



eEye® Digital Security Technical Brief

Maintain Scheduled Patching Cycles

Software vulnerabilities are being leveraged into attack faster than ever before. What once took months (e.g. NIMDA leveraged an 11 month old vulnerability) is now happening in a matter of days (Sasser appeared 18 days after the vulnerability on which it preyed was announced and the patch was made available). This rapidly shrinking window to remediate has created a critical process execution challenge – how to effectively shield individual assets in the face of relentless attacks which are occurring at a record pace.

In theory, the frequency and ease of which vendors make patches available would seem to facilitate a more secure enterprise. In practice, however, organizations are faced with the daunting reality of the testing, deploying and verifying a myriad of patch installations. Under the best of circumstances, this process requires a combination of process and technology to minimize business disruption and associated costs. Unfortunately, the rapidly shrinking window to remediate means patch deployments are rarely carried out under the “best of circumstances”. Frequently, security and IT teams are forced into “panic patching” systems without the proper testing and validation. This hurried approach results in tangible losses in end-user productivity, business disruptions and related IT resource drain.

A closer look at the recent Sasser worm attack demonstrates a real world example of this challenge:

In April, 2004, Microsoft released security bulletin MS04-011, addressing 14 separate vulnerabilities with a single patch. The sheer size of this patch, coupled with the wide range of applications and services affected by its application gave security administrators considerable pause as they evaluated deployment options. This pause was short-lived, as Sasser quickly appeared, leveraging the LSA Service vulnerability and infecting nearly a half million machines in less than 24 hours. Organizations were compelled to patch systems immediately, forcing patches across the network, disrupting end-user productivity and service availability as machines were taken through the deployment process.

Unfortunately, the patch did not perform as the panacea that was intended. Soon after, Microsoft released an update warning of issues discovered with MS04-011. These issues included incompatibility with certain drivers, custom device libraries and instances of log-on access problems. These issues clearly demonstrated the requirement for rigorous testing and planning for patch deployments during regularly scheduled maintenance windows, rather than during panic cycles, regardless of the presence of imminent threats to the network.

For enterprises to obviate the need for “panic patching” requires a solution that can protect against classes of attack, not simply relying on signatures of attacks, which can be easily altered with each new attack variant.

Blink® End-Point Vulnerability Prevention

The Blink End-Point Vulnerability Prevention solution is comprised of several proven security technologies, including system and application firewalls, intrusion prevention system, non-intrusive protocol analysis, and local vulnerability assessment scanning technology based on eEye's award winning Retina® Network Security Scanner. With this comprehensive protection, Blink repels all classes of attacks, including known vulnerabilities and exploits, as well as undetected vulnerabilities, which could result in "zero-day" attacks.

How Blink Protected Against Sasser

Sasser was a buffer overrun class of attack, whereby an attacker exploits an unchecked buffer in a program and overwrites the program code with data. If the program code is overwritten with new executable code, the program's operation can be dictated by the attacker. No signature existed for Sasser, yet systems protected by Blink were impervious to the attack - even if the related patch had yet to be implemented. Blink was able to detect the buffer overrun attack and thwart the attempted code execution.

Blinks protects from *within* the network, providing the network administrator with the ability to plan the necessary down time to patch systems, rather than running the risks associated with "panic patching".

For the added protection of individual digital assets, Blink includes the following features:

- **Malicious Application Control Prevention** protects against application hijacking via DLL control hooking
- **Application Policy Control** prevents abusive user behavior within applications, such as downloading files via p2p or Instant Messenger applications
- **Buffer Overflow Protection** protects against known and unknown buffer overflow attacks against network applications
- **Non-Signature Based Attack Prevention** detects and blocks attacks without the need or use of attack signatures. This translates into complete protection, even when an attack is circulating, but the vendor has not yet created signatures or provided patches. This also removes the administrative burden associated with updating signatures files
- **Inbound and Outbound port blocking:** Blink controls all aspects of network traffic including all inbound and outbound connections. Blink also controls traffic based on protocol, port, and communicating host address
- **Configurable rules:** Blink's policies are customizable by the administrator and can be tailored to each particular worker's access or configuration requirement
- **Operating System Hardening:** Blink acts as Windows hardening solution, preventing attacks from modifying critical OS binary files or configuration settings
- **Centralized Deployment, Administration and Management:** Blink's Security Console allows for the centralized management of Blink deployments and allows for central policies to be created and implemented to one, many or all of the Blink agents in the environment.

To learn how the Blink Vulnerability Prevention solution can **Protect Remote and Mobile users**, [click here](#).

To request more information regarding Blink, [please click here](#).