IBM Security Solutions
July 2010

IBM X-Force Threat Insight Quarterly

Contents

About the report

The IBM X-Force® Threat Insight Quarterly is designed to highlight some of the most significant threats and challenges facing security professionals today. This report is a product of IBM Managed Security Services and the IBM X-Force research and development team. Each issue focuses on specific challenges and provides a recap of the most significant recent online threats.

IBM Managed Security Services are designed to help an organization improve its information security, by outsourcing security operations or supplementing your existing security teams. The IBM protection on-demand platform helps deliver Managed Security Services and the expertise, knowledge and infrastructure an organization needs to secure its information assets from Internet attacks.

The X-Force team provides the foundation for a preemptive approach to Internet security. The X-Force team is one of the best-known commercial security research groups in the world. This group of security experts researches and evaluates vulnerabilities and security issues, develops assessment and countermeasure technology for IBM security products, and educates the public about emerging Internet threats.

We welcome your feedback. Questions or comments regarding the content of this report should be addressed to XFTAS@us.ibm.com.

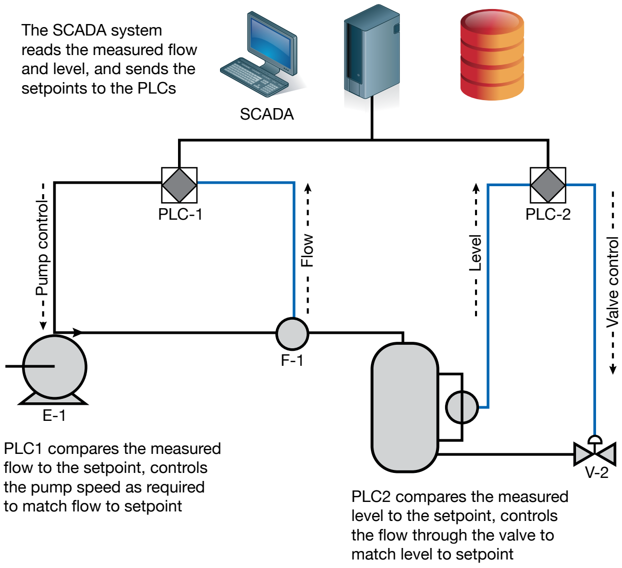
SCADA Systems – A Computer Security Nightmare?

By Jerome Radcliffe

When looking over the list of topics at computer security conferences, I have seen a trend of more talks on SCADA and SCADA related items. What is SCADA and why is it getting such attention from those in the security community? SCADA stands for Supervisory Control and Data Acquisition and it is used in a wide array of power plants, chemical processing facilities and other manufacturing environments. These SCADA systems allow operators to monitor and track measurements from a central location. This is often combined with a similar system used to remotely control operations that impact those measurements. For example the thermostat in a house acts like a SCADA sensor. It measures and records temperature and interacts with a furnace or air conditioner in order to regulate the temperature in the house. These systems have been in place for quite some time. In the last ten years these systems have become more networked and connected to the Internet. The movement to connect these systems is driven by the advantages that the Internet provides including inexpensive hardware, wireless connectivity, common protocols and intercompatability just to name a few. It also comes with the native problems inherent to the Internet, namely security problems.

SCADA systems in more detail

The implementation of SCADA systems across critical infrastructure is growing rapidly. Every industry faces pressure to cuts costs and maintain a lean headcount. In previous decades there were employees that would manually verify measurements on valves and gauges and then make decisions on adjustments that should be made to keep those conditions within a certain tolerance. Rather than have employees take those measurements and make those adjustments, the industry has moved to automated SCADA and Distributed Control System (DCS) to make it easier for a smaller team to manage conditions from a central location. These automated systems have additional advantages such as instant reading, the ability to record all measurements over time, and greater accuracy. In the case of a power plant, thousands of miles of transmissions lines and sub-stations can be maintained from a single central location. Figure 1 shows a simplified SCADA/DCS setup. Each of the Programmable Logic Controller (PLC) sensors continuously monitors pressure levels and adjusts valves to keep that pressure within a certain tolerance. This same setup could be applied to a multitude of different systems using other methods of control (switches and relays) and different measures of control (voltage, current, pressure, temperature).



The SCADA system reads the measured flow and level, and sends the setpoints to the PLCs

PLC1 compares the measured flow to the setpoint, controls the pump speed as required to match flow to setpoint

PLC2 compares the measured level to the setpoint, controls the flow through the valve to match level to setpoint

Figure 1: Diagram of a typical SCADA/DCS system

SCADA's Nightmare: Computer Security

The increased use of networked SCADA systems combined with the ever increasing threat from terrorism and cyberwar has prompted the industry to pay more attention to the security implications of SCADA installations. There are several plausible scenarios that give those in computer security a serious cause for concern.

1. **Denial of service attacks:** These types of attacks are designed to disrupt and prevent networks from communicating. These attacks can emanate from large scale botnets and are difficult to prevent or defend. If a SCADA network was flooded with traffic those sensors might not be able to transmit measurements back to a central location. Those measurements are key to making the proper adjustments to maintain service and to maintain the safety of the entire system. An example of this would be if an oil refinery or chemical plant that used SCADA and DCS in their plant were to be hit with a denial of service attack. Those sensors and controls that manage the levels of chemicals and heat would not function properly. This scenario would potentially lead to dangerous conditions which could result in wasted product, chemical leaks or explosions. In 2003 when the Slammer Worm hit a nuclear power plant was taken offline for over five hours due to Slammer causing a denial of service (Poulsen, 2003). Fortunately there were no injuries or damages associated with that incident.
2. **Data Injection and Fuzzing attacks:** These types of attacks are designed to intentionally insert false information into the data stream. An attacker might be able to masquerade as a legitimate sensor and send false readings back to the central control station. These types of attacks could be used to cause outages at utility companies, explosions at chemical plants, and denial of service for a variety of physical delivery systems such as water, gas, or electricity. The potential also exists for this type of attack to occur without detection until after the attack is complete. SCADA systems that utilize a wireless infrastructure are of unique concern especially in older installations where the use of encryption was not available.
3. **Logic Bombs and Backdoors:** One of the growing threat vectors is through the supply chain. Much of the equipment purchased in SCADA systems and in networking equipment comes from other countries and the review of how that equipment operates is minimal to non-existent. Some of the countries that produce the equipment have also developed cyber warfare capabilities as a nation-state through their military. In Richard Clarke's recent book *"Cyber War: The Next Threat to National Security"*, he writes that countries like China have the ability to embed logic bombs and secret backdoor access to devices that Chinese companies produce. This would give them an advantage if there was a cyberwar because they could use logic bombs and backdoors against countries that had purchased their devices to control critical infrastructure such as utility companies or military installations. In an example cited in the book the Russians were looking for technology to help advance their oil pipelines in the early 1980's. They attempted to purchase that technology from the United States but were denied. The Russians then opted to utilize the KGB for industrial espionage and stole the technology from a Canadian firm. What the Russians did not know was the CIA was working with that company and planted code that would cause malfunctions. Once installed the technology worked as designed but after several months things started to malfunction. One of those malfunctions led to an explosion measuring over three kilotons, the largest non-nuclear explosion recorded (Clarke, 2010).
4. **Internal Threats:** Often times the biggest threats come from within. Employees that have intimate knowledge of systems and how they are controlled are sometimes as dangerous as outside threats. In some cases the ability to change passwords in older SCADA systems is not possible so former employees would still have passwords to access these devices. In 2001 a disgruntled employee in Australia remotely opened some sewage lines and purposely caused millions of liters of raw sewage to be spilled out (Smith, 2001).

Recommended Actions

As national security concerns grow, so does the scrutiny on the critical infrastructures of nations. SCADA and DCS are critical working parts to that infrastructure and are the most vulnerable to attack. While there is no perfect solution there are steps that can be taken to minimize the threat while maintaining continuous affordable service. The one avenue to secure is around the authentication methods used in relation with the systems. Authentication in many SCADA systems is simply a single shared password that allows full access. This is not ideal as there is no audit trail to show which individual accessed the system. In many cases this password is rarely changed, not changed at all, or not changeable. An option to increase the authentication security is to implement a Virtual Private Network (VPN). This would require an individual to authenticate in order to access any equipment on the network. Additionally it would also provide an audit trail as to who was accessing the network and at what time. Implementing a VPN would be cost effective as it would not require the replacement of existing SCADA equipment, much of which is expensive and difficult to replace. In an incident previously mentioned in Australia, an ex-employee knew the password to access the system even after termination. A VPN would address that concern as terminated employees could easily be removed from VPN access whereas changing the passwords for all the SCADA equipment might not be feasible.

Another problem area is in the use of wireless infrastructure for the SCADA equipment to communicate. The use of wireless networking technology is cost effective in many scenarios where remote equipment is needed. In some cases like monitoring pipelines or equipment in rural unpopulated areas it might be the only option. If wireless infrastructure must be used every effort should be made to use equipment that supports some type of encryption, preferably WiFi Protected Access (Usually referred to as WPA or WPA2). Implementing wireless without encryption can result in data interception and opens up the possibility for data injection from a rouge host.

Another action that should be taken is to take an assessment of the state of security on the network SCADA systems reside on. This is usually referred to as a Penetration test (Pen Test). This type of test is conducted by professional security companies and provides details on where the weaknesses are in their computer security profile and what actions can be taken to minimize those risks. Quite often organizations are unaware of the weaknesses in their networks. That lack of knowledge makes it difficult to address what areas need to have their security bolstered.

SCADA systems deserve greater attention at security conferences. These are the types of systems that if compromised might impact millions of people with potentially deadly consequences. The water we drink, the electricity we use, and the natural gas that drives our industries are all controlled and monitored by these systems. Increased awareness on how these SCADA systems work and how they could be exploited is a serious matter and should remain a hot topic of discussion for quite some time.

Covert Channels and Virtual Private Networks

By Michael H. Warfield

Introduction

Many security tools have multiple mixed uses and may be utilized for either malicious purposes or for useful benign purposes. Some can be quite valuable. Some can be quite controversial. Few have more controversy attached to them than "covert channel tunnels" or "covert channel tools" or simply "covert channels". Most administrators view justifiably covert channels as being malicious software ranking right up there with trojans and backdoors (many of which support some form of covert channel tunneling).

The authors of these packages argue that these are legitimate packages with legitimate uses. And the debate rages on. Both sides are correct in as far as they go. Yes covert channels can be argued to have some limited legitimate use while they can also be argued to be primarily designed for malicious purposes. This debate will never be resolved. The real question is how much of a risk really exists from covert channels and how much effort is needed to protect our networks from them.

Recent releases and updates to some well known covert channel packages have once again raised awareness and concern over the risk or potential presence of these tools on a network. Over the last few months some DNS based covert channel tools such as "Iodine" have been updated. Articles have been published on using Open Virtual Private Network (OpenVPN), a popular OpenSource VPN package, as a DNS based covert channel tool. These have caused some concern for network administrators but probably more concern than what is really warranted.

Nature of Covert Channels and VPNs

Covert channel tunnels are fundamentally a form of Virtual Private Network or VPN. In this particular case it is a VPN designed to hide. It hides by appearing to be something else perfectly normal. In that regard it shares characteristics with rootkits and trojan horses in that part of its nature is to hide its presence. It is that "hide its presence" nature that generally marks it as malicious although some researchers may find legitimate uses for all of them.

Covert channels have been known about for a long time. The term "covert channel" was actually coined in 1972 even before the term "computer virus" entered our lexicon.

Covert channels are to networking, what steganography is to encryption. Both perform their function by "hiding in plain sight". You can see what is there but it appears to be legitimate. You have to look deeper to see what is hidden and even then may still miss it. But you have to know to look and it is this chicken and egg situation that makes covert channels difficult to detect.

Regardless of its nature to be hiding, a covert channel is still a VPN at heart. A virtual private network serves to make two points on a network congruent. That is, it makes those two points or interfaces on a network act as if they were adjacent. It does this by encapsulating network traffic in its own protocol and transporting it to the other side. Each end point sees the other endpoint as merely being "one hop" away. These are the fundamental characteristics of a VPN. You have two (or more) endpoints and you have some sort of encapsulation and transport. The "private" part of the VPN is the privacy of the endpoints. They are typically private to where they reside or terminate. It is "virtual" because it is assembled over another network through the encapsulation.

Contrary to common misconception VPNs are not required to be encrypted although most generally are. Some VPNs have encryption as part of their protocol specification and have no mode which is not encrypted. Other very common VPNs do have non-encrypted modes such as the NULL cipher for IPsec or disabling encryption in OpenVPN. Most covert channel tunnels are not encrypted. That does not mean they are any less of a VPN. It does mean that they depend on higher networking layers to provide encryption or confidentiality if desired.

Some but not all VPNs provide generic routing capability. That is they will map generic addresses on or beyond the endpoints into routes and addresses which can be routed through the endpoints. Without this the VPN is merely an application level VPN providing access to specific resources accessible between the end points. With generic routing the VPN will transport application independent traffic which may or may not terminate on either or both of the end points. Again, generic routing is not fundamental to the nature of a VPN but is a feature available with most VPNs. Some covert channel tools and tunnels provide for routing while others merely provide the end points and transport while depending on higher layers for generic routing if required at all.

A VPN provides encapsulation and transport between two or more end points and may provide encryption, authentication, and generic routing as options or may rely on higher layers for these optional features. A covert channel is simply a VPN having the additional feature that it is attempting to hide its presence in amongst more common mundane traffic.

Another class of tunnels shares many of these characteristics. Internet Protocol version 6 (IPv6) transitional tunnels also provide encapsulation and transport between two or more end points, generally provide generic (IPv6) routing and may include encryption and authentication. IPv6 tunnels are also a form of VPN which is transporting an entire orthogonal routing infrastructure above it. IPv6 is an outstanding routing infrastructure for covert channels exposing even private Internet Protocol version 4 (IPv4) networks to the global IPv6 infrastructure. It is no surprise that covert channels have been described utilizing IPv6. IPv6 has the added advantage that, even today, many network users and administrators remain ignorant of IPv6, in spite of its presence on almost every modern network. IPv6 provides an additional layer of obfuscation and evasion on top of the nature of the covert channel itself.

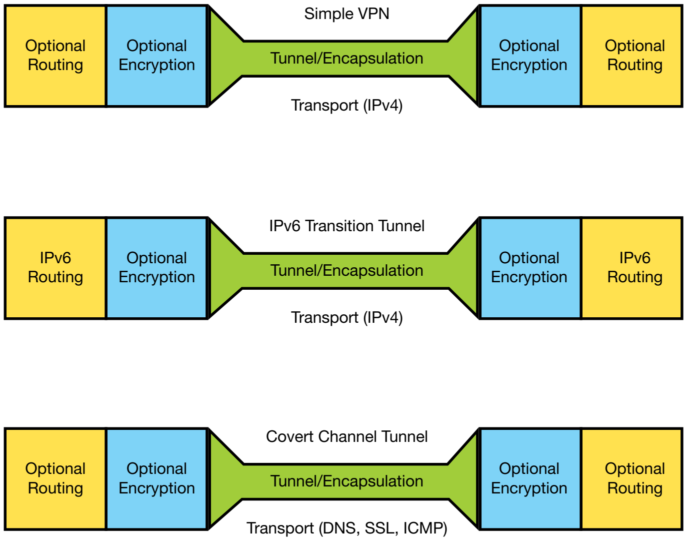


Figure 2: VPN, IPv6 and Covert Channel Tunnels

Transports and Encapsulations

VPNs and covert channels may exist at any of several layers in the network. Some VPNs such as IPSec, Layer 2 Tunneling Protocol (L2TP), and Generic Routing Encapsulation (GRE) operate directly as a layer on top of IP. Some IPv6 tunnels operate here as well.

A common transport for VPNs, covert channels, and IPv6 tunnels is UDP. OpenVPN, Teredo (IPv6 over UDP), and others utilize UDP as an efficient high performance transport which is blocked by very few firewalls for outgoing traffic. Additionally UDP supports a technique called STUN (Simple Translation of UDP over NAT) to allow independent devices, all behind Network Address Translation (NAT) devices or stateful firewalls, to communicate directly with each other needing only a STUN server to tell each other how to communicate with the others. The STUN server handles none of the data traffic and merely coordinates the initial establishment of the communications channel. Techniques have also been published for a STUN like protocol requiring no STUN server at all for mediating connections which utilizes ICMP error returns from non-existent addresses in place of the STUN servers. With STUN very few firewalls block UDP based tunnels.

Even IPSec, which nominally operates directly on IP, has a NAT Traversal (NAT-T) mode which transports IPSec over UDP port 4500. This enables IPSec to easily pass through NAT devices which do not properly support IPSec Passthrough. This all makes UDP a very popular transport for VPNs and tunnels of all sorts.

Some VPNs can operate over TCP as well although TCP is a poor transport for datagram oriented generic VPNs. Secure Shell (SSH) and Secure Sockets Layer (SSL) are both common encrypted transports which can be used for VPNs. OpenVPN also has a TCP mode as well. TCP based tunnels cannot provide the same level of performance, due to TCP overhead and management, as can IP or UDP based tunnels. However performance is often not a requirement. TCP tunnels, in particular HTTP and HTTPS, can operate on some networks where other protocols are strictly blocked. Those VPNs may even work through proxy servers. For some covert channels performance is not a consideration at all and even getting a few bits of data through per packet by tunneling in header flag fields or id fields is perfectly acceptable.

Because covert channels tunnels are attempting to hide in plain sight, it's advantageous to choose a common protocol to hide behind. The more likely it is to find on the network during normal operation the less likely it is to raise suspicion. The more the covert channel looks like legitimate traffic the easier it is to hide and the harder it is to spot. Consequently, DNS is a popular mechanism for a covert channel on UDP as are HTTP and HTTPS on TCP.

In addition to Iodine mentioned earlier, some known covert channel tools include:

- DNScat – Another tunnel over DNS
- ptunnel – Tunnel over ICMP
- htunnel – Tunnel over HTTP or HTTPS
- tcp tunnel – Tunnel over tcp flags
- id tunnel – Tunnel over IP id field
- stunnel – A common and popular legitimate tool for tunneling over SSL

Some of these tools are brought together in Gray World's "Covert Channel Tunneling Toolkit" or CCTT, in a single bundle of covert channel tools which was published a number of years ago.

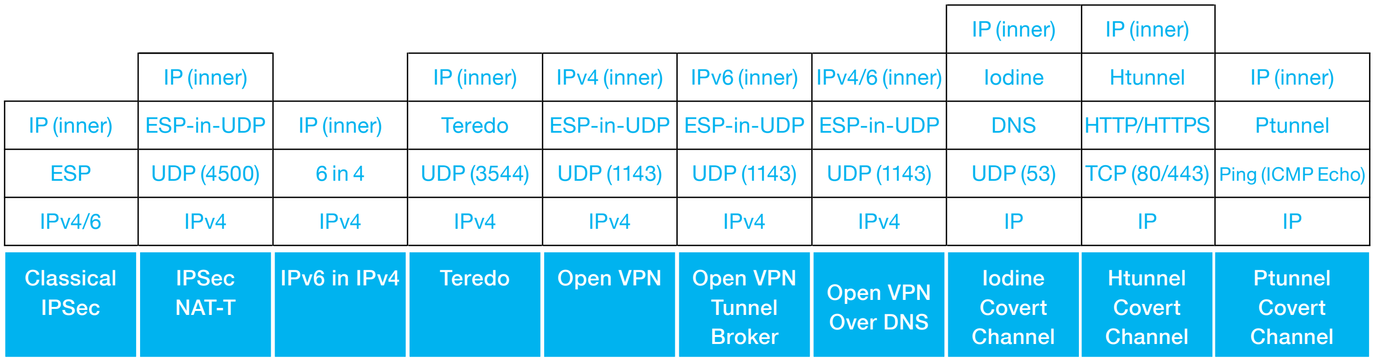
Some covert channels may merely piggy back on the well known ports utilized by the services they're hiding behind without actually emulating those services. If OpenVPN is used over port 53/UDP, it does not appear to be DNS traffic. But it requires packet inspection and discrimination to determine what is and is not legitimate DNS traffic. Furthermore, if DNS is generically blocked and forced to go through local resolvers OpenVPN cannot work with local DNS servers.

On the other hand, both DNScat and Iodine actually encapsulate the packets into the DNS high-level request response packets and send them through the full DNS infrastructure. This is less efficient but harder to detect. This also requires that the owner of the tunnels have control of the DNS servers for a legitimate zone but this isn't particularly difficult. Registering obscure zones to purely act as anchors for a covert channel is not expensive or difficult. An attacker can also take advantage of a compromised name server to insert a subdomain and delegate it back to another name server of their choice.

Unlike DNScat, Iodine can also detect when it can communicate directly back to an Iodine server and can drop into a "turbo" mode where it doesn't attempt to encapsulate the traffic in DNS packets. This provides better performance at the expense of becoming more detectable much like OpenVPN. Unlike OpenVPN, its traffic remains unencrypted and subject to deep packet inspection.

SSH, a very common high security shell and transport, can also be tunneled over HTTPS. Front end packages exist for SSL web servers that can detect the difference between an SSH connection attempt and an SSL connection attempt and route the request accordingly to the proper server. The SSH server thus hides behind the HTTPS/SSL port making it harder to detect and more accessible from sites which would otherwise block access to SSH but would allow secure HTTPS Web access. Deeper tunnels can be then transported over SSH using protocols such as Point-to-Point Protocol (PPP) or it can simply forward ports and connections for reverse shells and backdoors. These sorts of tunnels may be detected through deep packet inspection to discriminate true SSL traffic from non-SSL traffic over SSL ports.

The stunnel utility, on the other hand, uses a full SSL session for tunneling. Malicious tunnels using stunnel cannot be discriminated from normal benign SSL sessions versus legitimate SSL based VPNs on the basis of deep packet inspection. Many services on the Internet also include SSL encrypted versions as well as unencrypted versions.



							IP (inner)	IP (inner)	
	IP (inner)		IP (inner)	IPv4 (inner)	IPv6 (inner)	IPv4/6 (inner)	Iodine	Htunnel	IP (inner)
IP (inner)	ESP-in-UDP	IP (inner)	Teredo	ESP-in-UDP	ESP-in-UDP	ESP-in-UDP	DNS	HTTP/HTTPS	Ptunnel
ESP	UDP (4500)	6 in 4	UDP (3544)	UDP (1143)	UDP (1143)	UDP (1143)	UDP (53)	TCP (80/443)	Ping (ICMP Echo)
IPv4/6	IPv4	IPv4	IPv4	IPv4	IPv4	IPv4	IP	IP	IP
Classical IPSec	IPSec NAT-T	IPv6 in IPv4	Teredo	Open VPN	Open VPN Tunnel Broker	Open VPN Over DNS	Iodine Covert Channel	Htunnel Covert Channel	Ptunnel Covert Channel

Figure 3: *Comparison of several VPN encapsulations*

Beaconing

Beaconing is one fairly common use for a type of covert channel. A lot of malware and botnets incorporate beaconing to “check in” with the command and control systems rather than attempting to maintain persistent connections. A covert channel type beaconing over DNS is as simple as checking a particular domain name periodically. No special covert channel tools are required for this. If an error or “no operation” is received back the malware can just go back to sleep again for hours, days, or even weeks. If some sort of action response is returned the malware can then act on that through other channels to check in and transfer more complex data and commands. DNS can also respond to arbitrary TXT, or text, queries with fairly significant strings. DNS is already so heavily overloaded with a variety of resource records and queries as to make it difficult to sort out what is legitimate and what is not. Spotting this sort of covert channel hiding within the daily flurry of DNS requests is almost impossible. More often the malware is spotted through other means and reverse engineered to determine the DNS name of the beacon server.

Some forms of beaconing can be detected by data mining of DNS logs when these logs are produced and kept in a location set aside for this purpose. Certain types of traffic can immediately stand out as anomalous. Large numbers of query replies containing localhost (127.0.0.1) or private address space (10.0.0.0/8, 172.16.0.0/12, or 192.168.0.0/16) for queries of non-local domain names are certainly worth investigating as these would not normally be legitimate. Large numbers of queries for a broad spectrum of host names under a particular non-local domain name are equally anomalous. Large numbers of host names within a malicious domain are used to circumvent caching and time-to-live latency in caching name servers. Normally there would only be a small number of host names within a non-local domain name of interest to any particular client. Correlations between these are most certainly malicious. This requires that both DNS queries and the resulting responses be logged to a central logging site where data mining and correlations can take place independent of the normal production DNS activities.

OpenVPN and DNS Based Covert Channels.

OpenVPN is a good example of a perfectly legitimate utility which can be used not only for the “normal” VPNs for which it was designed but also serves as a powerful and flexible tool for providing IPv6 tunnel broker services, with or without encryption. On top of that, when residing on port 53/UDP, OpenVPN is also the core of a DNS covert channel which can be utilized to bypass many application level gateways such as those found in WiFi hotspots around the world. The same common popular package is capable of building all of these types of tunnels.

In the case of OpenVPN no attempt is made to emulate the DNS protocol itself and it can not propagate through caching name servers. By blocking port 53/UDP at the perimeter and forcing clients to use established caching name servers, this type of VPN bypass can be thwarted.

Other than beaconing, which a lot of malware and Advanced Persistant Threats (APTs) engage in, the most common use for DNS based covert channels such as OpenVPN on DNS, DNScat, or Iodine is for bypassing application layer gateways in WiFi hotspots such as those found in airports. A read of some of the “Changelog” for Iodine typifies this where the author has named releases such as “Hotspotify” or “WifiFree“ or “iPassed”. In this case the purpose is not to break into systems or maintain communications with compromised systems and networks. Rather the purpose is to bypass restrictions and avoid paying for services.

This is another case where DNS logging of both queries and response can produce forensically productive information. DNS based covert channels which actually emulate the DNS protocol can be distinguished in logs by excessively long and seemingly random host names in relatively short domain names in queries. The outbound information is encoded in these queries and stand out readily to cursory inspection of query logs. The responses also contain detectable information in the seemingly random encoded data of the response payloads. These types of covert channels are not very covert in the face of anomaly detection when DNS queries and responses are logged.

Covert Channels and Advanced Persistent Threats

A hot topic in recent months has been that of "Advanced Persistent Threats" or APTs. One of the characteristics of APTs has been very low level beaconing that is extremely difficult to detect. Components of the APT can use DNS based beaconing to periodically check in and establish a long latency heart beat to let the control point know it's still there and still alive. Should a component fail to check in, another component can then be notified through the same covert channel mechanism to take action to recover or reinfect. This very low level covert channel activity carries almost no significant amount of data but is a critical piece of the "persistent" part of APTs. As such it's merely using the established DNS system with no special tools outside of itself but communicating what it needs to communicate very effectively at very low bit rates.

There have been several instances of record where APTs were detected by companies as a result of huge amounts of data being transferred over port 53/UDP. Normally an APT will lay in wait watching for data of interest. Once it encounters that data it attempts to transfer that data back to a contact point. Using a covert channel for these sorts of bulk transfers can reveal both the covert channel and the APT due to the anomalously high traffic not normally associated with those services. In this way using a covert channel for large data transfers makes it more detectable through anomaly detection. This ultimately defeats the very purpose of a covert channel in the first place.

In a recent prominent case Google used their pre-existing logs to detect the existence of an APT on their networks. They were then able to roll back through those logs and trace the origin of the Aurora invasion to a "patient zero" and how that machine had become infected through a "spear phishing" attack. Rolling forward through the ongoing running logs they were able to detect new variants of the malware at the core of the infestation long before anti-virus was able to pick up those traces. Even though the volume of logged information must have been staggering for a network like Google, it was still possible to weed out the anomalous traffic to isolate and track its behavior. Without DNS logging this may well have been impossible. This is a case where an attempt to use a covert channel became the Achilles Heel of the APT leading to its detection and eradication.

Covert Channel Legitimate Uses

As much as some administrators wish it were not true, even "tools of ill repute" such as covert channel tools can have some legitimate use. These are generally of a scope where there is legitimate reason for using a side channel or out of band channel for some sort of security or controlling mechanism. This may be to make the control or monitoring channels transparent to other applications or may be to hide activities from potential intruders or protect services from intruders.

Port knocking is a prime example of this. Port knocking uses a sequence and/or timing on a series of ports to open up a service on another port. This "secret knock" technique is used on some servers to secure access to non-public services such as SSH. Without issuing the "secret knock", the SSH service never appears on the network. The knock does not have anything to do with the service itself and may not be on the same ports at all or even on the same address as the service. The "port knock" is out of band from the desired service and is designed to look like normal requests. However it changes the behavior of the target host upon receipt of a legitimate knock.

Other security software and research software such as honeypots and honeynets use hidden communications channels to monitor critical systems. The honeynet project tool Sebec is one such tool. Activities taking place in a potentially compromised honeypot are reported back to the monitoring stations using a hidden protocol. This covert channel is used to hide from the bad guys, in this case.

Other legitimate uses for covert channels may include "out of band signaling" and "heart beats" for high availability clusters or virtualization control systems.

Conclusion

A "covert channel tunnel" is more covert when it does not appear to be a covert channel. The instant a covert channel tunnel is discovered it can be assumed to be malicious (guilty until proven innocent) which is not what an attacker wants. An encrypted VPN can hide among the existing encrypted VPNs and appear no different and is difficult to prove malicious (innocent until proven guilty). This is what an attacker would prefer. In that regard, simple normal VPNs become more "covert" and more preferable than true "covert channel" VPNs. Thus standard VPNs and tunnels are more attractive to attackers and malware writers. As a result, the presence of unmanaged VPNs on a network is a far more serious threat than merely the prospect of covert channel tunnels.

On most networks some form of VPN is going to be allowed and even supported. If nothing else SSL based tunnels over HTTPS are almost certain to be permitted. Short of identifying malware, spotting intrusions, and tracking and blocking malicious sites, it's almost impossible to block these sorts of tunnels. Nothing in these tunnels indicates one may be malicious while another is perfectly routine. On such networks, true covert channel tunnels are unnecessary and may even be more at risk for discovery than standard VPN. The path of least resistance for attackers and malware writers is to do what is effective for the least amount of effort and the most difficult to detect. In most cases this is not through the use of a covert channel tunnel.

Worrying about covert channels while other, easier, communications channels exist is probably a wasted effort. It's not where the threat is and it is not likely to yield a lot of benefit in attempting to obstruct them. A more productive course of action is to gain and maintain control over some of the network resources these things exploit. Below are some suggestions on protecting these resources.

- VPNs should be managed as best as possible. While it may be impossible to completely detect and block all VPN activity, what can be managed should be managed and what can not be managed should be minimized. Depending on site policies this may mean blocking outbound well-known ports and protocols associated with the common VPN technologies with the understanding that most VPNs can be migrated to non-standard ports.
- DNS needs to be properly marshaled and managed. Wide open access to any DNS anywhere on the Internet from any machine within a network is an invitation for very simple covert channel tunnels.
- Designated caching name servers should be used and anomalous "non-DNS" traffic or requests to them should be investigated for the source.
- DNS queries and responses should be logged to a central location where it can be archived and where heuristic data mining can be applied to detect malicious activity or investigate past history of attacks and invasions.
- Stray, non-marshaled DNS requests should be trapped at the firewalls and logged.
- Detect and investigate any anomalously high traffic on services which are not normally associated with high traffic, such as DNS or NTP, particularly if the source or destination are unusual for the environment of the enterprise network.
- IPv6 should be recognized and understood and properly supported and managed. The time for ignoring IPv6 passed years ago. Like it or not, IPv6 is present on almost every modern network whether the administrators are aware of it or not. It can be taken advantage of by those who are better versed in it.
- Malicious covert channel tunnels are going to be associated with malware and intrusions. Ultimately it is the detection of the malware and intrusions that are the key.

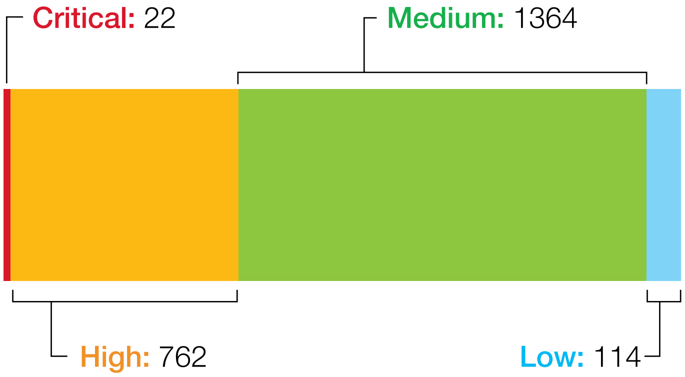
Ignorance is always dangerous but covert channels are not worth losing a lot of sleep over.

Prolific and Impacting Issues of Q2 2010

Significant disclosures

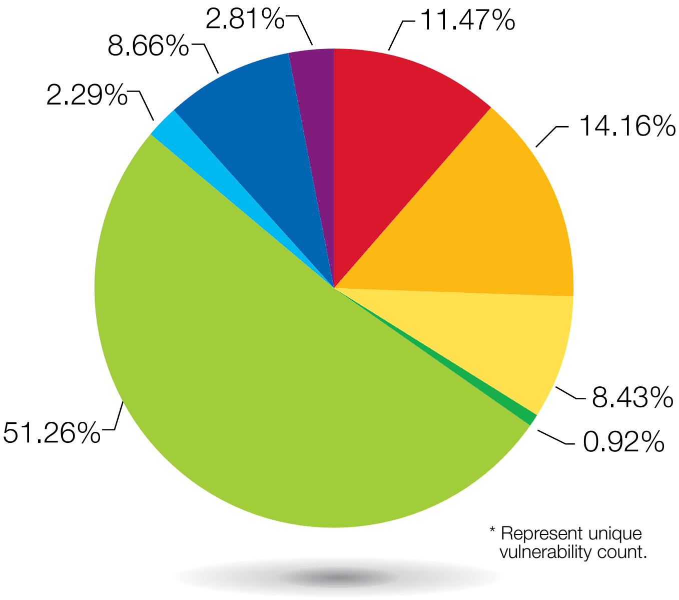
In Q2 2010, the X-Force team researched and assessed 2262 security related threats. A significant percentage of the vulnerabilities featured within the X-Force database became the focal point of malicious code writers whose productions included malware and targeted exploits.

Total Vulnerabilities in Q2 2010: 2262



Source: IBM X-Force

The chart below categorizes the vulnerabilities researched by X-Force team analysts according to what they believe would be the greatest categories of security consequences resulting from exploitation of the vulnerability. The categories are: Bypass Security, Data Manipulation, Denial of Service, File Manipulation, Gain Access, Gain Privileges, Obtain Information, and Other. *



* Represent unique vulnerability count.

Source: IBM X-Force

Bypass Security	Circumvent security restrictions such as a firewall or proxy, and IDS system or a virus scanner.
Data Manipulation	Manipulate data used or stored by the host associated with the service or application.
Denial of Service	Crash or disrupt a service or system to take down a network.
File Manipulation	Create, delete, read, modify, or overwrite files.
Gain Access	Obtain local and remote access. This also includes vulnerabilities by which an attacker can execute code or commands, because this usually allows the attacker to gain access to the system.
Gain Privileges	Privileges can be gained on the local system only.
Obtain Information	Obtain information such as file and path names, source code, passwords, or server configuration details.
Other	Anything not covered by the other categories.

The X-Force team published two Protection Alerts to address critical vulnerabilities disclosed in Microsoft's April 2010 Security Release. Microsoft® Windows® SMTP Service and Microsoft Exchange are vulnerable to a denial of service (DoS) caused by the improper handling of DNS Mail Exchanger (MX) resource records by the Simple Mail Transfer Protocol (SMTP) component. As SMTP services are often exposed to the Internet and email is usually considered a business critical function,the business impact of this vulnerability is more significant than for typical DoS issues.

- A protection alert provided by IBM: Denial of Service Conditions in Microsoft Exchange and Microsoft SMTP Service[1]
  - IBM Protection Signature: DNS_Windows_SMTP_MX_DoS
- CVE-2010-0024
- Microsoft Security Bulletin MS10-024: Vulnerabilities in Microsoft Exchange and Windows SMTP Service Could Allow Denial of Service (981832)[2]

Microsoft Windows is vulnerable to a stack-based buffer overflow caused by improper bounds checking by the MPEG Layer-3 audio codecs when handling malicious files. Successful exploitation of this issue would provide an attacker with complete control over the endpoint target. The use of malicious media files like images and movies has been prevalent in past years.

- A protection alert provided by IBM: Microsoft DirectShow Remote Code Execution[3]
  - IBM Protection Signature: AVI_DirectShow_MPEG3_Overflow
- CVE-2010-0480
- Microsoft Security Bulletin MS10-026: Vulnerability in Microsoft MPEG Layer-3 Codecs Could Allow Remote Code Execution (977816)[4]

[1] A protection alert provided by IBM: Denial of Service Conditions in Microsoft Exchange and Microsoft SMTP Service http://www.iss.net/threats/365.html

[2] Microsoft Security Bulletin MS10-024: Vulnerabilities in Microsoft Exchange and Windows SMTP Service Could Allow Denial of Service (981832) http://www.microsoft.com/technet/security/bulletin/ms10-024.mspx

[3] A protection alert provided by IBM: Microsoft DirectShow Remote Code Execution http://www.iss.net/threats/366.html

[4] Vulnerability in Microsoft MPEG Layer-3 Codecs Could Allow Remote Code Execution (977816) http://www.microsoft.com/technet/security/bulletin/ms10-026.mspx

On April 9, a proof of concept (PoC) exploit targeting the Java Deployment Toolkit was publicly disclosed. Less than a week later, reports began surfacing of attempts to exploit the then unpatched issue in the wild. Sun JRE could allow a remote attacker to execute arbitrary commands on the system caused by the improper validation of input by the launch method in the Java Deployment Toolkit ActiveX control and NPAPI plugin. By persuading a victim to visit a specially-crafted Web page, an attacker could pass arbitrary command line arguments to javaws to download and execute a malicious JAR file placed on a network share. Sun released Java 6 update 20 (1.6.0_20-b02) to address the vulnerability.

- A protection alert provided by IBM: Java Web Start[5]
  - IBM Protection Signatures: HTML_Java_Web_Start_Jailbreak, Script_Java_Web_Start_Jailbreak, HTML_Java_Web_Start_ActiveX
- CVE-2010-1423
- Sun Java SE 6 Update 20 Release Notes[6]

In April, the popular Zeus botnet began utilizing a vulnerability affecting Adobe® Systems' PDF format. PDF documents abusing the Launch feature can run arbitrary executables and the Zeus implementation drops the malicious binary with a deceiving PDF file extension for execution. Despite the spike in PDF exploitation in 2009, PDF is often considered to be safe and users are likely to be unaware of the potential for exploitation. However alternate PDF readers such as Foxit Reader include this feature without requiring user interaction. In cases where organizations have moved away from Adobe's implementation this is of particular concern.

- A protection alert provided by IBM: PDF-based Zeus attacks[7]
  - IBM Protection Signature: PDF_Launch_Program

While May remained relatively quiet on the cyber threat front, June proved to be a fairly active month. In early June, attackers began exploiting a zero-day vulnerability affecting Adobe Flash Player, Adobe Reader and Acrobat. Samples of the exploit soon were made public. This critical vulnerability exists in the Adobe Flash Player version 10.0.45.2 and earlier as well as in the authplay.dll component that ships with Adobe Reader and Acrobat 9.x. An attacker could exploit this issue to cause a vulnerable system to crash and potentially take control of the vulnerable system.

[5] Java Web Start http://www.iss.net/threats/367.html

[6] Java SE 6 Update 20 Release Notes http://java.sun.com/javase/6/webnotes/6u20.html

[7] A protection alert provided by IBM: PDF-based Zeus attacks http://www.iss.net/threats/PDFbasedZeusAttack.html

In addition to this exploitation, IBM X-Force received a report of a sophisticated attack occurring in the wild targeting this issue later in the month. This particular attack involves placing a specially-crafted Flash file within a PDF file. The IBM signature PDF_Swf_Detected detects this attack. As a conservative measure, customers may want to set this signature to blocking. While this change may also block legitimate traffic, this type of traffic (a Flash file embedded in a PDF file) is not commonly seen.

- A protection alert provided by IBM: Flash Player, Adobe Acrobat and Acrobat Reader Remote Code Execution[8]
  – IBM Protection Signatures:
  PDF versions: PDF_JavaScript_Exploit_JavaScript_Unescape_Obfuscation, PDF_JavaScript_Detected, PDF_Swf_Detected; SWF versions: Swf_Missing_
- ActionEndFlag, Swf_RealPlayer_Frame_Overflow
- CVE-2010-1297
- Adobe Security Advisory for Flash Player, Adobe Reader and Acrobat[9]

The X-Force team discovered one of the vulnerabilities disclosed in Microsoft's June 2010 Security Release. Microsoft Office applications fail to properly validate Component Object Model (COM) objects embedded in compound documents. This allows attackers to bypass the security settings of Office and embed known flawed objects in Office files. Upon exploitation of the pre-existing flaws in these controls, attackers can achieve arbitrary code execution.

- A protection advisory provided by IBM: Improper Validation of COM Objects in Microsoft Office[10]
  – IBM Protection Signature: CompoundFile_Shellcode_Detected
- CVE-2010-1263
- Microsoft Security Bulletin MS10-036: Vulnerability in COM Validation in Microsoft Office Could Allow Remote Code Execution (983235)[11]

[8] A protection alert provided by IBM: Flash Player, Adobe Acrobat and Acrobat Reader Remote Code Execution http://www.iss.net/threats/369.html

[9] Adobe Security Advisory for Flash Player, Adobe Reader and Acrobat http://www.adobe.com/support/security/advisories/apsa10-01.html

[10] A protection advisory provided by IBM: Improper Validation of COM Objects in Microsoft Office http://www.iss.net/threats/368.html

[11] Microsoft Security Bulletin MS10-036: Vulnerability in COM Validation in Microsoft Office Could Allow Remote Code Execution (983235) http://www.microsoft.com/technet/security/bulletin/ms10-036.mspx

On June 11, the threat level was elevated to AlertCon 2 to draw increased awareness to a zero-day Microsoft Windows Help Center Protocol Handler vulnerability. The remote code execution issue is trivial to exploit via a specially-crafted Web page. Multiple proof of concept exploits were made public at the time of disclosure and IBM analysts expected to see attacks in the wild. Microsoft later indicated that they were aware of limited targeted active attacks using the published proof-of-concept exploit code.

- A protection advisory provided by IBM:
  Improper Validation of COM Objects in Microsoft Office[12]
  - IBM Protection Signature:
    CompoundFile_Shellcode_Detected
- CVE-2010-1263
- Microsoft Security Bulletin MS10-036: Vulnerability in COM Validation in Microsoft Office Could Allow Remote Code Execution (983235)[13]

[12] A protection advisory provided by IBM: Improper Validation of COM Objects in Microsoft Office http://www.iss.net/threats/368.html

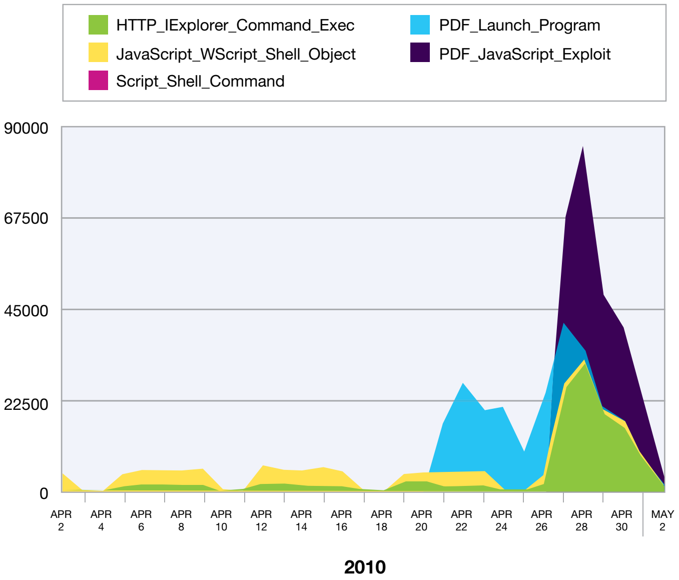
[13] Microsoft Security Bulletin MS10-036: Vulnerability in COM Validation in Microsoft Office Could Allow Remote Code Execution (983235) http://www.microsoft.com/technet/security/bulletin/ms10-036.mspx

Additional Q2 2010 Quarter highlights

This section of the report briefly covers some of the additional threats facing security professionals during Q2 2010.

Targeting Adobe

In late April, IBM Managed Security Services (MSS) observed a massive increase in malicious PDF spam attacks. These attacks continued to escalate over two days. Analysis showed that the spam attack itself was largely conventional but the malicious payload contained a novel exploit against the / Launch PDF command. When opened, this PDF file ran an executable file, game.exe, on the system being attacked. This executable appears to be a launcher for the Zeus botnet, while earlier reports were attributing the spam attack itself to the Pushdo botnet.



At the peak of the attacks, IBM MSS received 85,000+ alerts in a single day. If attackers were successful at a 10 percent rate of infection, that is easily 8,500 infections. This number does not take into account the amount of these attacks worldwide which could have been in the millions. This attack reinforces the fact that the Zeus botnet is a force to be reckoned with, will continue to evolve, and is not going away any time soon. While the attacks have subsided, IBM continues to encourage end-users to be vigilant.

SEO Attacks Continue

Search Engine Optimization (SEO) scams seemed to be a popular trend during the first quarter of 2010. This trend appears to have continued over into the second quarter with attackers targeting high-profile incidents and events including the eruption of Eyjafjallajokull and the World Cup.

SEO is a way in which to optimize a Web page to improve either the volume or the quality of traffic visiting the page so that it is ranked higher in the search results. This strategy has been around for a while and attackers have found a way to make SEO serve their own purposes. They modify or essentially poison the optimized search results of search engines to direct users to their malicious sites.

When it comes to SEO poisoning, anything can be a target. Attackers tend to jump on the bandwagon of what is popular in the media. This year there have been a number of high-profile incidents or tragedies for attackers to capitalize on. For instance, the earthquakes in Haiti and Chile drew major public interest. These incidents, unfortunately, also became a perfect way for attackers to launch SEO attacks. Searching on these terms shortly after the incidents not only led to legitimate sites offering information and ways to help but also malicious ones containing malware.

Attackers are not just targeting tragedy-themed terms. Maybe the latest and greatest gadget has come out from company X which is generating a lot of buzz or perhaps there is an international event such as the Olympics or the World Cup. The more popular the subject the more enticing it is to attackers because of the number of potential victims it could lead to.

In order to protect themselves against SEO attacks, users should be cautious when clicking on links from search results. Not all hosts that search engines point to are trustworthy. What appears to be a benign .edu or .com hostname could be a compromised site redirecting your browser to malware. If possible visit the official Web site directly. Maintaining up-to-date anti-virus software is key. Additionally some browsers allow blacklisting, such as the 'Block reported attack sites' setting in Firefox. Enabling this feature can also help mitigate against this type of threat.

List of Contributors for this paper include:

Jerome Radcliffe – Cyber Threat Intelligence Analyst
IBM MSS Intelligence Center

Michael H. Warfield – Senior Researcher, X-Force
IBM MSS Intelligence Center

Michelle Alvarez – Team Lead & Cyber Threat Intelligence Analyst
IBM MSS Intelligence Center

IBM X-Force Database Team

References

SCADA Systems - A Computer Security Nightmare?
http://commons.wikimedia.org/wiki/File:SCADA_schematic_overview-s.svg

Clarke, R. (2010). Cyber War: The Next Threat to National Security.

Poulsen, K. (2003, August 19). Retrieved June 24, 2010
http://www.securityfocus.com/news/6767

Smith, T. (2001, October 31). The Register. Retrieved June 24, 2010,
http://www.theregister.co.uk/2001/10/31/hacker_jailed_for_revenge_sewage/

Covert Channels and Virtual Private Networks
Covert Channels
http://en.wikipedia.org/wiki/Covert_channel

Gray World, Covert Channel Tunneling Toolkit
http://www.gray-world.net/

Iodine
http://code.kryo.se/iodine/

DNScat
http://tadek.pietraszek.org/projects/DNScat/

OpenVPN
http://www.openvpn.org

UDP Tunneling to avoid hotspot or firewall restrictions
http://www.adamsinfo.com/udp-tunneling-to-avoid-hotspot-or-firewall-restrictions/

Intrusion Detection FAQ: What is covert channel and what are some examples?
http://www.sans.org/security-resources/idfaq/covert_chan.php

Covert channel tool hides data in IPv6
http://www.securityfocus.com/news/11406

Sebek project site
https://projects.honeynet.org/sebek/

For Google, DNS log analysis essential in Aurora attack investigation
http://searchsecurity.techtarget.com/news/article/0,289142,sid14_gci1514965,00.html

X-Force Threat Insight Monthly, January 2008, “Toward a Robust Domain Naming System”
http://www-935.ibm.com/services/us/iss/pdf/x-force/xftim_0801.pdf

Prolific and Impacting Issues of Q2 2010
The Aftermath of doc.pdf, statistics, payload, and spam
http://blogs.iss.net/archive/aftermathofdocpdf.html

*Information in this document concerning non-IBM products was obtained from the suppliers of these products, published announcement material or other publicly available sources. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All performance data contained in this publication was obtained in the specific operating environment and under the conditions described above and is presented as an illustration. Performance obtained in other operating environments may vary and customers should conduct their own testing.

IBM Global Services
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
July 2010
All Rights Reserved

The customer is responsible for ensuring compliance with legal requirements. It is the customer's sole responsibility to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the reader may have to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law or regulation.

The use of third-party data, studies and/or quoted material does not represent an endorsement by IBM of the publishing organization, nor does it necessarily represent the viewpoint of IBM.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.

U.S. Patent No. 7,093,239