

# Securing the Software Supply Chain

June 18, 2009



## Introduction

Much of the emphasis on IT security is placed on preventing attacks from outside sources that use a variety of methods to find their way to endpoint machines or servers in order to perform some form of malicious activity once they reach their target. Multiple layers of defensive technology exist to recognize such attacks in the network with the intent of preventing these malicious payloads from reaching their targets, while other software prevents such payloads from executing if they slip through the network protection and reach their target. The presumption is that these exploits must find some back door to enter the target systems in order to perform their designated activity.

The global economy and the growing demand for new applications software have created opportunities for cyber criminals to walk through the front door, hidden in what is believed to be trusted software willingly introduced to endpoints machines and servers. By subverting the software supply chain, savvy cyber criminals can circumvent existing defensive software and broadly position malicious code across the organization's endpoint population, thereby significantly increasing the chance for success. Entering from the inside allows these programs to often operate without detection for extended periods resulting in significant breaches and potential data loss.

The software supply chain is noted explicitly in the White House Cybersecurity Policy Review as an area of concern:

One of the results of the information technology revolution and free trade policies is a global environment for research, design, manufacturing, and servicing of information and communications products by corporations with facilities spread across the globe. This global marketplace has created tremendous benefits for U.S. industry by opening markets worldwide for high-tech U.S. goods and services. However, the emergence of new centers for manufacturing, design, and research across the globe raises concerns about the potential for easier subversion of computers and networks through subtle hardware or software manipulations.<sup>1</sup>

Businesses and government agencies must consider the software supply chain as an area of potential risk and take steps to evaluate software to identify supply chain vulnerabilities before the software is deployed, and to detect malicious activity if the proactive measures fail.

## Incompetence versus Malicious Intent

There is a corollary to Murphy's Law called Hanlon's Razor that goes as follows:

*"Never attribute to malice that which can be adequately explained by stupidity, but don't rule out malice."*

In the spirit of Hanlon's Razor, it should be noted that not all security vulnerabilities that result from issues in the software supply chain are overt and purposeful attempts at malicious activity. Many of the vulnerabilities that can be traced back to the software supply chain result from a lack of security disciplines in the development process versus a purposeful subversion of the code. But these

---

<sup>1</sup> The White House Cybersecurity Policy Review; May 29, 2009; p. 34

vulnerabilities are no less dangerous and can be widely exploited by cyber criminals always looking for cracks in the wall to exploit. It is therefore prudent to consider both incompetence and malicious intent when examining the problem.

### **Ignorance/Incompetence**

A painful truth is that the majority of software is neither designed nor built to be secure. The root of this truth lies in the lack of secure development lifecycle training in the development of programmers both domestically and abroad. Today's programmers emerge from programs that do not expose them to the basics of secure coding and engineering, much less fully integrate security into their foundational principles and practices. Add to this lack of training the emerging rapid development methodologies and the growing use of mash-ups and the result is software that is an easy target for exploitation. The absence of malicious intent does not minimize the potentially disastrous effect of poorly written software, particularly when cyber criminals identify the areas of vulnerability and write exploits to mine these vulnerabilities.

A good example is found in the continued prevalence of SQL injection attacks. These attacks are essentially based on a technique that exploits failures to properly validate user input. SQL injection has been on the scene for years, yet they continue to be successful because programming standards to prevent such attacks are still not being put into practice. As a result, the *Verizon Business 2009 Data Breach Investigations Report* notes that SQL injection attacks:

“...are growing notably more sophisticated, especially for data compromise scenarios. It is often used to gain deeper access into systems and plant malicious software.”<sup>2</sup>

The VZB study notes other tried and true exploits such as buffer overflows, access control lists, exploitation of session variables and privilege escalation show up frequently in the investigated breaches.

Other reports such as the *Microsoft Security Intelligence Report* indicate that attack vectors have shifted significantly to an application orientation as operating system vendors continue to close vulnerabilities. Further enabling the shift is the increasing capability of defensive software in detecting OS exploits. The latest version of the *Microsoft Security Intelligence Report* noted the following:

“The proportion of vulnerabilities disclosed in operating systems continues to decline; more than 90 percent of vulnerabilities disclosed in 1H08 affected applications, rather than operating systems.”<sup>3</sup>

Consequently, organizations must make an evaluation of the vendor's track record in properly applying the rigors of secure development into their development methodology and investigate the historical performance of the application in regards to reported breaches. Such consideration can no longer be an afterthought or secondary requirement; it must be one of the primary requirements that must be satisfied before an application is deployed into the organization's endpoint population.

---

<sup>2</sup> 2009 Verizon Business Data Breach Investigations Report; Verizon Business RISK Team; 2009; p. 17

<sup>3</sup> Microsoft Security Intelligence Report, Volume 6; Microsoft, Inc; p. 8

## **Malicious Intent**

The globalization of software design and manufacturing raises the potential for the subversion of software either by organized groups of cyber criminals seeking financial gain or by unfriendly nation-states with more hostile intent. In some cases, the subversion creates vulnerabilities that can then be exploited remotely by those with knowledge of the created exploit to carry out all nature of malicious activity. The second category actually places malicious software on the machine such as key logging software, command shell processes or routines to capture and store data.

Regardless of the ultimate attack vector, the goal of the cyber criminal is to sufficiently hide the vulnerability or malicious activity from detection until the software is willingly installed, often directly onto machines that contain sensitive or confidential information. Once the subverted application is installed, the malicious code is activated through a variety of triggers so that it can perform its intended activity in anonymity. This process circumvents the normal entry points that are patrolled by signature based software targeted at blocking the entry of malicious payloads through the standard pathways for attack. As a further bonus, the cyber criminal does not have to worry over propagating the exploit to other machines as the trusted application is willingly installed on multiple endpoints by the IT support staff.

The resulting breach is a cyber criminal's nirvana: an intrusion that operates in silence, continuously extracting the targeted data at a rate that is profitable but not so extreme as to call attention to itself and cause detection. Such breaches can run for indeterminate periods resulting in a significant loss of data. Organizational goodwill is also affected by questions of how such an attack could operate for so long without detection. The resulting loss of confidence in the breached organization can have negative effects on its ability to do business both short and long term.

## **Filling the Gaps in the Software Supply Chain**

Many businesses and government agencies have processes in place to safeguard their respective organizations against problems with the software supply chain. For example, many larger organizations thoroughly evaluate new software in a laboratory environment to uncover any potential exploits built into the code before it is deployed. It is not the purpose of this paper to define or critique methodologies for such testing, as there is much written on the subject. What is germane to this paper is the fact that in spite of sophisticated and rigorous testing processes, many exploits and vulnerabilities escape detection.

Vulnerabilities resulting from coding errors or a lack of security rigor in the development process place organizations at risk until the vulnerability is found, reported, and then acted upon by the vendor or the IT security community. Ironically, the highest exposure may come after the vulnerability is made public and before the associated "fix" is created, distributed and ultimately applied to the organization's endpoints. Cyber criminals have proven time and again that they are able to exploit reported vulnerabilities faster than systems can be hardened by the good guys to counter the vulnerability. In fact, the Verizon Business report notes that of the breaches that exploited a known vulnerability, all of these breaches were against vulnerabilities that had been public for six months or more.<sup>4</sup> It should also be noted that such vulnerabilities are not isolated to smaller vendors with lesser

---

<sup>4</sup> 2009 Verizon Business Data Breach Investigations Report; p. 18

known applications. Well established and ubiquitous software vendors such as Microsoft, which has long had a patch Tuesday for vulnerabilities, and Adobe, who recently started their own version of “patch Tuesday” to regularly disclose vulnerabilities and exploits found in their software, are highly visible examples.

Logic would dictate that vulnerabilities will be found first by those expending the most time and energy to find them. In most cases, that would be the cyber criminals. There was a day when a discovered vulnerability would publically exploited to generate fame and notoriety to the hacker that made the discovery. Not so with the new breed of organized cyber criminals seeking financial gain over attention. These hackers value stealth and anonymity above all else and will carefully and deliberately exploit a vulnerability for as long as possible. The news is full of stories of companies discovering breaches that have been ongoing for months and even years.

When malicious intent is at the heart of the problem, detection becomes far more complex as cyber criminals are constantly and relentlessly looking for new ways to hide their activity. In regards to the software supply chain, new exploits are being developed daily that successfully hide from testing activity and only activate after the software has been broadly installed and presumed to be safe. It stands to reason that if someone goes through the trouble to corrupt application code and wait patiently for it to be distributed and eventually installed, they will want their patience rewarded by a long term infiltration and will therefore take steps to avoid detection at all costs.

Traditional defensive software on balance is simply not designed to see such attacks. This software is designed to be a shield against the introduction of malware from outside means such as the network, email, and the Web. Most of it is based on signatures to detect the attack, and new exploits will not yet have an associated signature allowing them to evade detection. There are some new applications of behavioral analysis and heuristics that may detect the attacks, but the vast majority will evade endpoint protections.

## **A New Way to Detect Malicious Activity**

IT Security experts have long believed that critical gaps in endpoint protection could be closed by a solution that identifies unexpected changes and conditions (anomalies) on computers to spot previously unidentified threats before they cause significant harm. Triumphant Resolution Manager is the realization of that promise, using granular change detection to detect, analyze, and remediate the cyber threats that evade traditional defensive software. Triumphant requires no prior knowledge of the attack in any form to be able to detect the attack, eliminating the need for signatures or specific heuristics. This unique method of detecting and analyzing malicious activity enables Triumphant to address the problems that result from gaps in the software supply chain unlike any other product.

Triumphant continually scans each endpoint computer for unexpected changes to over 200,000 attributes, including registry keys, security settings, port settings, and performance statistics. Initially, Resolution Manager takes the full set of collected attributes and processes that data into a highly detailed analytic model of the endpoint population, called the Adaptive Reference Model. This model creates a set of analytical rules that serve as a baseline of the population (or defined groups within that population) for the purpose of identifying unexpected changes and conditions. Resolution Manager

---

continually refreshes the Adaptive Reference Model in order to assimilate the evolutionary changes that are normal to every endpoint environment. As a result, the Triumphant platform uniquely learns about the distinct profile of each organization and adapts its analytical rule to that profile even as the organization evolves over time.

Once the Adaptive Reference Model is built, Resolution Manager performs two continuous scans:

- The first scan looks for changes to all 200,000 plus attributes for each machine, collects those changes, and sends them to the server once per day (default, can be more frequent) for analysis. This broader scan looks at elemental attributes such as the registry, physical attributes and memory tables and performs an MD5 hash of all system files.
- The second scan loops approximately every thirty seconds looking for markers that are consistent with the behavior and structure of malicious applications. These markers include things as unusual auto-start methods, stealth techniques such as those used by root kits, and unusual firewall exceptions. This information is sent directly to the server, which makes an initial analysis and then requests additional information from the endpoint. The purpose of this request is to ensure that all of the elements of the attack are discovered by performing such analysis as dependency walks on affected files and gather data that has a temporal relationship to the event to ensure all possible information is gathered and that the analysis can establish the boundaries of the event. It is this scan that enables Triumphant to detect malicious activity in real time.

The patent pending analytics that build the Adaptive Reference Model are the true differentiator of Triumphant and enables Resolution Manager to see the malicious activity that evade other products and methods, particularly signature based tools. When a change is detected by either scan, the analysis of that change leverages patent pending analytics that consider the change in the broad context of the endpoint population via the Adaptive Reference Model, not just within the context of the individual machine. This allows Triumphant to determine if the change is anomalous or part of normal operational changes to the population, effectively eliminating false positives. This is critical, as the false positive problem has been the limiting factor in the success of anomaly-based detection until now.

Once malicious activity is confirmed, the server uses the gathered information to build a remediation in real-time to address the detected attack. The remediation is surgical and comprehensive, deleting the malicious code and addressing all collateral damage from the attack, effectively eliminating the need to re-image the machine. Other products delete the offending executable only, leaving the machine highly vulnerable to subsequent attacks or to multi-step, multi-payload attacks. Resolution Manager builds sophisticated remediations that perform complex operations to eject rootkits, neutralize watchdog processes, uncloak hidden processes, and identify randomly named executables. Triumphant sends the remediation to the affected machine for execution and notifies the administrator with the appropriate alert. This ability to build sophisticated remediations without human intervention significantly shortens the time between detection and remediation, all but eliminating the associated labor costs.

Best of all, the longer Triumphant runs, the smarter it gets, because it continually captures knowledge in tool's Incident Knowledge Base as it encounters problems. Once Triumphant detects a malicious attack, it assimilates the data about the event and can then use that information to look for other instances of the activity or handle subsequent encounters with the same problem. In fact, the administrator has the option to force a higher priority scan of specific machines or a group of machines if there is a

suspicion that these endpoints may be under a group attack. This scan will leverage the information collected in the initial event to identify other occurrences of the problem and apply the appropriate remediation.

## **Applying Triumphant to the Supply Chain Problem**

The unique capability of Triumphant to detect malicious activity is a perfect antidote for the problems that result from the software supply chain. Not only will Triumphant see what other defensive tools will miss, it will remediate the attack and all collateral damage. Once the problem is detected and remediated, Resolution Manager will capture the information about the attack so it can be readily detected in other endpoint machines. Because Triumphant captures the data about each application and the endpoint machines running those applications, it is a simple task to scan the machines running the infected application for the malicious activity and eradicate the problem across the endpoint population before it becomes a widespread attack that causes major problems for the organization. Combining the ability of Triumphant to learn about previously unknown malicious attacks and readily know what machines are running the infected application means that the span from detection to complete eradication can be minutes instead of hours or days. Triumphant will continue to scan for the attack after the initial wave addressed, ensuring that any future encounters with the malware will be quickly addressed.

For issues that arise from ignorance or incompetence, Triumphant will see the attacks that seek to exploit the newly discovered vulnerabilities in the application software. Such new exploits will likely have no associated signature and will therefore be missed by traditional defensive software at a rate that exceeds fifty percent. Triumphant will detect the changes that mark the attack, perform the analysis and create the associated remediation. This information will be stored in the Incident Knowledge Base, so subsequent attacks will also be detected and remediated. The endpoint population will be protected through the process of reporting the vulnerability and having either the organizations antivirus vendor of record or the application's vendor build either a patch to fix the vulnerability or a signature to detect exploits targeting the vulnerability. In either case, this process may take days or weeks, and Triumphant is there to detect and remediate the problem in five minutes or less, effectively closing this gap and eliminating a period of significant risk for the organization.

Triumphant also performs continuous enforcement of security policies and configurations, enabling the creation of policies that dictate specific configuration settings for machines. Once such policies are in place resolution Manager will detect when machines are in non-compliance with these settings and build a remediation to return them to compliance. In many cases when an exploit is found in an application, the vendor may recommend that certain registry settings be changed as a stop gap measure until the vendor can write and deploy a patch. Using Triumphant, a policy to deploy such settings can be built through the wizard driven interface in minutes and implemented on the appropriate endpoint machines. Triumphant then enforces that policy on every machine, every day, ensuring that the interim measures are in place until the appropriate patch or fix is received by the vendor and deployed.

For malicious subversion of the application code, Triumphant will detect the changes to the endpoint machine performed by the malicious code as it comes to life and begins its activity. Most defensive software is positioned to see malicious payload as they arrive at the machine through the network, through email, or through external storage devices. In this case, the payload has been placed on the

machine through a seemingly benign installation process, but has been inert. Regardless of whether the malicious code has been lying dormant on the machine for weeks or months, when it begins to load and execute it will cause changes to the endpoint machine, and Triumphant will see those changes when they do finally occur. Again, it is important to note that Triumphant succeeds where other products fail because it requires no prior knowledge of the attack, and it is using the malware's own activity as the trigger for detection. Once the malware surfaces on any given machine, Triumphant will learn about the attack and will subsequently be able to detect it when it appears on other machines.

If the attack is more passive in nature and designed to create vulnerabilities in the machine – such as changing a firewall setting or opening a port - these changes will be also detected by Triumphant and remediated. Changes to physical settings of a machine will be detected as anomalous without the need for specific configuration policies. Because Triumphant takes an MD5 hash of all of the system files, if there is an attempt to replace a valid executable with a corrupted version of the same name and size, the corrupted version will hash differently and will therefore be detected. If the attack deletes files on the machine, that activity will show up as unexpectedly absent anomalies. For either corrupted or missing executable, Triumphant has the unique capability to use machines with the same versions of the software as donors to obtain copies of the problem files and replace them to remediate the problem with the need for IT support or software installs.

## **Conclusion**

The potential security issues inherent in the software supply chain must be considered when building a defense in depth strategy. A globalized economy, rapidly evolving development techniques, a growing lack of security discipline in development methodologies and the organization of cyber crime all converge to make this an issue that simply must be addressed. Statistics shows that the exploits created in application development through either incompetence or malicious intent are a significant source of costly data breaches. Further compounding the problem is the fact that these exploits are not readily detected by the current defensive software.

The unique capability of Triumphant to detect malicious activity is a perfect antidote for the problems that result from the software supply chain. Triumphant uses granular change detect, analyze and remediate malicious activity on endpoint machines, including those attacks that can be tied back to the software supply chain whether they be the result of incompetence or malicious intent. Triumphant is able to see the attacks that evade other defensive tools, and because it sees everything that was changed on the endpoint machine as a result of the attack it can build holistic and surgical remediations that stop the attack and repair all collateral damage. Triumphant Resolution Manager actually learns and adapts to the organization's endpoint environment and will catalog what machines are running specific applications in that environment. As a result, if an attack comes through a specific application, Triumphant can quickly eradicate the problem across the entire population, making the span from detection to complete eradication minutes instead of hours or days.

**About Triumphant, Inc.**

Triumfant® leverages a one-of-a-kind ability to detect, analyze and remediate changes to endpoint computers and servers to create compelling solutions for endpoint security, compliance and configuration management, and incident and problem management. These solutions, powered by the Triumphant Resolution Manager™ platform, enable businesses and government agencies to reduce IT support costs, minimize security risks, enforce continuous compliance and increase quality of service. For more information, visit [www.triumfant.com](http://www.triumfant.com) and follow us on Twitter at [www.twitter.com/Triumfant](https://www.twitter.com/Triumfant)

**Contact Triumphant:**

Triumfant, Incorporated  
800 King Farm Boulevard  
Rockville, MD 20850  
301.917.6280  
301.917.6299