October 2002

InVircible for the Enterprise

by

Zvi Netiv, chief designer of InVircible

InVircible Enterprise is a security software package devised to protect computer systems from cyber threats and attacks, including worms, Trojans, viruses, hacking tools, backdoors and compound attacks.

InVircible is based on proven generic technology that provides self-sufficiency in the protection of computer systems, with fail-safe performance, and does not depend on critical updates. These goals are attained with no adverse effects on computers' performance, and no penalty on system resources.

InVircible Enterprise consists of the following main elements:

- ***Interceptor*** – a process that runs in the background on clients' machines
- ***IV Administrator*** – an administration and command module that coordinates and manages the functioning of InVircible throughout the enterprise and WAN
- A proprietary **communication protocol**, embedded in both *Interceptor* and in *IV Administrator*, that integrates InVircible into a coherent enterprise protection system. The IV internal protocol is independent of those used for networking and is indifferent to them

Interceptor is the key element of the InVircible system. The *Interceptor* roles are:

- Interception and prevention of hostile presence and offensive activity on the local machine
- Real time protection of the local machine
- To report intercepted events and activities to *IV Administrator*
- React to threats in predefined scenarios, or under the command of *IV Administrator*

IV Administrator enables command and control of InVircible's operation, in real time, throughout the entire organization. The *IV Administrator* functions are:

- Merge the reports from individual clients' *Interceptor* into a unified database, sort the reported events, correlate them for similarities, and present the data in prioritized profiles, on demand

- Issue “red alerts” on predetermined sets of conditions and events
- Act as a crisis management system and a real time assistant in decision making on the best reaction to an evolving threat or attack
- Order, monitor and selectively control the *Interceptor* response on clients’ machines, as well as at the enterprise level
- Allow remote administration of InVircible in the enterprise, through remote access
- Provide database services for the assessment of events in progress, as well as for debriefing after the incident is over

Threat Assessment

The virus scene has changed drastically over the last few years, necessitating the reassessment of both threats’ characteristics and the methods used to counter them. The following are collective characteristics and trends identified through analysis of the most “successful” malware from the last few years:

- Threats now propagate faster than ever before, through multiple channels and highly diversified propagation modes. Nimda, for example, propagates in any of six distribution modes, each of which is entirely independent of the others.
- Destructive payloads, where they exist, trigger immediately or after a short delay. In the enterprise environment, threat alerts force the power down of critical services like mail servers and gateways.
- New malware uses diversified technologies, in complex and compound attack schemes.
- Most modern threats take advantage of vulnerabilities that exist in every aspect of computing and networking. A couple of examples are the “incorrect MIME header” exploit of e-mail, and the “share level password” vulnerability in Microsoft’s networking.

Even an optimistic forecast of future trends must assume that new threats will be at least as fast, complex, diversified and unpredictable, as those of recent vintage.

Effective protection from both current and new threats requires a clear change of focus from reactive identification of known threats, exclusively, to proactive alerting and prevention on generic grounds. The generic approach provides better protection against a wider spectrum of threats, old, current and new, in real time, at a smaller cost and with less effort, and with negligible penalty on performance and system resources.

InVircible’s Generic Technology

In contrast to virus-specific software, InVircible uses no virus-specific information or database. The methods used by InVircible are generic, which means they are effective against groups of threats that share a common characteristic or behavior. Unlike virus-specific AV, which use pattern recognition as their only detection method, InVircible implements multiple and mutually independent methods, simultaneously.

A few examples of generic methods follow. Note that the examples given are just a few of the many methods implemented in InVircible. The reader may be led in thinking that these are all there is in InVircible. This is definitely not the case. The following examples were chosen simply because they are easy to understand. Each represents a generic principle, and together they demonstrate well the diversity in methods.

Example 1 – deceptive file naming: a trivial yet effective method to block certain types of attacks. For some years now, malware writers discovered that users can be misled by apparently innocent yet bogus naming of attachments to e-mail. The deception is sometimes accomplished by using double extension naming, like '.jpg.exe', or '.jpg.pif'. The latter is trickier since the PIF extension won't show in Windows Explorer, being interpreted as a shortcut. InVircible examines such file naming and flags the bogus ones.

Example 2 – suspicious PE code: Portable executables (PE) are the file type used for applications under 32 bit Windows. It turns out that much malware (viruses, Trojans, droppers, etc.) discloses its presence in PE objects when examined for giveaways stored in the InVircible knowledge base. The Bugbear worm, for example, was blocked on the basis of this method, by a one year old version of IV, a couple of weeks before the AV industry gave the worm its name, and without requiring any update.

Example 3 – intrusion monitoring: Most new "compound threats", especially worms, hacking tools and backdoor Trojans, require initialization under Windows either by installing themselves in the startup queue, or chaining themselves to the Explorer "shell", or stealing the 'exefile' shell 'open' and allocating it to themselves. InVircible constantly monitors all strategic entrances to the system for attempts to take over the system and allows the reversing of the changes before the intruder has the chance to entrench. The more critical functions like 'shell open' are restored automatically by IV, in order to prevent losing control of the system.

Example 4 – real time integrity. One of the first generic techniques pioneered by NetZ Computing is integrity checking and file recovery. Over the last twelve years, NetZ has perfected the method to its current state, as implemented in the Audit & Integrity (A&I) module. The incorporation of real time integrity monitoring in *Interceptor*, was the next logical step. This captures PE virus infection (compound threats sometimes use secondary PE infection as part of their attack scheme, e.g. Elkern which is dropped from Klez), and blocks the infection in its tracks.

Additional generic methods implemented in InVircible include: sequential launch of executable baits (effective against memory resident infectors), test for spawning, content interpretation by inference engine (effective against macro and script viruses, like LoveLetter among others), self integrity testing (prevents IV from becoming infectious itself, and effective against many viruses), black-listing. The list goes on …

The implementation of multiple generic methods in a single application provides the exceptionally high probability of detection that characterizes InVircible, regardless of the threat being of a known nature, or entirely new, and without depending on critical or scheduled updates.

The following summarize the design philosophy of *Interceptor*:

- Simultaneous use of multiple methods
- No interdependence between the methods, each one is based on a unique generic principle
- Linear combination of all methods for maximum probability of interception
- Maximize the probability of interception of every particular threat, by maximizing the number of methods to which the threat responds. For example, Klez and its Elkern load are intercepted by not less than five different generic methods, implemented in *Interceptor*

Criteria for success: From the way InVircible is implemented, tripping on a single method implemented in *Interceptor* is all that is needed to trigger an alert, hence this is considered "success", while only evading all IV methods can be considered as total failure to perform.

InVircible in the Enterprise Environment

The InVircible mission in the enterprise is to assure uninterrupted operation of applications and critical services by protecting client platforms in a self-sufficient way, without depending on critical updates, with minimal maintenance requirements, and using minimal manpower.

InVircible is deployed to the clients from the server, through the logon script. An installation wizard is available for installing IV to the server, and configuring InVircible on both clients and server.

At the client level, InVircible provides self-contained protection to the local machine, either independently, or under *IV Administrator*'s command.

At the enterprise level, IV provides timely alerting, with detailed information on the nature of the reported events. Thanks to its generic nature and to the enterprise integration system, the person in charge can assess the problems, plan a reaction and conduct the required activities, without leaving his console, in a self-sufficient way. *IV Administrator* also enables security management through remote access, across the entire system, at all levels, from single client to the entire network/domain.

InVircible provides the enterprise with the means for taking complete control of security problems, regardless of whether the attacker is common and known, or totally new, and critically, to prevent the situation from evolving into a full scale crisis, without depending on assistance from outside.