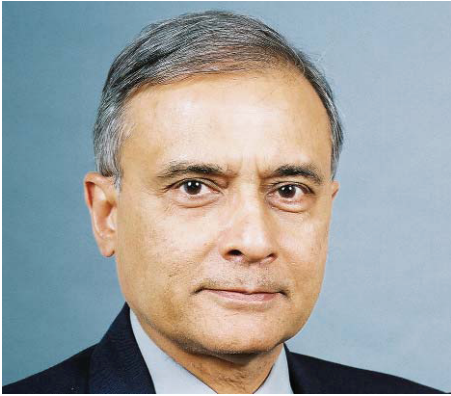


Self Cleansing and Intrusion Tolerance

Another layer of server protection

Arun Sood and Andy Purdy



Networks and the systems that run on them have become essential to national defence, the continuous operation of business enterprises, the proper functioning of the global economy, enhancing the productivity of business processes and exploiting the conveniences and ease of use provided by Internet technologies. Yet in spite of increased focus and large investments in computer security, critical information systems remain vulnerable to attacks. The problem stems in large part from the constant innovation and evolution of attack techniques, and the rapid development of exploits based on recently discovered software vulnerabilities. Some recent attacks appear to be the work of organisations with a financial or political motivation. The increasing sophistication and incessant morphing of cyber-attacks lend importance to the concept of intrusion tolerance: a critical system must fend off, or at least limit, the damage caused by unknown and/or undetected attacks.

Intrusion tolerance – a new computer security paradigm

Current security architectures depend on intrusion prevention systems, including firewalls, and intrusion detection systems (IDSs). However, both these approaches are reactive – they require prior knowledge of all the attack modalities and software vulnerabilities, together with the development of corresponding reaction rules. It is difficult enough keeping track of the attack methods currently being employed; it is nearly impossible to predict future attacks and anticipate undetected vulnerabilities. **Prevention and detection approaches are good at fighting yesterday's wars, but what about the serious current and future threats?** What about the malware installed on servers? What about the situation of inadvertent configuration errors by system



administrators? We suggest that a new layer of defence is needed that is proactive and adds to existing reactive approaches. Intrusion tolerance is one way to achieve this objective and to provide additional protection for computing resources.

Intrusion tolerance can be an effective add-on layer of defence only if it does not depend on the complete characterisation of the attacks and of the software vulnerabilities. Our implementation, Self Cleansing Intrusion Tolerance (SCIT), is focused on protecting systems by minimising the losses that can occur because of a successful intrusion. SCIT is, by design, completely independent of prevention and detection approaches. For example, while prevention and detection involve packet content analysis, SCIT does not require packet analysis.

Our underlying assumption is that all software has vulnerabilities – the more complex the software, the greater the likelihood of vulnerabilities. The discovery of a vulnerability leads the manufacturer to develop and distribute a patch, hopefully before an exploit is implemented. Constant patching of the software has now become an acceptable – though burdensome and costly – way of doing business. There are many ongoing research efforts to develop methodologies that will lead to less vulnerable software products. This research is still in its early stages and, in the meantime, there is a huge base of existing computing resources to be defended. Currently, servers that are exposed to the Internet, like those servers in the Demilitarised Zone (DMZ), rely on perimeter defence techniques to protect their computing resources. To emphasise the differences, the table below compares the intrusion tolerance with the perimeter defence paradigms.

In the SCIT approach, a server that has been online is assumed to have been compromised. To date, SCIT research has focused on a class of servers located in the DMZ. For these servers SCIT represents a paradigm shift as compared with firewalls and other intrusion prevention and detection systems. SCIT servers are focused on limiting the losses that can occur because of an external intrusion,

Comparison of intrusion tolerance and perimeter defence paradigms

Issue	Firewall, Intrusion Prevention and Intrusion Detection Systems	Intrusion Tolerance
<i>Risk management approach</i>	Reactive	Proactive
<i>Information required a priori</i>	Attack models. Software vulnerabilities. Reaction rules.	Exposure time selection. Length of longest transaction.
<i>Protection approach</i>	Prevent all intrusions. Impossible to achieve.	Limit losses
<i>System Administrator workload</i>	High. Manage reaction rules. Manage false alarms.	Less. No false alarms generated.
<i>Design metric</i>	Unspecified	Exposure time: deterministic
<i>Packet/Data stream monitoring</i>	Required	Not required
<i>Higher traffic/computation volume</i>	More computations are needed	Computation burden remains the same
<i>Patching</i>	Must be applied immediately	Can be planned

and achieve this goal by limiting the exposure time of the server to the Internet – i.e., the duration that a server is continuously connected to the Internet. In the SCIT approach, our goal is to achieve sub-minute exposure time for servers without service interruption. The SCIT approach is limited to servers that process short transactions. The SCIT research team has built laboratory prototypes of SCIT web servers, Domain Name System (DNS) servers and single sign on servers with sub-minute exposure times. In these systems redundant servers are used to ensure uninterrupted service. Our prototypes employ virtual servers based on VMware platform.

We have noted that successful intrusions usually involve the installation of malicious software on the target server. This approach generally requires the exploitation of a known vulnerability. The main objective of the SCIT methodology is to disrupt the intrusion process. We focus on making the exploitation of the vulnerabilities more difficult, an approach that is in contrast to the use of patches to eliminate the vulnerabilities. Toward this end, an online server is periodically cleansed and restored to a known clean state, regardless of whether an intrusion has been detected. This self-cleansing process depends only on the internal server clock, and we aim to repeat this process as frequently as possible. We anticipate SCIT products that achieve sub-minute exposure time. The shorter the exposure time, the less the opportunity hackers have to do damage.

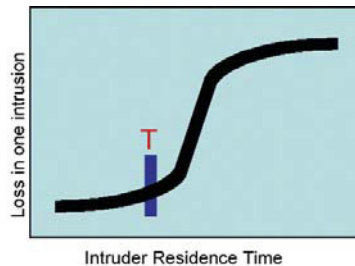
Choosing exposure time

Selection of exposure time is a compromise between the compute cycles allocated to the self-cleansing process and security requirements. Lower exposure times provide better protection, but also require more compute cycles.

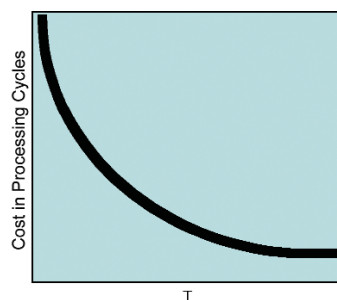
The first figure below shows the increase of loss – monetary, intellectual property or data – as the intruder residence time in the system increases. Initially the loss is small. As the intruder explores, the potential loss increases. As the intruder learns the configuration, the losses increase more rapidly. The loss curve represents a single intrusion. Subsequent intruders would go through a similar cycle. The optimal exposure time should be less than the lower knee of the curve – marked 'T'. On the other hand, reducing T

increases the cost in terms of processing cycles (see second diagram below).

Loss Curve



Processing cycles vs exposure time



The selection of exposure time is based on an assessment of the value of the resources being protected; for example, the crown jewels need more protection, and thus lower exposure times are justified. Exposure time does not depend on examining the incoming or outgoing packets, but only on the system administrator's assessment of the resources and the risk. In this sense SCIT enables data centre managers to add a proactive risk management layer of defence to the existing reactive risk management approaches. Since the SCIT approach depends on exposure time assessment, there is no interference with the perimeter defence approaches that primarily rely on examination of incoming or outgoing packets.

This discussion shows another very powerful application of the concept of exposure time – the use of exposure time as a deterministic and easily measurable security metric. Different lengths of exposure time provide different levels of security and corresponding transaction processing throughput: lower exposure time leads to more security and lower throughput, while higher exposure time leads to less security and higher throughput.

Our focus is on servers that are located in the DMZ and thus most exposed to attacks. SCIT improves security by regular automatic cleansing of the server. This

approach, when coupled with the existing firewalls and IDSs, leads to increased overall security. For example, IDSs using statistical techniques can detect sudden increases in data throughput from a server. To avoid detection by IDSs, hackers steal data at low rates. SCIT interrupts the flow of data regularly and automatically, and the data ex-filtration process is interrupted every cleansing cycle (at sub-minute intervals). Thus SCIT, in partnership with IDSs, limits the volume of data that can be stolen.

Conclusions

Many security problems nowadays are the result of exposure of the system vulnerabilities because of zero-day exploits, inadvertent configuration errors, the delayed application of patches etc. SCIT makes it difficult to exploit vulnerabilities. By reducing exposure time, SCIT provides an additional level of protection while efforts are ongoing to find and fix vulnerabilities and correct configuration errors.

In general, today's servers are online for long periods – in some cases servers are disconnected from the Internet only for software or hardware upgrades and patch applications. A typical system administrator attitude is "if it is working leave it alone". These long exposure times make servers easy targets for attacker exploration and consequent intrusion. By reducing exposure time, SCIT provides an additional layer of defence, making the attackers' job more difficult and reducing the potential damage from an intrusion.

For further information about SCIT, visit: <http://cs.gmu.edu/~asood/scit>.

This work is partially supported by a Lockheed Martin Corporation contract.

Dr. Arun Sood (asood@gmu.edu) is Professor of Computer Science and Director of the Laboratory of Interdisciplinary Computer Science at George Mason University, Fairfax, Virginia, USA. He is involved with a start-up company that is licensing SCIT technology from the University.

Andy Purdy (andy.purdy@andypurdy.com) is a member of the BigFix Executive Advisory Board, President of DRA Enterprises (www.andypurdy.com), and a partner in the law firm of Allenbaugh Samini, LLP (www.alsalaw.com).