A PRACTICAL UNDERSTANDING OF MALWARE SECURITY

Greg Day
McAfee, Alton House, Gatehouse Way, Aylesbury
HP19 8YD, UK

Tel +44 1296 617008 • Email
Greg_Day@McAfee.com

ABSTRACT

In the past decade we have seen the time shrinking between initial discovery of an attack and widespread customer infection. Today, the term 'zero-day attack' has become a major topic of discussion for security teams, as it serves to both highlight a weakness in the traditional first line of defence against malicious code attacks; the signature update, while at the same time prompting re-evaluation of what is required to maintain an effective barrier.

With a plethora of security products available on the market, each offering their own unique value, we must each look to understand in what direction to evolve our security strategies. To achieve this we must understand how attacks function, what their objectives are, and how they will impact our businesses, both directly and indirectly.

The aim of this session will be to show, via demonstrations, the methodology used by today's attackers. We will then discuss and demonstrate alternative security solutions such as intrusion prevention systems (IPS), personal firewalls and behavioural tools to understand the value they bring as malware defence tools. Do they replace or complement our existing protection?

INTRODUCTION

Back in the mid-1980s when the need to protect against hostile attacks became commonplace you could count on one hand the number of incidents from which your organisation suffered annually. Indeed, the spread of each new attack was a newsworthy event.

Two decades later we have reached the point at which there are so many attacks that the average IT administrator no longer knows or cares how many there really are. The reality is we all have to deal on a daily basis with a broad spectrum of attacks that have the potential to seriously impact the revenue of any business.

In 2005 the average investment in IT is predicted to grow between 2.5% and 5.6%, depending on which analyst's report you read, with security being one of the top drivers. Every year there are new cutting-edge technologies being released into the security marketplace that claim to offer organisations a solution that fills the gap between current security practices and the holy grail of perfect security; each vying for this expanding investment in security. Businesses must navigate this minefield of solutions, determining what technology investment will serve their specific security needs.

Before we can gain an understanding of the value each new security solution offers we must be clear that we understand what we are looking to solve.

DEFINING THE BUSINESS PROBLEM

Traditionally the success of threat protection has been defined as preventing the attack from being successful; preventing the hacker from stealing data, the virus infection from occurring. Each of these carries a tangible cost to any business. Most commonly tracked for virus infections is the cleanup cost. However, in recent years the realisation has occurred that this is no longer a sufficiently exacting measure.

Although we may stop the attack occurring, in many of today's attacks businesses still suffer impact, which carries a financial cost. Let me explain by example:

The Sasser worm was one of the major network worms in 2004. It used a Buffer Overflow exploit to gain the required privileges to infect each system. Take the assumption that all your known systems were protected with anti-virus and had signatures in place to detect the infection. In theory, the level of infection and associated cost to the business would be zero.

Next, a rogue Sasser-infected PC comes onto the network. As it attempts to infect other protected systems you may claim success as each attempted infection is blocked by the anti-virus. However, all the systems that do not have the appropriate patch in place will suffer the buffer overflow security breach, which commonly results in the system rebooting. As Sasser attempts to make up to several hundred IP connections per second there is potential for large numbers of systems to suffer impact as the attempted infection causes the system to reboot. The end point is a loss of productivity as users are broken from their thought processes as the reboot occurs, and bandwidth usage from the attempted infections.

Bandwidth wastage is also an issue as mass mailers and network worms alike have the potential to flood network segments. This can result in indirect cost to business due to loss of connectivity.

From these experiences businesses have had to redefine the goal of attack protection as stopping their impact on our environments, rather than simply mitigating the cost of cleanup.

If the business challenge is to prevent or minimise the impact an attack has on our environment it is important to understand why our existing security technologies leave scope for business impact from attack.

TRADITIONAL SOLUTIONS

If we start with anti-virus, traditionally it has been based around the basic premise of signature matching, which means that for attack detection to occur it must have been analysed by the security vendor, resulting in a new signature being added to the detection database.

For many years this has been an effective strategy. Anti-virus vendors balanced the speed with which the signature sets were released against the speed with which attacks commonly spread in the wild. As the speed of attacks have increased so we have seen the schedule for releasing signature updates drop from months, to weeks, to days. Unfortunately the reality today is such that we have seen attacks spread around the globe in minutes (such as SQL/Slammer and CodeRed) and commonly in a few hours (such as many of the mass-mailers).

This has highlighted the need for more proactive protection strategies. As you can see from the graph shown in Figure 1, the ratio between getting the signature to the customer before an attack became common in the wild (i.e. existing detection was in place to stop the attack) and needing a new signature to stop the attack has been sliding towards the latter.

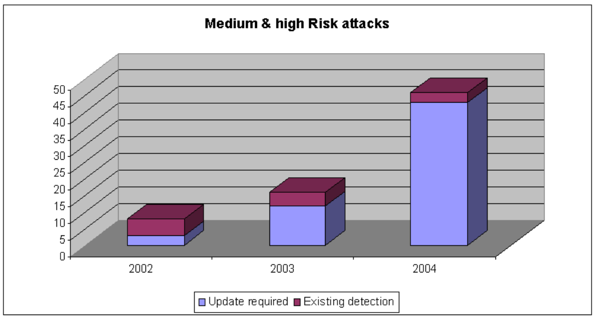

Figure 1: Medium and high risk attacks 2002–2004.

Detection by family (generic detection) and heuristic analysis have helped, but techniques such as packing (which allow you to reuse old attacks by repackaging the attack using a software tool not dissimilar to compression tools that let you create self-extracting files) mean that attackers can reuse old code in such a way that you need another signature in a matter of minutes.

This does not mean the end of anti-virus however. It provides a unique value in terms of its ability to give an absolute definition of the problem and an ability to undo the damage when a system has become infected. To achieve this you need to know exactly what you have been attacked by in order to undo the changes it has made. This is something behavioural tools can not achieve, more of which later.

Patching is a phenomenon that has been with us for decades in terms of servers and critical resources, yet only in the last few years has it become recognised as a fundamental part of client security protection. It falls foul of the same challenges as anti-virus. That is, it can only solve a problem that is known, i.e. it's only once a vendor knows about a software error, such as a security weakness, that they can produce a solution; a patch or service pack.

The commonality with the above solutions is their reactionary nature. When a new attack starts the assumption is made that customers can deploy whatever fix is required at a suitable speed. There are numerous tools, both vendor-specific and neutral, that focus purely on getting updates deployed at speed; the better solutions aiming to offer broad deployment within an hour. However, in most situations businesses will want to test solutions prior to broad scale rollout to ensure that they are not creating new problems for themselves.

Solutions based around reacting to the problem have in part led to a term that is all too commonly known amongst today's security vendors – the 'zero-day attack'. Put simply, the zero-day attack is the instance in which there was zero time to act against the problem. Depending on who you are talking to, that may be based on your ability to react or the earlier point, which is the vendors' ability to react; this can be a security vendor with a solution or a software vendor with a patch.

In either instance this has been one of the key stimuli that has driven both security vendors and customers alike to look for alternative security solutions that do not rely on a reactionary approach to security threats and attacks.

A DIFFERENT APPROACH TO SOLVING THE PROBLEM

It seems today that there is now a myriad of security solutions that all seem to solve the same problem. Take for example (again) a network worm such as Sasser. This is a type of attack that anti-virus, firewall, network and host intrusion prevention, patching and numerous other technologies all claim they could play a part in preventing.

When looking for new methods of attack protection there is a very logical complement to the reactionary technologies already discussed. If protection has traditionally been based around solving the known bad problem, the reverse would be to define good practice and then try to enforce this, so only it can occur.

There are two levels at which this can be achieved.

- Good working practice – based around the premise of users need to achieve with their systems only what is needed for their business role. Anything else is a waste of that business resource and an opportunity for attack.
- Good security practice within the IT infrastructure – within the operating systems, applications and networking used, enforce good security standards based on the industry RFCs.

GOOD WORKING PRACTICE

This concept has been around for a number of years, but has not been adopted across the industry in the same way as signature detection. Why?

Desktop firewalls are a prime example. Any customer with a connection to the Internet will have a firewall in place to ensure that only genuine connections are made through required ports. However, when you take that philosophy to the desktop you hit a challenge. Traditionally desktops have been heterogeneous environments. Whilst it's easy to define a single set of rules for a single point of entry, it can be hugely challenging to do the same to hundreds or thousands of autonomous systems.

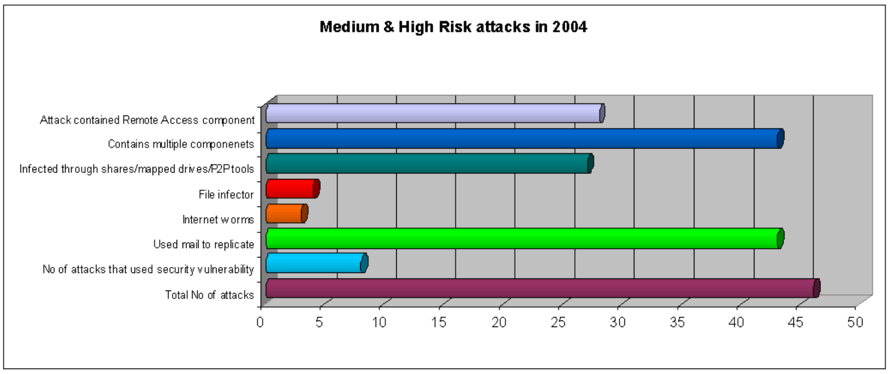

Figure 2: The common behavioural traits exhibited in the medium and high risk attacks of 2004.

Today we are moving towards standardisation in our IT working environment for a number of motives; most simply the need to simplify support and reduce costs. Even with this, to enforce good working practices on our systems is a huge challenge, as the scope for use and technical options available are massive. Again, configuring a desktop firewall is a prime example; with over 65,000 ports that can be used by thousands of applications and processes, it seems like a gargantuan feat to define which ports are required by which processes, so that you can the block all the others.

What is required is to take this to the next evolutionary phase. That is not to try to define perfect working security across the board, but to focus in on the specific areas of behavioural control that are relevant to stopping the behavioural traits and methods commonly exhibited by attacks.

In Figure 2 you can see the common behavioural traits exhibited in the medium and high risk attacks of 2004. The mass-mailer, for example, is the most common method of attack, however P2P replication and backdoor behaviour are common traits in the majority of the medium and high risk attacks as well. These become areas upon which we can focus with behavioural controls as there is direct and obvious value. If you can separate this behaviour from genuine email, P2P and remote access tools, then you have a valuable generic solution to the attack problem.

These are not short-term fixes either; as you can see from Figure 3 these traits have been commonly used by attackers over the last few years. (Note that the volumes in 2004 were much higher than 2002 and 2003, due to the re-cycling of attack code with packing tools.)

With a focus on common attack behaviour we can look as to which security solutions can best segregate the attack behaviour from genuine business use. Technologies such as personal firewalls, host Intrusion Protection Systems (IPS) and, more simplistically, OS security policies allow granular control of each system. For most we are not looking to lock down the system to the nth degree, we simply want to separate working practice from the behaviour exhibited by attacks.

A couple of simple examples help clarify this.

The most common form of infection we see today is a worm, whether it is network or email-based. In both instances the worm will commonly write itself to the Windows system folder and modify the registry 'run=' command to ensure the attack becomes resident each time the system is booted. The logic for adding itself to the Windows system folder is simple; it's a programmatic variable. This makes writing the attack simpler, the attacker doesn't have to be concerned about where the OS is installed to or even which *Windows* OS you are running – this is all handled by the OS.

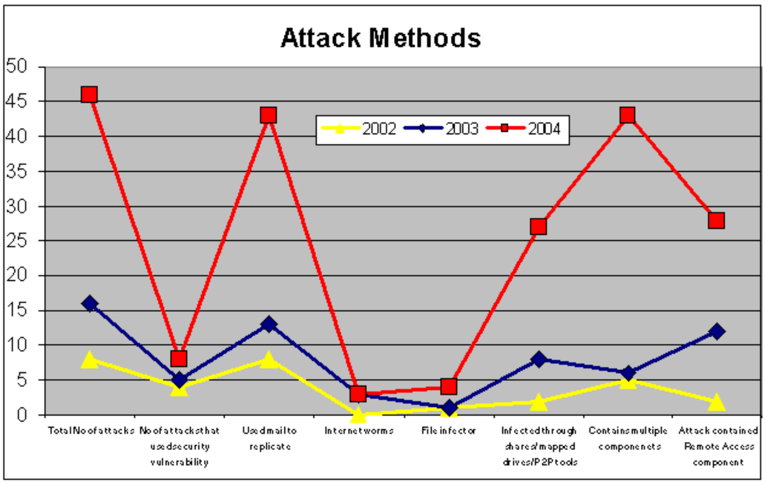

Figure 3: Attack methods 2002–2004.

Also ask yourself this; do you know how many files are in your own Windows system folder? If not, how would you spot a new file added?

Finally, have a look at your own Windows system folder (commonly WINNT\SYSTEM32) and check the last time an executable file (.EXE) was created (not modified) in that folder. Executable files should be written to the Program files folder unless they are part of the OS. This is good programming practice.

Taking these common traits it is simple to create behavioural rules that block changes to the registry and new .EXE files (executables) being written to the Windows system folder. Depending on the level of granular control offered by the security product, you may simply block all access, which can lead to false positives and the need to disable or remove the rule at times such as applying OS service packs. With more in-depth behaviour control tools you can set very granular controls, such as defining which users or processes can make these system changes.

A second example of this level of control would be the common method of infection that is to write temptingly named executables '.EXEs' to folders whose names contain the string 'Shar' or 'Sharing', which are commonly used as default by many of the P2P sharing tools.

Corporate customers generally do not view P2P sharing tools as valuable corporate tools and so discourage their use at a policy level. However this can enforced with a simple behavioural rule that would block executable files from being created in folders containing the phrases outlined above. Where users can justify this need, exceptions can be created. This rule would add huge value both in blocking P2P tools being used by attacks and also highlighting where users have installed P2P sharing tools if the scope of blocking were broadened.

GOOD SECURITY PRACTICE

When looking at how attacks gain permissions to write themselves to your environments there are a number of common tactics; tricking the user, exploiting poor security configuration in your environment and exploiting software vulnerabilities. The latter is a technique that comes from the hacking fraternity, and is a method that is increasingly being used by virus authors, both as a method of bypassing the user's involvement with mass-mailers and allowing the rapid proliferation of network worms.

The logic of good security practice is simple; if you don't allow the attacker to gain permissions to your environment then the attack can not succeed. However, with the growth in volume and frequency of security vulnerabilities being discovered, the timescales involved in waiting for a patch, testing and deployment mean that security exploitation has become a major pitfall that requires proactive coverage.

The simplest solution would be to ensure that all systems and infrastructure devices are perfectly patched all the time, but there are challenges to achieving this. First is the age old issue of compliancy. When I have more than a handful of nodes it becomes a challenge to maintain such a level of coverage. The second is the more visible issue of the availability of a patch to fix a discovered vulnerability. This is the attacker's Holy Grail; when there is a vulnerability, in a common application or OS, for which, at the point of releasing the attack, there is no security patch, the attacker knows they have a high probability of success. This, again, is what you'll commonly hear as the zero-day attack.

Products such as Intrusion Protection Systems (IPS) aim to combat such security breach-based attacks by adding a second layer of security protection. It's a simple approach that most of us use in everyday life: two locks are better that one.

There are two main methodologies to enforcing good security practice. The first takes the same approach as anti-virus, which is a database of known security vulnerability signatures. As each new exploit is discovered so a signature is written. Its success it based on the premise that the security exploit has to be known about for an attack to be written based on it.

The value of this form of detection is that often a security signature can be in place as soon as the security exploit is discovered, in many instances this is before any attack is written based on the security exploit. As such, it provides an earlier level of protection over an anti-virus signature, assuming the attack uses a security exploit. This style of detection is generally very accurate as it is based on a pattern match of those security exploits that are known about. Its weakness is that it does not solve the issue of the 'zero-day' exploit.

For this the second methodology is required, which is the security behaviour analysis. In just the same way that we define good working practices we can define good security practices. There are RFCs and numerous papers that define how our networks and systems should function. By creating rules that enforce these defined security practices we can monitor for anything outside the scope of the defined good security practice.

An example of this form of defence would be Buffer Overflow (a common technique used to gain security permissions) behavioural monitoring. By monitoring the requests made to the kernel it is possible to detect the specific behaviour that occurs only during a buffer overflow. In such an instance you do not actually stop the security breach itself (the buffer overflow), but stop the attack that would occur as a result of the breach. The value in such protection is its ability to stop a very broad spectrum of attacks with a generic monitor. As it is looking for a security breach method, it does not care what attack may follow the breach, only that the breach has occurred.

At a system level, such protection may still allow for some impact to the business. A security breach such as a buffer overflow can result in system instabilities as the process attacked has effectively been damaged by the attack. In such instances it isn't uncommon for the application or operating system to detect the broken process and attempt to resolve it. In *Windows* OSs this may be an error message or prompt to restart the system.

Network Intrusion Protection is better at reducing the impact of such attacks. By examining network packets on the wire, any detected security attacks can be discarded before the attack reaches the intended host. In such instances no impact occurs on the client, and you could argue bandwidth is saved depending on where the network IPS sensor was installed.

However, its strength is also its weakness. If an attack does not pass through a sensor then it will not be detected i.e. if every byte of traffic is not routed through an IPS sensor. If I have protection on the client system, it does not matter how the attack reaches the system, it will be analysed as it executes.

With both approaches (Network and Host) to IPS behavioural security protection there is a reality that should be considered. Not all software truly follows the rules defined in the RFCs for security, and as such behavioural protection can have a higher possibility for false alerts compared to scanning using database of known security exploit signatures. A good IPS product should give guidance on the confidence in which each behavioural control can be used and give an easy method of excluding false positive detections.

Unlike the signature matching, which is considered a 'set-it-and-forget-it' solution, behavioural matching is considered more a path to protection.

REACTIVE CONTROLS (FORENSICS)

An aspect that is often overlooked when any attack occurs is the concept of using the knowledge you have gained against it. Security vendors have become very proficient in giving detailed descriptions about each attack as soon as they are discovered and analysed.

Whilst signature updates are becoming smaller in size, it can often still take some time to be able to deploy them across the business. This can be due to connectivity, testing cycles required and other processes.

Ideally the best solution is to prevent attacks with proactive behavioural control tools, whether based on good working practice or security enforcement. However, such methodologies are not implemented overnight; as discussed earlier they are a path to better security protection.

Where you are still in the cycle of implementing these solutions, it is worth remembering that as any new attack becomes public knowledge these same behavioural techniques can be used reactively as a quick and effective method to block a specific attack.

Each of the medium and above risk attacks for 2004 had a unique characteristic that could be used to identify the infection. Most commonly a registry key or file added to the infected host. With a good behavioural enforcement tool you can quickly create a new rule or policy that would implicitly block that unique change made by the attack, so preventing the infection. This is a good half-way point when you are not yet to a point of being able to block such attackers at a more generic level.

Take, for example, MyDoom.O@MM – a variant of Mydoom that hit in 2004: this attack added java.exe and services.exe to

	2004 Medium+ risk attacks affected (Total for year - 46)
IntruShield (Network IPS)	
(Signature detection) SMTP: Worm Detected in Attachment Looks for attachments with an EXE, PIF or SCR extension. *Note: in some instances this would not completely contain the infection, as the mailer may use additional attachment types*	39
(Signature detection) P2P: KaZaA, Gntella, Gnucleus, Morpheus, BearShare, LimeWire, Grokster, Phex, Xolox, eDonkey, WinMX & Swapper File Transferring.	24
(Signature detection) SMTP: Possible Virus Attachment File with Double Extension	7
(Signature detection) NETBIOS-SS: Copy Executable File Attempt, when copying itself to a remote file share (if the file is an executable).	4
(Signature Detection) DCERPC: Microsoft Windows LSASS Buffer Overflow	3
(Signature detection) SMTP: Incorrect MIME Header with Executable Attachment Found	2
Entercept (Host IPS)	
(Shield Signature) System Executable Creation or deletion- detects New EXE's to Windows system folder.	35
(Shield Signature) New Startup folder program creation - detects registry write	45
(Custom Rule) Block new EXE's being created in the Windows folder	12
(Custom Rule) Block new EXE writes to folders with "Shar" or "Sharing". Use block sting "**shar**.exe" & "**sharing**.exe"	18
(Signature): Generic Buffer Overflow detection	5
VS8.0i (Access Control rules)	
(Network Access control - Port Rule) Block mass mailing worms from sending mail	41
(System Access control - File/folder protection) - Prevent the creation of new files the System32 folder (.EXE)	34
(Access control - File/folder protection) - Prevent the creation of new files the System32 folder (.DLL)	4
(Buffer Overflow protection)	5

Figure 4: Examples of some of the logical first steps that can be taken with McAfee*'s proactive products.*

the Windows folder, modified the HKLM 'Run=' command and opened up port 1034 as a backdoor to the system. Any of these abnormal behavioural traits could be converted into rules that would allow monitoring of infected systems, and the two former could be used to block the infection.

PROTECTION STRATEGY SUMMARY

If you consider all of the above, it starts to become clear as to why there are so many different solutions that all claim to solve the same attacks, even though they achieve it through a number of different mechanisms.

The challenge for any business is to understand what each of these technologies offers, analyse their existing protection tools and outline the gaps in your strategy.

For most the motivation is to move from a state where we are reacting to each attack as it occurs to a more proactive solution. This is not an overnight process; however there can be some quick wins with only a few basic behavioural controls.

The table in Figure 4 gives examples of some of the logical first steps that can be taken with *McAfee*'s proactive products: *IntruShield* (Network IPS), *Entercept* (Host IPS) and the proactive behavioural controls included in *VirusScan 8.0i*.

This is not a comprehensive list of all the proactive rules and signatures that could be used, but an indicator of the common behavioural controls that would give the quick wins, as highlighted by the volume of attacks that would have been blocked or contained if they had been in place. The data in the table is based on the 46 medium and above risked attacks in 2004 based on *McAfee*'s risk rating.

When you look at proactive security solutions it is important to understand what level of control they give you. All of the solutions discussed above work on the premise that you should block only that which you know (through either behaviour or signature) to be bad, which, in turn, is based on either practical experience or it being outside the definitions of what is good security or working practice. Some solutions offer far more granular advanced levels of enforcement and control than others.

Whichever solution suits your needs you should aim to gain the maximum from the technology by using at its different levels:

- Prevention – generic blocking by enforcing good security and business practices across your systems and infrastructure.
- Forensics – when you are not ready to prevent, as you cannot yet separate at a sufficiently granular level attack behaviour from normal practice, or do not yet have the confidence in your policies to enforce. Warning level modes allow you to log, which both helps gain confidence and also can be your early warning system.

- Containment – where you know about the attack's behaviour and look for unique behaviour that could be used to quarantine the attack, should it enter your business.

The challenge we all face is what and where best to implement new technologies; what is critical to one organisation may be just another asset to another. As we strive to build a security model that allows impact-free business practice the threats continue to evolve, pushing your security strategy towards obsolescence unless you keep pace.

Managing the security risk in its own right has become a business process which, not unsurprisingly, technology is also trying to solve in the guise of vulnerability assessment and management tools that look to automate this process for business on an ongoing basis. In the future these will become the eyes and ears that feed the knowledge to the business in terms of critical areas of infrastructure, the potential threats and attacks that can be run against them and, most importantly, offer guidance as to what behavioural controls we should be implementing to mitigate such challenges to our business.