine. The vulnerability it exploited in the explorer process needs only to render the contents of the USB drive for it to propagate, (MS10-046 refers[13]).

   Note: - There have been four detected variants of the Stuxnet worm, the oldest of which dates back to Jun 09 which used autorun as its means of propagation before the .LNK vulnerability was utilised.

[13] http://www.microsoft.com/technet/security/bulletin/ms10-046.mspx

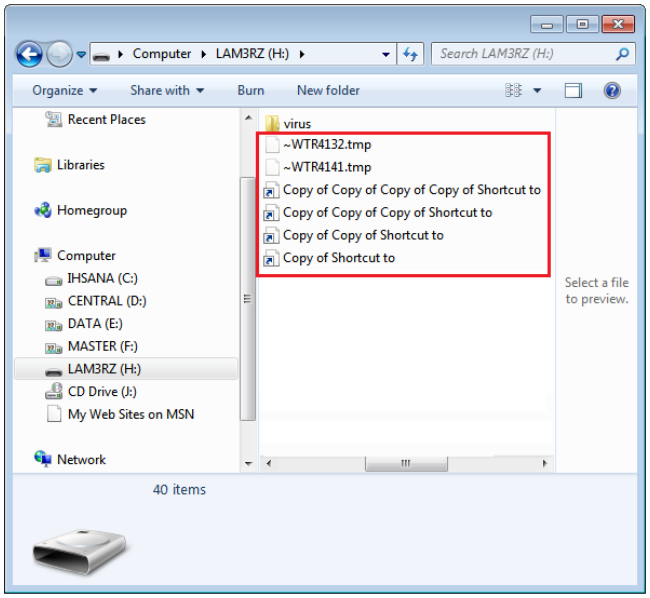

Figure 6 - Stuxnet USB Presence (Ihsana IT Solution, 2010)

b. Peer to Peer – The Remote Procedure Call (RPC) Server started as part of the installation process listens for connections, a RPC Client will connect and determine if there version of Stuxnet is up to date, if not it will be updated by the RPC Server.

c. WinCC – Stuxnet will send malicious SQL queries to the WinCC SQL Server Database using a hard coded default password that cannot be changed by the vendor, (CVE-2010-2772 refers[14]). This enables the copying and execution of Stuxnet to the remote host.

d. Network Shares – Using user credential tokens or the explorer.exe process users on the domain are enumerated and Stuxnet is installed to remote shares using Windows Management Instrumentation (WMI) and the scheduling service[15].

e. Print Spooler Vulnerability (MS10-061 refers[16]) – The Print Spooler process allows files to be written to the %SYSTEM% folder and executed if a user is sharing a printer on the network.

f. Windows Server Service Vulnerability (MS08-067 refers[17]) – Uses a previously identified vulnerability to connect via Server Message Block (SMB) to copy itself to a remote machine. Stuxnet would check that the patch is not installed and AV signatures were no newer than 1 Jan 09 to ensure it was not “caught” during this process.

An example of the propagation process is summarised in figure 7.

[14] http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2010-2772
[15] This schedules a job to execute 2 minutes which starts the Stuxnet process ensuring it remains resident on disk and runs on startup.
[16] http://www.microsoft.com/technet/security/bulletin/ms10-061.mspx
[17] http://www.microsoft.com/technet/security/bulletin/ms08-067.mspx

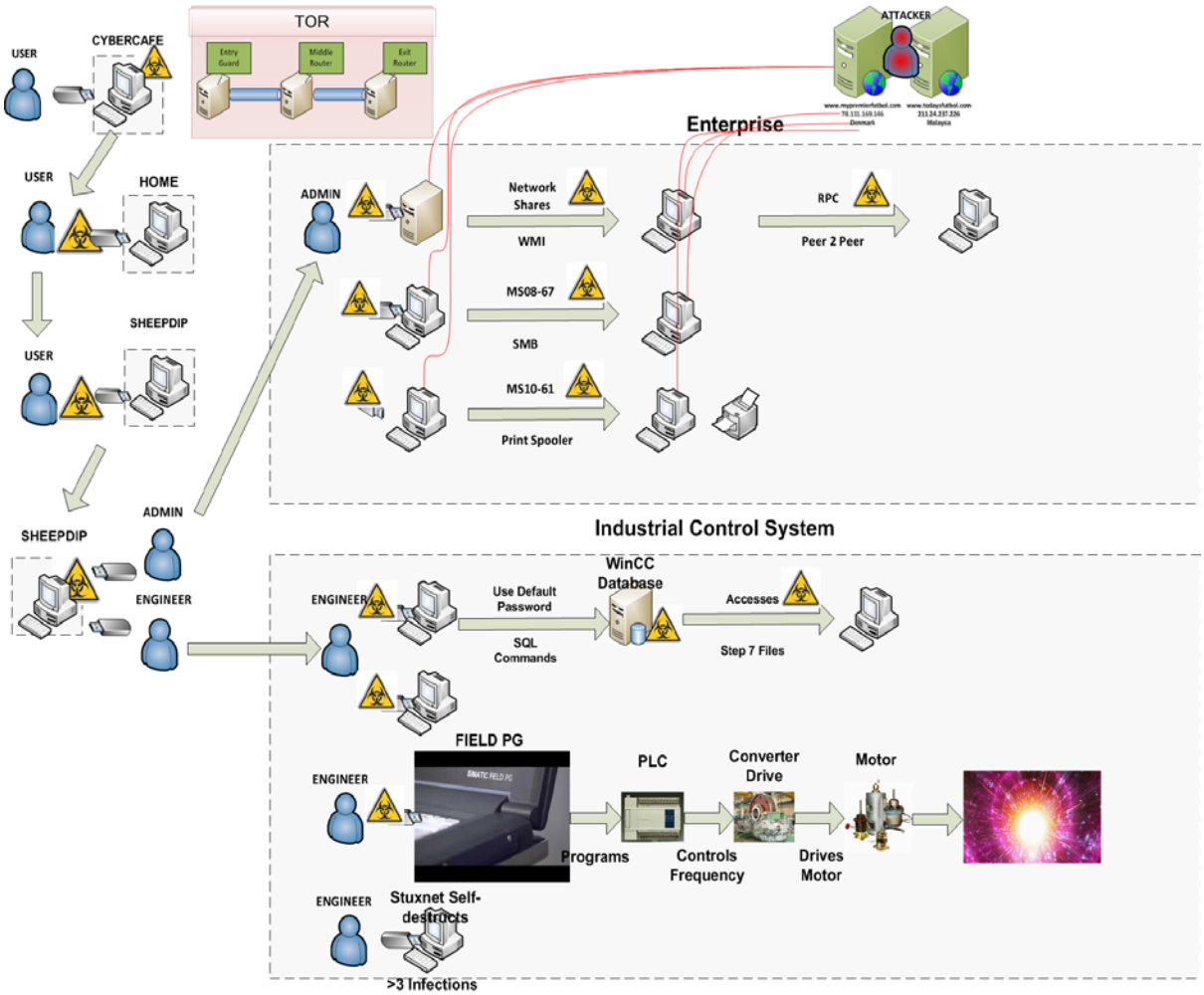

Figure 7 - Stuxnet Propagation

5.3 Stuxnet Adaptation Techniques

A survey of numerous resources, (Davies et al, (2009) and SCMagazine, (2010) amongst others) identified a number of other technologies that given the complexity and thought that had gone into coding Stuxnet may have been utilised to improve on and obfuscate the Stuxnet worm further. These may have prevented Stuxnet from being detected by anti-virus and other security vendors and in addition it may have also made it more difficult for cleanup and effective incident response to be carried out once it had been detected.

Domain Generation Algorithm

According to Symantec, (2011) the Stuxnet worm was coded in such a way to allow the update and change of the original command and control domains, yet this did not happen. According to varied sources, Shantanu Ghosh, (Symantec India) is quoted as saying that the malware writers expected to lose their control servers "*so they built in a P2P update function to prepare for that eventuality*". This would be forward thinking but the worm itself would only be able to update itself to the latest version of itself dependant on the other hosts it could contact and thus would not be able to received new commands. P2P/ RPC network traffic would most likely be restricted to the internal LAN so it would have proven difficult to use full P2P functionality to carry out updates etc. An alternative method that could have been employed that may have provided a more stealthy and resilient solution would have been the use of the Domain Generation Algorithm (DGA) in the core command and control

code. Ligh et al (2010) point to the extensive use of DGA in malware whereby specific strains, Conficker, Kraken etc. have been seen to take an input for example the date or time and then utilising DGA generate a list of domains that the worm will contact. Conficker required the set up of the Conficker working group[18], to deal with this piece of malware. Potential employment of DGA may have provided the ability of Stuxnet to cycle through URI's and corresponding command and control servers and thus hinder the security community's efforts to take down the servers and thus remove the worms main control and update mechanism. In this way incident response would have been much more difficult.

Fast flux attacks

Fast flux attacks have been seen in numerous phishing and malware attacks in the past notably the DanMec Bot, (MalwareIntelligence, 2009), Storm botnet (TheRegister, 2007). The basic premise of fast flux attacks is that the command and control domains have a rapidly changing set of multiple IP addresses, usually comprising previously compromised hosts that are swapped out at very short intervals through quick changing DNS records, (SpamHaus, 2011). In combination with this, and as a secondary protection mechanism, blind proxy redirection is sometimes utilised, according to the HoneyNet Project, (2008). Blind proxy redirection entails redirecting the original domain request which initially lands at the front-end IP of a compromised host which simply forwards the request further to the final backend server, figure 7.

For further protection and obfuscation and to add an extra layer of redundancy fast flux attacks could use ever changing and evolving DNS Address (A) records and authoritative Name Server (NS) records for each command and control domain.

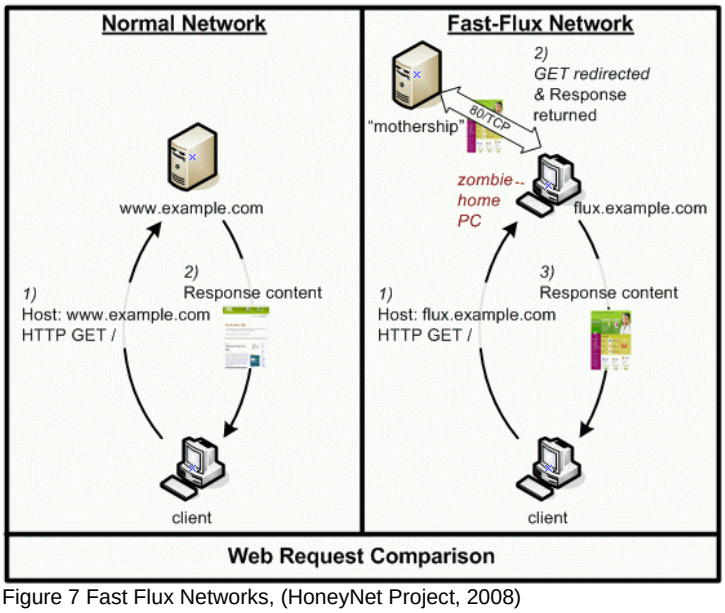

Figure 7 Fast Flux Networks, (HoneyNet Project, 2008)

Rock Phish Attacks

Rock Phish Attacks are generally utilised according to MarkMonitor Inc., (2008) in so called large-scale phishing attacks which can be attributed to criminals who have prior purchased a large number of domains, usually with meaningless names i.e. wasu69.biz. A phishing attack would then prepend the real domain name to be targeted onto these domains i.e.

[18] http://www.confickerworkinggroup.org/wiki/

http://www.hsbc.com.id345.wasu.biz. The id345 is a unique identifier that is used to defeat potentially spam filtering technology. The attackers DNS is configured to process all similar URI's as a wildcard all of which resolve to a single IP address which is actually a proxy server. All web requests are then relayed to an obfuscated server hosted elsewhere. Attempts then to takedown the proxy requires the attacker to just change the DNS entries to a new one and so traffic will be automatically re-routed and attacks will continue. Whilst the use of this technique would not be suitable in this respect, the use though of unique identifiers prepending the command and control server URI could provide a means to better manage hosts into disparate geographic regions. Stuxnet infections were seen world-wide and thus different networks could be given more fine grained control if country or network specific URI's were utilised, a potential adaptation using rock phish techniques could thus make the worm more easily managed. This could potentially provide a more extensible mechanism for updating, reporting and migration giving the potential for selective culling of compromised hosts if they were found to have infected a host/ network that was not of interest.

Obfuscated files and binaries

A survey of malware and the processes they use to try and evade antivirus software determined that a large number of disparate and well known strains utilise pseudo-random files to obfuscate files, dynamic link libraries (dll) and binaries together with obfuscating their respective registry locations, (zeus, AVG, (2010), Conficker-b, Microsoft, (2010) et al). This makes it that much more difficult to detect and cleanup the malware itself. Stuxnets' core files and dll had defined filenames and thus did not provide a further level of obfuscation to defeat a simple file search.

Antivirus (AV) vendors also rely on MD5 hashes of specific filenames to identify the presence of malware, (Ligh et al, 2010), with major AV vendors storing and checking the hashes of known malicious binaries alongside there respective Portable Executable (PE) formats, however, certain strains ensure the MD5 hash of the malware changes every time the malware executes, thus defeating this check, Downloader-CJX etc. (McAfee, 2010). With Stuxnet, again, no attempt was made to obfuscate files and processes which may have prolonged its life pre detection.

6. Conclusion

Stuxnet was a game changer in the way it was targeted, delivered and propagated, never before as such a complex worm been released, yet when it was discovered it was relatively easily taken down and its command and control servers severed. Conficker and other such malware strains were not as complex yet caused so much more pain, Stuxnet although technically complex could have been better, perhaps in hindsight future versions may learn from this and use such technologies as DGA, fast flux and obfuscation, only time will tell.

The history of attacks utilising USB is a long and ever progressing road. Migration using autorun from Microsoft Windows to Linux shows that other attack avenues and platforms are able to be exploited. More and more will Linux/ Unix and Windows co-exist and have to interoperate and thus exploits will and should cross these fine OS lines more often in the future.

A multitude of countermeasures to thwart such attacks are available and corporations and organisations alike should employ a comprehensive defence in depth strategy to mitigate them. At the end of the day though, the user is the final hurdle in any defence that's why education is truly key to defending any network.

References

Anderson Brian, Anderson, Barbara, (2010) "*Seven Deadliest USB Attacks*" Syngress

AVG, (2010) "*Zeus 2.0*" [Online], Available: http://viruslab.blog.avg.com/2010/04/zeus-20.html [Accessed 20 Feb 11]

Bayak, Jennifer, (2010) "*CyberForensics: Understanding Information Security Investigations*" Springer

BinarySec, (2009) "*Social Engineering*" [Online], Available: http://www.binarysec.com/cms/docs/resources/glossary/p-s.html [Accessed 16 Feb 11]

Birkholz, Eric Pace, (2003) "Special ops: host and network security for Microsoft, UNIX, and Oracle" Syngress

Cabinet Office, (2011) "*HMG Security Policy Framework v5*" [Online] Available: http://www.cabinetoffice.gov.uk/resource-library/security-policy-framework [Accessed 21 Feb 11]

CESG, (2009) "*HMG IA Standard No.1 - Technical Risk Assessment - Issue 3.51, October 2009*" Communications Electronics Security Group

CESG, (2009) "*Good Practice Guide 23- Assessing the Threat of Technical Attack Against ICT Systems*" Communications Electronics Security Group

CIRT, Inc., (2010) "*Default Passwords*" [Online] Available: http://cirt.net/passwords [Accessed 21 Feb 11]

Computersnetworking.info, (2010) "*Reset or Bypass (hack) BIOS Password*" [Online], Available: http://www.computersnetworking.info/reset-or-bypass-hack-bios-password.html [Accessed 20 Feb 11]

Crenshaw, Adrian, (2011) *"Plug and Prey: Malicious USB Devices"* [Online], Available: http://www.irongeek.com/downloads/Malicious%20USB%20Devices.pdf [Accessed 14 Feb 11]

Crenshaw, Adrian, (2011) *"Malicious USB Devices: Is that an attack vector in your pocket or are you just happy to see me?"* [Online], Available: http://www.irongeek.com/downloads/malusbphreaknic.pdf [Accessed 14 Feb 11]

CSFI (2010) "*Stuxnet Report v1.0*" Cyber Security Forum Initiative

DarkReading, (2006) "*Social Engineering, the USB Way*" [Online], Available: http://www.darkreading.com/security/perimeter-security/208803634/index.html [Accessed 16 Feb 11]

DarkReading, (2010) "*Stuxnet: An Amateur's Weapon*" [Online], Available: http://www.darkreading.com/blog/228200580/stuxnet-an-amateur-s-weapon.html [Accessed 14 Feb 11]

Davies, Michael, Bodmer, Sean, Lemasters, Aaron, (2009) "*HACKING EXPOSED MALWARE & ROOTKITS: MALWARE & ROOTKITS SECURITY SECRETS & SOLUTIONS*" McGraw Hill

Dubrawsky, Ido, (2009) "Eleventh Hour Security+: Exam SYO-201 Study Guide" Syngress

EnterpriseStrategyGroup, (2010) "*Assessing Cyber Supply Chain Security Vulnerabilities Within the U.S. Critical Infrastructure"* [Online], Available: http://www.enterprisestrategygroup.com/media/wordpress/2010/11/ESG-Research-Report-Cyber-Supply-Chain-Security-Nov-10.pdf?utm_source=website&utm_medium=reportpage&utm_campaign=cybersupplychain [Accessed 17 Feb 11]

Georgia Tech Information Security Centre, (2008) "*Emerging Cyber Threats Report for 2009 - Data, Mobility and Questions of Responsibility will Drive Cyber Threats in 2009 and Beyond*" [Online] Available: http://www.gtisc.gatech.edu/pdf/CyberThreatsReport2009.pdf [Accessed 20 Feb 11]

GFI Software, (2011) "*GFI Endpoint Security*" [Online] Available: http://www.gfi.com/endpointsecurity [Accessed 20 Feb 11]

Hakin9, (2009) "*Print Your Shell*" [Online] Available: http://hakin9.org/magazine/885-my-erp-got-hacked [Accessed 20 Feb 11]

Hack In The Box, (2010) "*19790509: The mysterious number inside the Stuxnet worm*" [Online], Available: http://www.hackinthebox.net/tag/habib-elghanian [Accessed 20 Feb 11]

hak5.org, (2011) "*Amish*" [Online], Available: http://www.hak5.org/releases/2x02/switchblade/AMISH1.0-payload.rar [Accessed 14 Feb 11]

hak5.org, (2011) "*USB Antidote*" [Online], Available: http://www.hak5.org/w/index.php/USB_Antidote [Accessed 14 Feb 11]

hak5.org, (2011) "*USB Hacksaw*" [Online], Available: http://www.hak5.org/wiki/USB_Hacksaw [Accessed 14 Feb 11]

hak5.org, (2011) "*USB Switchblade*" [Online], Available: http://www.hak5.org/wiki/USB_Switchblade [Accessed 14 Feb 11]

Homeland Security, (2011) "*Catalog of Control Systems Security: Recommendations for Standards Developers*" [Online] Available: http://www.us-cert.gov/control_systems/pdf/Catalog_of_Control_Systems_Security_Recommendations.pdf [Accessed 20 Feb 11]

Honeynet Project, (2008) "*HOW FAST-FLUX SERVICE NETWORKS WORK*" [Online], Available: http://www.honeynet.org/node/132 [Accessed 20 Feb 11]

Intel, (2010) "*Evaluating Thin-Client Security in a Changing Threat Landscape*" [Online], Available: http://download.intel.com/it/pdf/Evaluating_Thin_Client_Security.pdf [Accessed 16 Feb 11]

International Organization for Standardization, (2007) "*New suite of ISO supply chain management standards to reduce risks of terrorism, piracy and fraud*" [Online], Available: http://www.iso.org/iso/pressrelease.htm?refid=Ref1086 [Accessed 17 Feb 11]

International Organization for Standardization, (2005) "*ISO/IEC 27001:2005*" [Online], Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=42103 [Accessed 17 Feb 11]

Ihsana IT Solution, (2010) "*Analisa Virus Tmphider atau Stuxnet [07-2010]*" [Online], Available: http://www.ihsana.com/?page=vblog&id=16 [Accessed 20 Feb 11]

Kaspersky, (2010) "*Unravelling Stuxnet*" [Online], Available: http://www.kaspersky.com/downloads/press/aleks_gostev,_costin_g._raiu._unravelling_stuxnet.zip [Accessed 20 Feb 11]

KPMG, (2010) "*DATA LOSS BAROMETER - Insights into lost and stolen information / November 2010*" [Online] Available : http://www.datalossbarometer.com/docs/KPMG_Data_Loss_Barometer_-_Issue_3_-_November_2010.pdf [Accessed 20 Feb 11]

Krebs on Security, (2010) "*Java: A Gift to Exploit Pack Makers*" [Online] Available: http://krebsonsecurity.com/2010/10/java-a-gift-to-exploit-pack-makers/ [Accessed 21 Feb 11]

Krutz, Ronald L, Vines, Russell Dean, (2010) "*Cloud Security: A Comprehensive Guide to Secure Cloud Computing*" John Wiley & Sons

Larimer, Jon, (2011) *"USB Autorun attacks against Linux"* [Online], Available: http://blogs.iss.net/archive/papers/ShmooCon2011-USB_Autorun_attacks_against_Linux.pdf [Accessed 14 Feb 11]

Ligh, Michael Hale, Adair Steven, Adair, Hartstein, Blake and Richard, Matthew, (2010) "*Malware Analyst's Cookbook - Tools and Techniques for Fighting Malicious Code*" Wiley

Lumension Security, (2010) "*Lumension Device Control (formerly sanctuary)*" [Online} Available: http://www.lumension.com/device-control-software/usb-security-protection.aspx [Accessed 20 Feb 11]

MalwareIntelligence, (2009), "*Danmec Bot, Fast-Flux networks and recruitment of Zombies PCs*" [Online], Available: http://malwareint.blogspot.com/2009/01/danmec-bot-fast-flux-networks-and.html [Accessed 20 Feb 11]

MalwareIntelligence, (2010) "*Malware and Exploit Packs*" [Online] Available: http://malwareint.blogspot.com/ [Accessed 20 Feb 11]

Mansourov, Nikolai, Campara, Djenana, (2010) "*System Assurance: Beyond Detecting Vulnerabilities*" The MK/OMG Press

MarkMonitor Inc., (2008) "*Rock Phishing: The Threat and Recommended Countermeasures*" MarkMonitor White Papers

McAfee, (2010) "*Downloader-CJX*" [Online], Available: http://vil.nai.com/vil/Content/v_268362.htm [Accessed 20 Feb 11]

Microsoft, (2011) "*Microsoft Security Bulletin MS08-067 – Critical*" [Online] Available: http://www.microsoft.com/technet/security/bulletin/ms08-067.mspx [Accessed 6 Jan 11]

Microsoft, (2011) "*Microsoft Security Bulletin MS10-046 - Critical*" [Online] Available: http://www.microsoft.com/technet/security/bulletin/ms10-046.mspx [Accessed 6 Jan 11]

Microsoft, (2011) "*Microsoft Security Bulletin MS10-061 - Critical*" [Online] Available: https://www.microsoft.com/technet/security/Bulletin/MS10-061.mspx [Accessed 6 Jan 11]

Microsoft, (2011) "*Microsoft Security Bulletin MS10-073 - Important*" [Online] Available: http://www.microsoft.com/technet/security/bulletin/ms10-073.mspx [Accessed 6 Jan 11]

Microsoft, (2011) "*Microsoft Security Bulletin MS10-092 - Important*" [Online] Available: http://www.microsoft.com/technet/security/bulletin/ms10-092.mspx [Accessed 6 Jan 11]

Microsoft, (2011) "*MS08-038: Vulnerability in Windows Explorer could allow remote code execution*" [Online] Available: http://support.microsoft.com/kb/950582/en-us [Accessed 6 Jan 11]

Microsoft, (2011), "*Microsoft Baseline Security Analyzer*" [Online] Available: http://technet.microsoft.com/en-us/security/cc184924 [Accessed 6 Jan 11]

Microsoft, (2010) "*Have you checked the Java?*" [Online] Available: http://blogs.technet.com/b/mmpc/archive/2010/10/18/have-you-checked-the-java.aspx [Accessed 21 Feb 11]

Microsoft, (2010) "*Worm:Win32/Conficker.B*" [Online], Available: http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?name=Worm%3aWin32%2fConficker.B [Accessed 20 Feb 11]

Microsoft, (2007) "*Step-By-Step Guide to Controlling Device Installation Using Group Policy*" [Online], Available: http://msdn.microsoft.com/en-us/library/bb530324.aspx [Accessed 14 Feb 11]

Microsoft, (2010) "*How to disable the Autorun functionality in Windows*" [Online], Available: http://support.microsoft.com/kb/967715 [Accessed 14 Feb 11]

Microsoft, (2011) "*Update to the AutoPlay functionality in Windows*" [Online], Available: http://support.microsoft.com/kb/971029 [Accessed 14 Feb 11]

Microsoft, (2011) "*To create a hash rule*" [Online], Available: http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/srp_hash.mspx?mfr=true [Accessed 14 Feb 11]

Microsoft, (2011) "*How To use Software Restriction Policies in Windows Server 2003*" [Online], Available: http://support.microsoft.com/kb/324036 [Accessed 14 Feb 11]

MWR Labs, (2009) "*USB Attacks: Fun with Plug and 0wn*" [Online], Available: http://labs.mwrinfosecurity.com/files/Publications/mwri_usb-attacks-defcon17_2009-08-02.pdf [Accessed 3 Dec 10]

National Institute of Standards and Technology (NIST), (2011) "*Guide to Security for Full Virtualisation Technologies*" [Online], Available: http://csrc.nist.gov/publications/nistpubs/800-125/SP800-125-final.pdf [Accessed 16 Feb 11]

National Security Agency, (2009) "*Security Configuration Guides*" [Online] Available: http://www.nsa.gov/ia/guidance/security_configuration_guides/index.shtml [Accessed 20 Feb 11]

NIST, (2011) "*Guide to Industrial Control Systems (ICS) Security*" [Online] Available: http://csrc.nist.gov/publications/drafts/800-82/draft_sp800-82-fpd.pdf [Accessed 20 Feb 11]

NIST, (2011) "*National Checklist Program Repository*" [Online] Available: http://web.nvd.nist.gov/view/ncp/repository?cid=1 [Accessed 20 Feb 11]

Noonan, Thomas, Archuleta, Edmund, (2008) "*The Insider Threat to Critical Infrastructures Study*" [Online], Available: http://www.dhs.gov/xlibrary/assets/niac/niac_insider_threat_to_critical_infrastructures_study.pdf [Accessed 16 Feb 11]

Orrey, (2010) "*Default Passwords*" [Online] Available: http://www.vulnerabilityassessment.co.uk/passwords.htm [Accessed 21 Feb 11]

Orrey, (2011) "*Cyber Attack: Exploiting the User - There are so many ways!*" [Online], Available: http://www.vulnerabilityassessment.co.uk/education/Thesis.pdf [Accessed 14 Feb 11]

Parker, Tom, (2011) "*Stuxnet Redux: Malware Attribution & Lessons Learned*" [Online], Available: https://media.blackhat.com/bh-dc-11/Parker/BlackHat_DC_2011_Parker_Finger%20Pointing-Slides.pdf [Accessed 14 Feb 11]

Phenoelit, (2010) "*Default Password List*" [Online] Available: http://www.phenoelit-us.org/dpl/dpl.html [Accessed 21 Feb 11]

Pisani, Jason, Caruga, Paul, Rushing, Richard, (2010) "USB –HID Hacker Interface Design" [Online], Available: https://media.blackhat.com/bh-us-10/presentations/Rushing/BlackHat-USA-2010-Rushing-USB-HID-slides.pdf [Accessed 17 Feb 11]

PJRC, (2011) "*Teensy USB Development Board"* [Online], Available: http://www.pjrc.com/teensy/ [Accessed 17 Feb 11]

PrincipledTechnologies, (2007) "*Options for reducing Intrusion Security Risks*" [Online], Available: http://www.principledtechnologies.com/clients/reports/Intel/ThinSecurity.pdf [Accessed 16 Feb 11]

RationallyPARANOID.com, (2010) "*List of commercial products that included malware*" [Online], Available: http://rationallyparanoid.com/articles/malware-in-commercial-products-list.html [Accessed 17 Feb 11]

Salomon, David, (2010) "*Elements of Computer Security*" Springer

SANS, (2007) "*SANS Laboratory – Defense in Depth Series*" [Online] Available: http://www.sans.edu/resources/securitylab/321.php [Accessed 20 Feb 11]

SANS, (2009) "*USB - Ubiquitous Security Backdoor*" [Online], Available: http://www.sans.org/reading_room/whitepapers/threats/usb-ubiquitous-security-backdoor_33173 [Accessed 17 Feb 11]

Secuobs, (2011) "*USBDumper*" [Online] Available: http://www.secuobs.com/USBDumper.rar [Accessed 20 Feb 11]

Secunia ApS, (2010) "*Secunia Personal Software Inspector (PSI)*" [Online] Available: http://secunia.com/vulnerability_scanning/personal/ [Accessed 20 Feb 11]

Secunia, (2010) "*Secunia Half Year Report 2010*" [Online] Available: http://secunia.com/gfx/pdf/Secunia_Half_Year_Report_2010.pdf [Accessed 21 Feb 11]

SCMagazine, (2010) "*Ten years of evolving threats: A look back at the impact of notable malicious wares of the past decade*" [Online], Available: http://www.scmagazineus.com/ten-years-of-evolving-threats-a-look-back-at-the-impact-of-notable-malicious-wares-of-the-past-decade/article/190864/ [20 Feb 11]

SecManiac, (2010) "*Hacking your perimeter - Not everyone needs to use zero days…*" [Online], Available: http://www.secmaniac.com/files/Hacking%20the%20perimeter.pdf [Accessed 17 Feb 11]

SecManiac, (2010) "*The Long Tail of Information Security*" [Online], Available: http://www.secmaniac.com/files/The_Long_Tail_of_Information_Security.pdf [Accessed 17 Feb 11]

Shiller, Jon, (2010) "*Cyber Attacks and Protection: Civilization Depends on Internet and Email*" CreateSpace

Sophos, (2011) "*Hardware keyloggers discovered at public libraries*" [Online], Available: http://nakedsecurity.sophos.com/2011/02/14/hardware-keyloggers-discovered-public-libraries/ [Accessed 17 Feb 11]

SpamHaus Project Ltd. (2011) "*What is "fast flux" hosting?"* [Online], Available: http://www.spamhaus.org/faq/answers.lasso?section=ISP%20Spam%20Issues#164 [Accessed 20 Feb11]

SPI Dynamics, (2005) "*Plug and Root: The USB Key to the Kingdom*" [Online], Available: http://www.blackhat.com/presentations/bh-usa-05/BH_US_05-Barrall-Dewey.pdf [Accessed 3 Dec 10]

Symantec Corporation, (2010) "*Stuxnet P2P component*" [Online], Available: http://www.symantec.com/connect/blogs/stuxnet-p2p-component [Accessed 20 Feb 11]

Symantec Corporation, (2011) "*W32.Stuxnet Dossier 1.4*" [Online], Available: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf [Accessed 14 Feb 11]

Symantec Corporation, (2011) "*Updated W32.Stuxnet Dossier is Available*" [Online], Available: http://www.symantec.com/connect/blogs/updated-w32stuxnet-dossier-available [Accessed 14 Feb 11]

The MITRE Corporation, (2011) "*CVE*" [Online] Available: http://cve.mitre.org/ [Accessed 20 Feb 11]

The MITRE Corporation, (2011) "*CVE-2010-2772 (under review)*" [Online] Available: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2772 [Accessed 20 Feb 11]

TheRegister, (2007) "*Fast flux foils botnet takedown*" [Online], Available: http://www.theregister.co.uk/2007/07/11/fast_flux_botnet/ [Accessed 20 Feb 11]

Trend Micro Incorporated, (2010), "*Stuxnet Malware Targeting SCADA Systems*" [Online] Available: http://threatinfo.trendmicro.com/vinfo/web_attacks/Stuxnet%20Malware%20Targeting%20SCADA%20Systems.html [Accessed 20 Feb 11]

Trend Micro Incorporated, (2010), "*STUXNET: Old Tricks, New Exploit*" [Online] Available: http://threatinfo.trendmicro.com/vinfo/articles/securityarticles.asp?xmlfile=091410-STUXNET.xml [Accessed 20 Feb 11]

US-CERT, (2008) "*Vulnerability Note VU#889747 - Microsoft Windows fails to properly handle the NoDriveTypeAutoRun registry value*" [Online], Available: http://www.kb.cert.org/vuls/id/889747 [Accessed 20 Feb 11]

Vacca, Jon, (2010) "*Managing Information Security*" Syngress

Vassilev, Apostle, (2007) "*Security Benefits from OS Virtualization: Real or Virtual?*" [Online], Available: http://netidsys.com/blog/papers/VMSecurity.pdf [Accessed 16 Feb 11]
Virus Bulletin Ltd, (2011), "*VB RAP test results*" [Online] Available: http://www.virusbtn.com/vb100/rap-index.xml [Accessed 20 Feb 11]

Zdnet, (2008) "*Malware-infected USB drives distributed at security conference* " [Online], Available: http://www.zdnet.com/blog/security/malware-infected-usb-drives-distributed-at-security-conference/1173 [Accessed 17 Feb 11]