Australian Government
Department of Defence
Defence Science and
Technology Organisation

Towards Countering the Rise of the Silicon Trojan

M.S. Anderson, C.J.G. North and K.K. Yiu

Command, Control, Communications and Intelligence Division

Defence Science and Technology Organisation

DSTO–TR–2220

ABSTRACT

The Trojan Horse has a venerable if unwelcome history and it is still regarded by many as the primary component in Computer Network Attack. Trojans have been the direct cause of significant economic loss over the years, and a large industry has grown to counter this insidious threat. To date, Trojans have in the vast majority taken the form of malicious software. However, more recent times have seen the emergence of what has been dubbed by some as the "Silicon Trojan"; these trojans are embedded at the hardware level and can be designed directly into chips and devices. The complexity of the design of the device or chip in which they are embedded, coupled with the severe difficulty of evaluating increasingly dense, proprietary hardware designs, can make their discovery extremely difficult. This paper explores the possible effectiveness of a Silicon Trojan, whether they form a credible ongoing threat, and describes possible approaches which can be used as countermeasures.

Published by

Defence Science and Technology Organisation
PO Box 1500
Edinburgh, South Australia 5111, Australia

Telephone: (08) 8259 5555
Facsimile: (08) 8259 6567

AR No. AR–014–344
December, 2038

APPROVED FOR PUBLIC RELEASE

Towards Countering the Rise of the Silicon Trojan

Executive Summary

The Trojan Horse has a venerable if unwelcome history and it is still regarded by many as the primary component in Computer Network Attack. Trojans have been the direct cause of significant economic loss over the years, and a large industry has grown to counter this insidious threat. To date, Trojans have in the vast majority taken the form of malicious software. However, more recent times have seen the emergence of what has been dubbed by some as the "Silicon Trojan"; these trojans are embedded at the hardware level and can be designed directly into chips and devices. The complexity of the design of the device or chip in which they are embedded, coupled with the severe difficulty of evaluating increasingly dense, proprietary hardware designs, can make their discovery extremely difficult. This paper explores the possible effectiveness of a Silicon Trojan, whether they form a credible ongoing threat, and describes possible approaches which can be used as countermeasures.

An overview of the basic operation of a Silicon Trojan is given and the methods by which an attacker may exploit them are explored. The basis of a potential countermeasure approach is outlined. Using these countermeasures a number of applications of the Silicon Harness are touched upon. These span a range of security goals from a Silicon Harness tightly integrated into a system design targetting high assurance applications, through to Commercial Off The Shelf (COTS) appliances retrofitted with Silicon Harness components for cost effective rapidly deployed trojan hardening. A central theme of the Silicon Harness is the provision of affordable security within a cost sensitive market.

Authors

Mark S Anderson
Command Control Communication and Intelligence Division

Mark Anderson holds a Ph.D in Computer Security from Monash University, a Bachelors first class honours in Computer Science from the University of New England, and a separate Bachelor of Science in Physics and Mathematics also from the University of New England. He also holds a Graduate Certificate in Management from Deakin University. He is the principal inventor of the Starlight suite of information security devices, and the principal inventor of the Shapes Vector system – an integrated surveillance and monitoring system for large cyberspace infrastructures. He is the Chief of Command Control Communication and Intelligence Division.

Christopher J G North
Command Control Communication and Intelligence Division

Christopher North received his B.E. (1st class Hons) in Electrical Engineering in 1989 from the University of New South Wales, an M.Sc. with Distinction in Computation from Oxford University in 1992, and a M.Comp.Sci. from Adelaide University in 2007. He now heads the Advanced Computer Capabilities Discipline of Information Operations Branch, Command Control Communication and Intelligence Division.

Kenneth K Yiu
Command Control Communication and Intelligence Division

Kenneth Yiu was awarded a B.Sc. (Appl.Ma. & Comp.Sci.) by the University of Adelaide in 1992, and a B.E. (Electrical & Electronics) (Hons.) in 1993, before joining DSTO. He now works within Advanced Computer Capabilities Discipline of for Information Operations Branch, Command Control Communication and Intelligence Division.

Contents

1 Introduction

The Trojan Horse has a venerable if unwelcome history. It is still regarded by many as the primary component in Computer Network Attack whether delivered by direct implantation, download, viral or worm propagation wrappers. Trojans have been the direct cause of significant economic loss over the years (Interlink Enterprise Computing 2006), and a large industry has grown to counter this insidious threat. Leaders amongst this industry are a number of security companies with antiviral and firewall products.

To date, Trojans have in the vast majority taken the form of malicious software. However, more recent times have seen the emergence of what has been dubbed by some as the "Silicon Trojan" (Schwartau 1994) (Adee 2008); these trojans are embedded at the hardware level and can be designed directly into chips and devices. The complexity of the design of the device or chip in which they are embedded, coupled with the severe difficulty of evaluating increasingly dense, proprietary hardware designs, can make their discovery extremely difficult. For the purpose of this paper we consider firmware embedded in devices such as hard drives and similar classes of peripherals as being included in the Silicon Trojan stable.

Of interest to us is the possible effectiveness of a Silicon Trojan, whether they form a credible ongoing threat, and possible approaches which can be used as countermeasures.

2 The Effectiveness of a Trojan

Earlier work in Information Security from the 1970s to the 1990s produced a number of basic theorems which show the potential of Trojans. Cohen, regarded as the "father" of the computer virus, showed that it was not possible to construct a program to inspect a data stream and detect arbitrary viruses (Cohen 1987) (Cohen 1989). Cohen's work showed therefore that firewalls could not be constructed which prevent the leakage or transmission of information across the firewall boundary if there was co-operating software on either side.

In the 1990s, there was a considerable effort put in by the security community in search of a "composability theorem". Discovery of such a theorem would permit the ability to fit one or more secure components together and prove that the combination was secure. McCullough developed a theorem based on "restrictiveness" (McCullough 1987) as one of the candidates in the search for composability. However, McClean (McClean 1993) subsequently showed that the successful search for an arbitrary composability theorem was not possible. The implication of this seminal work was enormous. It meant that despite all components undergoing rigorous security testing to show they operated as they were intended, their combinations could not be guaranteed to be secure without a full evaluation of the components and particular combination or overall system itself. This meant that if we consider Silicon Trojans without a fully trusted design and implementation of the component AND system, the result may be insecure, even if the overall design of the system and architecture may be ostensibly secure.

The preceding discussion shows that an arbitrary Silicon Trojan, or any Trojan for that matter, cannot be detected reliably. This would seem to indicate that the countermeasure

providers have "lost", yet total disaster has not eventuated. What is important to realise is that while a great potential exists, one must take into account whether it is practical to realise this potential. Various countermeasures, while they cannot be guaranteed to stop all Trojans, can make their use either very expensive or impractical for the desired circumstances. Firewall and anti virus technology providers rely on this maxim through providing constrained interfaces and rapid solution response times. In fact, the anti viral community operates very much in a cyber equivalent of an immune system.

3 The Changing Economics

During the 1980s National Security concerns focussed on making the operating system (OS) secure. Very large investments were made in applying military security models (Bell,LaPadula 1976) to OS such as Unix and evaluating them to various levels of assurance (Smith 2000). The most famous of the assurance criteria used was the Orange Book and its associated companions (DoD 5200.28-STD 1983) (NIST 1983). However, the small PC OS (such as Microsoft derivatives), with its high volatility and high market penetration, made the expensive evaluation process non-viable. A US Army effort to evaluate a version of UNIX cost approximately an extra US$640 per line of code (loc); experience with safety critical systems has shown this figure has risen to $1000 per loc. (O'Dowd 2004). Further, the commoditisation of smaller, powerful hardware systems which drove out the specialised custom mainframe, forced the National Security market to relinquish any real influence over the security of the base systems or their operating systems; they were no longer customers of sufficient volume for Information Communications Technology (ICT) goods and services to warrant special consideration. The National Security area, in order to utilise the ubiquity of the ICT product space had to turn to maintaining "air gapping" (i.e. not connecting their systems to anything else), network duplications, specialised security devices (Anderson 1996b) and applications, and their focus turned to "trusted" custom applications for differentiation.

But the 1990s introduced yet a further layer of complexity. Due to the Internet becoming a major economic force, the applications themselves became the major commodity factor and outstripped many previous National Security applications functionally for business needs. Further, the demand for Internet type connectivity and interoperability continued to neutralise National Security type (Evaluated Product) offerings in their applicability; the re-evaluation cost for security purposes for each upgrade simply could not be tolerated. Hence the connectivity and functionality enjoyed by millions on the Internet were denied to National Security efforts due to a need to protect high value data elements from leakage.

When even the air gapping approach became too expensive for some requirements in National Security (it was simply too expensive to maintain the human and procedural intensive data transfer methods between air gapped networks), "risk management" approaches were introduced whereby through a variety of methodologies, it was considered that the benefit of connectivity outweighed the risk of leakage of unauthorised information; the authors contend that a number of these methodologies do not necessarily hold sufficient rigour to justify the so-called acceptable risk rating supplied.

It is interesting but not surprising to note that many in the National Security space now consider the connected systems to be more valuable than the air gapped systems containing highly classified data. This attitude is in tune with well known pronouncements relating to the value being in the level of connectivity. Perhaps one of the more famous of these relates to the increase in the value of a telephone network from the early years of telecommunications (Metcalfe 1980).

In considering the connectivity issue and security, it is interesting to note that the globalisation of economies has forced a fundamental change in Government procurement practices for critical systems, including those on which National Security depend. It is now almost inconceivable for many Western nations to source most of their chips and low level devices from inside their own country, let alone Government controlled foundries. In the near future if not already, it must be expected that very low level components for weapons and critical National Security systems will be sourced from nations which could be adversaries. This curious arrangement makes the effort of a provider nation in doping low level components with Silicon Trojans an attractive potential option.

For a National Security concern, maintenance of the ICT security architecture has never been more difficult. There is little control or influence at the hardware, network, or application level for the products manufactured. Updates to all these elements occur at a pace much faster than any independent evaluation can take place and occur from multiple locations across the globe. Certificates, digital signatures etc are not of great value when the company supplying the update is in a conglomerate of nations, some of which are probably adversaries to the National Security concern. There is a current attempt to provide technologies such as separation kernels and microkernels which can be evaluated and which permit the running of "personalities" such as different operating systems on the resultant "virtual machines". Offerings such as Green Hills' Integrity kernel (GreenHills 2008) undergoing evaluation under the NSA's Common Criteria (NIAP 2008) form one example. However, such offerings rely on the underlying hardware operating as intended, i.e. the absence of the Silicon Trojan.

A 2005 report by the US Defence Science Board (OUSDATL 2005) recognised that the loss of direct control of semiconductor fabrication and electronics manufacturing capability was "directly contrary to the best interests of the Department of Defense". The report recommended efforts to counter Trojans developed in the design process, and inserted during the manufacture process. This resulted in both a "Trusted Foundry Program" used for security critical ICs, and a DARPA program to examine trust in ICs created by untrusted processes (Microsystems Technology Office 2007). Even if this program is underway, the ubiquity of use of untrusted ICs sourced from multiple nations is so voluminous one wonders how effective this effort is for the US, let alone other nations which cannot afford a trusted foundry program.

It is well known (Kumagai 2000) that reverse engineering facilities exist; techniques, machinery, and indeed commercial services exist to decapsulate chips, and to image and analyse them, layer by layer. These efforts are expensive, but the duplication of a chip design by reverse engineering (and subsequently the insertion of extra electronics) is also possible and does take place. Somewhat easier is the addition of additional electronics to existing hardware, duplicating existing packaging. The presence of a grey market in electronic components has already allowed the supply chain to be adulterated with sub-

standard components (e.g. counterfeit power supply capacitors) (KPMG 2003); purchase, modification and distribution of components via this channel is only one vector for insertion.

The authors contend that to secure the critical infrastructure ICT based systems for a Nation, insertion and control of security elements will have to be in multiple, selected portions where security architectures presume the presence of Trojans at all levels, ranging from the high end applications to the contents of the chips themselves. Security Architectures have to be Trojan tolerant rather than Trojan averse. In this case, it is suggested that rather than having a philosophy of obtaining commodity based hardware and operating system infrastructure with tailored custom applications, we assume that we should have semi-custom hardware which is able to operate with a stronger commodity based middleware and applications environment. This approach inverts the previous model that National Security concerns have been following to date in pursuing their emphasis on Commercial Off The Shelf (COTS) utilisation, i.e. assume commodity hardware and middleware, and focus on the supply of tailored applications where necessary.

One may wonder even if such possibilities of Silicon Trojans exist why we should be concerned. After all, why would a chip or device be "doped" with one or more Silicon Trojans, say at manufacture, and then sold? Will the device end up in a target of interest?

The threat becomes more tangible if the resources of a nation state which manufactures vast numbers of the products is applied to the task for espionage purposes. If each product from the various factories is doped with a type of Silicon Trojan, then targets of interest will probably end up utilising one or more of the devices, thereby providing leverage. It is a scatter gun approach, but that is the beauty of the Silicon Trojan: normal precision targeting may not necessarily be required.

4 Silicon Trojan Concepts

In this section, we introduce a number of Silicon Trojan concepts to illustrate the threat they represent, and thereby open up a discussion on approaches for effective countermeasures.

4.1 Threat Model

Traditionally, information exploitation operations have targetted high value assets. This has been dictated by the return on investment decision as the required investment to run such operations has been substantial. As such, the information exploitation operations have been highly focussed with the target being heavily surveilled and access leveraged either by physical access, or by subverting the target's hardware through some means. Ongoing access to the target typically relied on some form of dedicated communications link, including radio frequency links, as the targets were not networked – either because networking was not prevalent or because the sensitivity of the target necessitated air gapped networks. The resultant cost of these operations were expensive requiring large investments in time and the development of specialised implants for a few specific installations.

The information environment has changed considerably in the last couple of decades. Today the majority of a nation's information state resides on internetworked computers. The return on investment decision has now substantially evolved for the information warrior. No longer do large numbers of man hours need to be expended, and specialised implants constructed targetting high value assets. Rather, with the assistance of large scale silicon trojan doping programs and automated control techniques the attacker can skew the probabilities of success and through large numbers of implants obtain the necessary access to make such operations profitable.

The silicon trojan threat that we are primarily concerned with in this paper, and against which we are constructing protective harnesses, are characterised by generality of approach, and the requirement for significant automation in the control and execution of the implant networks. The highly specialised silicon implant directed against high value assets, typically requiring a priori knowledge of the target's protective measures, is outside the threat model that we are concerned with.

4.2 Intended Effects

There are a number of desired effects that an attacker may wish to exercise. Broadly characterised these are:

- Exfiltration of data
- Logical exploitation
- Disruption
- Denial of service

Exfiltration of data may include traditional network traffic redirection or content adulteration to support collection operations. However, with the aid of silicon trojans more sophisticated data exfiltration operations can be executed. These include redirection or copying of e-mail traffic via the modification of message headers. Unlike network traffic redirection which is normally targetted at physical interfaces and network infrastructure, data exfiltration attacks may be leveraged by a silicon trojan with access to network traffic residing on a computer system bus, or contained within a computer storage element such as a hard disk. In this way the e-mail messages may be redirected whilst being communicated, or whilst buffered in transit. Attacking an e-mail message in transit whilst it is buffered on a storage element such as a hard disk may have certain advantages for the attacker; the scope of attack is increased and the technical requirements for attacking data at rest may be less than for a high speed network link.

Logical exploitation is the ability for an attacker to execute arbitrary code on a target computer. Current remote network attacks rely on vulnerabilities within the operating system or applications of the target system that allow an attacker to execute unintended code sequences. In the case of silicon trojans the silicon trojan guarantees the existence of these vulnerabilities that are exploitable by an attacker at their command. The nature of these vulnerabilities are explored further in Section 4.3.

Disruption and denial of service are related concepts. Whereas denial of service aims to remove a service, disruption seeks to undermine the reliability of the service being provided. For the purposes of our discussion disruption also include the deliberate modification of data and state of the target system. In a disruptive attack an adversary may slowly degrade the effectiveness of a target system, over an extended time period, without alerting the target and having them move to a backup system.

4.3 Silicon Trojan Construction

In developing Silicon Trojans, an adversary can expend a great deal of effort in designing the most minimal of changes to the host integrated circuit in order to avoid detection. In the other extreme, sophisticated circuitry may be inserted, with the attacker then relying on the scale and complexity of the host device to hide the Trojan, or with the attacker then expending modest effort on camouflaging the Trojan. There is a continuum of approaches between these two extremes. In developing a viable Silicon Trojan, the attacker requires balancing complexity of Trojan circuitry against complexity of exploiting the Trojan. The other dimension facing the attacker in the development of an effective Silicon Trojan is evading detection, both of the Silicon Trojan and its exploitation vector.

Another difference in style of Silicon Trojans is their intended effect. The intent of many Trojans will be to reliably modify the operation of the trojaned device, usually with the goal of leveraging privilege and then executing the attacker's software payload. Another form of Silicon Trojan (primarily targeting military operations) may simply seek to disrupt, or sabotage, operations. This form of Trojan can be much simpler to design and develop, and has the advantage for the attacker in that simpler triggers may be defined. Such triggers could involve a variety of methods such as being vulnerable to wireless transmissions. A third form may be to produce compromising emanations to allow TEMPEST style attacks.

For the purposes of discussing silicon trojan construction and triggering issues we broadly categorise silicon trojans into two classes; *independent* and *hybrid* silicon trojans. Independent silicon trojans rely only on the modifications introduced by the silicon trojan into the host device – be it additional circuitry, modified firmware, substituted logic blocks, rerouted signals, or introduced electrical artifacts. Hybrid silicon trojans, however, require co-resident software processes in addition to the modifications they introduce into the host device.

To better understand the trade offs between these two classes of implants, we briefly touch upon two hypothetical scenarios. The first deals with a sophisticated adversary who has nuanced the design and/or layout of a silicon circuit to introduce a subtle exploit. The pressures and complexities facing Integrated Circuit (IC) manufacturers has meant that many commercial ICs now ship with known bugs, or errata as they are known in the industry. One of the most famous examples of these would be the Pentium FDIV bug discovered publicly in 1994 (Nicely 1994) and known about by Intel many months before the public disclosure (Intel 1994). For the defender, is anomalous behaviour the result of a manufacturer's defect, or adversary action?

4.3.1 Hybrid Silicon Trojans

A subtle exploit that can be introduced by an adversary may be as simple as intentionally violating a timing constraint, introducing deterministic cross talk, or adding a stray passive component such as a capacitor. Such a Trojan would be exceptionally difficult to detect, however likewise it may also be very difficult to reliably exploit. In this scenario it is highly likely that the adversary would utilise sophisticated system models coupled with automated search algorithms to guide the design of a minimalist Silicon Trojan whilst achieving a usable exploit. The nature of the exploit may well require executing unusual instruction sequences to trigger the exploit, resulting in the processor deviating from its intended operation, or modifying otherwise protected data. The Java virtual machine attack (Govindavajhala 2003) is an example of the style of secondary software Trojan that may be required to capitalise on this form of Silicon Trojan. The Java virtual machine attack operates by introducing bit errors into the computer's memory. These errors are then leveraged by a Java program to break out of Java's strict type enforcement, allowing the program unfettered access to system memory. However, to implement this attack, the attacker is forced to perform a brute force statistical attack with a high, but not assured, probability of success. Whilst the attack described is based around a byte code interpreted virtual machine system, the same concepts map across to a Silicon Trojan targeting native machine instructions. It should also be realised that a capable attacker is not constrained to traditional fault injection techniques and should be able to construct a far more capable and reliable attack vector than those briefly mentioned here.

4.3.2 Independent Silicon Trojans

The second hypothetical scenario investigates the sophisticated Silicon Trojan. This Silicon Trojan can be as complicated as a microcontroller or a sophisticated state machine. The great advantage with having a processor – or state machine, is that the requirement for secondary software Trojans on the target system is reduced, or removed completely. With more sophisticated hardware/firmware the Silicon Trojan is able to be triggered and controlled via purely data centric activities. One concrete example of a Silicon Trojan in this class is circuitry capable of both monitoring and modifying the contents of system buses. Through monitoring data flowing across the system bus, a covert data sequence, known only to the attacker, can be used to communicate and control the actions of the Silicon Trojan. The attacker need now only ensure that the covert data sequence (which may be contained in an e-mail, JPEG picture, or even a video stream) is processed by the circuit containing the Silicon Trojan. Predefined actions may also be encoded within the data stream allowing the actions of the Silicon Trojan to be programmed and controlled remotely. In this scenario whilst the requirement for a co-resident software Trojan is removed the physical complexity of the Silicon Trojan is significantly increased.

4.4 Leveraging Control of Silicon Trojans

To leverage control of a silicon trojan requires an attacker communicate with the silicon trojan – assuming that the silicon trojan is not a logic bomb with a predetermined course of actions and triggering events. The style and amount of communications needed to control

a silicon trojan will be dependent on its intended effect. For silicon trojans intended for disruption and denial of service this may be as simple as triggering the trojan, either via a predetermined data sequence, the data trigger, or via covert channel of some description.

Silicon trojans designed for data exfiltration and logical exploitation have more substantial communications requirements. In order to discuss the control requirements for a silicon trojan we briefly outline the phases needed to remotely exploit a silicon trojan:

- Location – Identification of hosts with silicon trojan
- Preparation – Communication of exploit and commands to silicon trojan
- Trigger – Control sequence to trigger silicon trojan
- Control – Command and control to exercise ongoing access to silicon trojan

Not all steps are necessary for all styles of silicon trojan. For example, preparation may be a necessary step for silicon trojans leveraging logical exploits (running code locally on a target), whereas it may not be required for a silicon trojan designed for certain data exfiltration activities.

With a scatter gun approach to deployment the location step may be required to identify those hosts that have an implanted silicon trojan and have the required access to data or networks of interest. There are a number of approaches that may be taken to locating silicon trojans within networks of interest, however the exact mechanism will be highly dependent on the nature of the silicon trojan and the manner in which they are integrated into the target. As a simple example we take the case of a silicon trojan that has network access and can monitor and modify network traffic. For this particular form of silicon trojan a data modulation technique may be employed whereby the silicon trojan modifies, or modulates, network traffic conforming to a predetermined pattern. The attacker need only send in network traffic conforming to this predetermined pattern and have it returned, and by doing so will be able to identify whether there are any silicon trojans within the target network.

Preparation is necessary where exploit payloads and complex command sequences need to be communicated with the silicon trojan in order to execute an attack. The complexity of the preparation phase is sensitive to the style of silicon trojan. Hybrid silicon trojans which rely on co-resident software processes can avail themselves of the co-resident software process to assist in the preparation phase. However, for the independent silicon trojan, relying only on the access and actions of the silicon trojan, the length and complexity of the preparation phase may be greater. Logical exploitation is one use of silicon trojans that will be most likely to have a substantial preparation phase; a fast changing software environment is likely to preclude the use of prepositioned exploit code.

The last two phases triggering and control are relatively self explanatory. In all phases communication with the silicon trojan can be severely affected by their location. To better understand the limitations under which an attacker may be placed let us consider three different silicon trojans; a memory implant, a network implant, and a hard disk implant. The memory implant and hard disk implant are discussed in Sections 4.5.3 and 4.5.2 respectively. The network implant is a silicon trojan implanted in network

communication circuitry and can monitor and modify network traffic. In the case of the memory implant the silicon trojan may have limitations on the data it can see and the format of that data. Some of these limitations may include paging characteristics of the system, and the memory architecture of the system such as dual channel memory. For silicon trojans embedded within hard drives they will be constrained by the nature of hard drive data storage. Data on hard drives is organised according to sectors where a sector consists of 512 bytes. Files are not guaranteed to reside on contiguous sectors, and further due to reordering and caching it cannot be guaranteed that file contents will be accessed sequentially, or even that they will reach the disk. In a similar vein silicon trojans targetting network communications will need to handle data streams that are broken up on variable length boundaries. Packet framing, variable Maximum Transmission Units (MTUs), and tunnelling can all affect the length of packets and communication frames. To ensure robust communications an attacker will need to take into account these limitations and frame their communications with the silicon trojan in such a fashion that it is resilient to the particular framing and formatting requirements.

4.5 Silicon Trojan Implementation Examples

4.5.1 A CPU Silicon Trojan

There are a relatively small number of CPUs which form the most common components in many systems and architectures. For the sake of our example, we focus on an Intel x86 style CPU.

Probably the simplest method for a Silicon Trojan in such a CPU is an instruction sequence, or data sequence which would trigger the CPU to switch to Kernel mode while still retaining the user page tables. For an x86 series type chip this would entail switching to Ring Zero (typically) from Ring Three, but without requiring the use of standard call gate type structures, privileged system calls or software interrupts for controlling the transition. This might occur for example by providing a hidden instruction (or overloading an existing one) to perform an uncontrolled transition.

The above means that even if you have a program operating well out in an outer ring, if it triggers the sequence it ends up with total kernel privileges.

This type of CPU option appears quite attractive conceptually, but is more difficult to exploit than the naive might at first imagine. For example, one must not have a trigger sequence which is easily or inadvertently duplicated by other, possible legitimate programs. Ideally, we would also want the trigger sequence to be data driven. Trigger sequences which are strongly code driven result in a requirement for a more visible secondary Trojan to be resident in the triggering software itself, thereby making themselves potentially revealed through evaluation of the software used. A data driven trigger carries the strong irony of quite legitimate, and even security evaluated software being hijacked for the Trojan purposes. However, a counter argument is that the software used is typically so complex in its own right it is not possible to undertake a proper security evaluation and thus would be of a low risk if it acted as a secondary Trojan. Nevertheless, a Silicon Trojan in a ubiquitous component which does not rely heavily on the presence of complex secondary

Trojans is far more useful to the exploiter since it is far more likely to be present in a target in a useful form.

It would not be surprising if the US National Security market encouraged CPU developers to promote new hardware security features in their processors to counter Trojans in software which cannot be detected easily and for which the cost of an evaluation is prohibitive. This is especially true of operating systems and applications which receive a seemingly continuous stream of updates. These make proper evaluation all but impossible. However, the authors suggest that just as commodity drivers caused the outsourcing of chips to overseas foundries to the point where even the design and basic masks may no longer be under the control of the client nation, so it must be expected that major chips such as CPUs with supposed inbuilt security features destined for the National Security market will, if not already, follow the same fate for even the most sensitive of uses.

4.5.2 The Hard Drive

In an example scenario, consider where a small piece of custom electronics is inserted into the hard disk interface controller ASIC in the chip foundry. It is soldered into millions of hard drives, but remains dormant until triggered. This may happen when a particular Internet cookie or email attachment targeted at one particular user or organization is saved into the temporary directory. At next reboot, a small Trojan program is inserted into the boot process, and a payload may be fetched from a remote site, or the hard drive might simply turn off permanently. The millions of drives not subject to the targeting trigger continue operation without the Trojan ever revealing itself.

The hardware drive example is quite attractive: extraction of information is still the primary motive for exploiters, and having a drive reliably yield information through trigger sequences on a disk would be an extremely valuable capability. The advantage of a silicon based Trojan is the hard drive can appear forensically clean, even after sophisticated and expert analysis. Access to the disk below the level of the file system means that the Trojan has access to large amount of persistent storage which is inaccessible without raw drive access, and can be simply triggered as it can monitor transactions across the hard drive bus. It is quite simple for a Trojan in this location to obtain system privileges by loading exploit/payload code attached to system device drivers and libraries, and this code would appear only in the computer's memory image. Any forensic analysis of the file image, without analysis of the loaded runtime memory image will not reveal the presence of the Trojan. This is particularly vexing for law enforcement standard procedure which is to seize the hard drive or copy of the hard drive, and conduct an offline analysis.

The hard drive example is not that far- fetched. Note the concern raised within the U.S. by the recent acquisition of a hard disk drive manufacturing firm by another nation (Markoff 2007).

4.5.3 Memory Example

The disadvantage of the previous example is that hard drive encryption techniques can make this type of exploitation complex. An alternate location for a Silicon Trojan is the

memory subsystem. This may be system memory itself, or any device with DMA capability. Targeting removable modules such as memory DIMMs mean that the Trojans can be rapidly installed and do not need to be customized for each motherboard. However, while memory DIMMs see raw memory, they may have to deal with a mismatch between physical and logical addressing. They also have to support high system memory speeds making insertion at this location expensive. For this reason, direct memory access devices such as Firewire (Boileau 2006) or PCI devices are simpler targets, and their silicon implementations change less frequently than memory. These devices can be configured for complete read and write access to memory, which makes inserting a payload, and extracting and modifying data a trivial exercise.

In an example scenario, a Silicon Trojan might be inserted into a Firewire interface controller. These could be soldered onto millions of personal audio devices, which synchronize with PCs via a Firewire interface. When a trigger file such as a podcast or a music file in a particular format is sent to the user and transferred to the player, it can take over the computer and execute the payload embedded in the media file.

4.5.4 Distributed Denial of Service

Much of our discussion and the current industry focus on Silicon Trojans is concerned with the threat pertaining to logical exploitation for information extraction. A much simpler attack, and one which can be very difficult to detect and respond to, is that of large-scale distributed denial of service. We briefly touched upon disruption in the context of military operations, however in this scenario we specifically look at the situation where a nation state wants to substantially degrade the communication and information infrastructure of its enemies. These attacks are easier to implement, harder to detect, harder to defend, and can cause considerable ongoing disruption. Moreover, this form of attack is particularly suited to adversaries that have some IC manufacturing facilities but may not be producing the high end ICs more typically associated with Silicon Trojans including CPUs, microcontrollers, and high-end chips associated with system bridges and high speed computer buses. In the case of distributed denial of service, an adversary preferably targets low cost, trailing edge general purpose circuits that will be integrated into a wide range of systems and peripherals. Prime targets for these chips will be physical layer chips (Ethernet PHY chips, USB transceivers), memory buffers and caches, and communication chips to name a few. Common to the Silicon Trojans described above, they need only monitor the data communications that traverse their circuits. Once triggered, through a covert data communication sequence, they can either disable the circuit, or degrade the circuit's performance, either by introducing errors, delays, or a combination of both. The overall effectiveness of this attack will be determined by the quantity and range of affected equipment. Malfunctioning chips could be scattered through computer systems, printers, network communication nodes, phones, and other digital system. As any engineer will attest, the difficulty of debugging hardware, software, or integrated systems is greatly increased when the failure mode is intermittent and when there are multiple simultaneous errors. The distributed denial of service can introduce these last two failure modalities. Couple this with a large and irregular distribution of failing equipment and the attack has the potential to bring down an adversary's networks for an extended period of time.

In keeping with other forms of Silicon Trojans, current network security measures including firewalls, virus detection, and intrusion detection systems, are ineffectual at combating the distributed denial of service Silicon Trojan. Executed correctly, and without prior software compromise, the distributed denial of service is a zero day attack. What is needed to mitigate this form of attack are hardware mechanisms to disrupt potential triggering sequences and enclave data communications.

5 The Silicon Security Harness

We discuss a concept called the "Silicon Security Harness". Just as in the macro architecture world comprising firewalls, network monitors, host based viral detectors etc., the harness involves one or more gates and monitors designed to retrofit to hardware and device components or that are installed as part of the architecture. We introduce hardware protective measures to increase the resistance of the system to resident Silicon Trojans. As with most computer security measures the aim is to increase the exploitation barrier above what is practical or economically viable for an attacker.

5.1 Protective Goals

In keeping with the threat model that we are concerned with the primary protective goals of the silicon security harness are aimed at reducing the effectiveness of wide spread general applicability silicon trojans. More specialised silicon security harness apparatus and architectures offer increased protection to silicon trojans that may be hidden in more sensitive equipment.

As already alluded to in Section 4.4 an attacker needs to carefully design and craft their silicon trojans so as to be able to communicate and control them when they are positioned deep inside a target system with a highly constrained communication path. The Silicon Security Harness leverages this and seeks to further disrupt the communication path between an attacker and the silicon trojan. To counter this defence an attacker would need to implement counter techniques which changes the original investment decision by the attacker. In this fashion we alter the cost benefit equation of the attacker by making it more difficult and costly; no longer can the attacker cast a wide net by deploying simpler general purpose silicon trojans.

The second defensive mechanism provided by the silicon security harness is provision of confidentiality for data flowing through a computer system. If communications are not able to be reliably disrupted, or if there is no requirement for in-band communications by an attacker to the silicon trojans then we can still protect the confidentiality of the data.

5.2 Silicon Security Harness – A Modular Approach

We propose a set of components which might be used as part of a security harness in order to constrain the operation of a Silicon Trojan and we briefly describe three of these harness components; Transformer, Gate, and Condition Monitor.

5.2.1 Data Transformer

The Transformer is similar to an encryption device. If the data transmission between system components is scrambled in some way, then a Silicon Trojan may not be easily activated or commanded to undertake its operation. The Transformer can be deployed in matching pairs to scramble communications across system buses, thereby thwarting components which need not be party to the transaction but have a resident Silicon Trojan. It can also be deployed in singleton configurations to scramble or encrypt data at rest within the computer system be it residing in RAM, hard disks, or buffers. Since a typical Silicon Trojan deployment is a "scatter gun" approach, the scrambling code need not be very sophisticated - even coded with a single key for the item they protect. This permits very fast transformation techniques. It should be noted that through simple defensive measures, such as periodically replacing the key and varying the key amongst devices, it can significantly strengthen this measure against evasion by the attacker.

5.2.2 Data Guard

A Gate is a device controllable by other Silicon Harness components to open a circuit and thus prevent further data flow. Many strong encryption devices carry such types of gates as low level devices and activate if they detect a failure in the crypto core. Typical robust implementations of these types of gates include in-series relays or solid state switches so that if the relays themselves fail, they fail safe in an open state, or a number of the relays have to fail the same way simultaneously for the gate to operate in a failed condition.

5.2.3 Condition Monitor

The Condition Monitor is encoded with expected valid transactions much like a firewall. It can be used to signal exceptional conditions to other security harness apparatus or it can be used to control a gate.

A security harness should not be just a set of components which monitor or condition the activity of devices and items they monitor. A sophisticated harness should have its own network in order to compare whether problem circumstances have arisen. This permits, just as in the macro architecture world, monitoring of distributed conditions and thus makes it more difficult for the Silicon Trojan to operate without detection.

5.3 Applying the Silicon Security Harness

Using two of the above components, let us now construct a harness for the simple architecture described in our CPU example. For the sake of our example the purpose of the harness is to ensure that classified secret information is not transmitted to the unclassified output. Figure 1 shows the architecture with a Condition (Access) Monitor, and several Gates. In fact, the example we are using is very similar to that originally patented by Anderson (Anderson 1996a) in the 1990s as part of a research program into this topic.

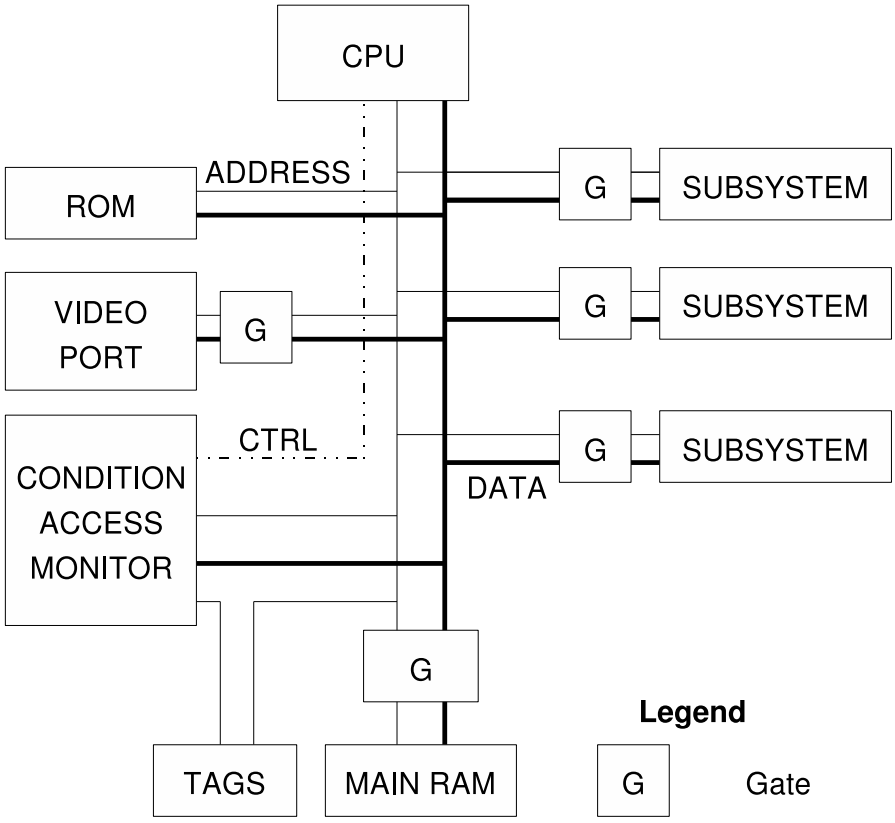

Figure 1: *Secure Computer Architecture*

An aspect of this design is that every time the CPU switches to process data of another level, the Condition Monitor forces a reset on the CPU and provides an initialisation sequence. If a Silicon Trojan existed along the lines described in the earlier section, then it would have to survive a hard reset and store information within the CPU itself for later transmission in the other state; it would have to cater for the hard reset and knowledgeably be able to detect, deal with, and bypass any initialisation sequence.

Another aspect of the design is the supply of a tag for blocks of memory locations. This permits the condition access monitor to enforce an access control policy for each block of locations and thus keep the various elements separate. However, it must be remembered that such a tag needs to be sufficiently sophisticated that a sequence does not cause the chip to yield one location in preference to another for the same address. Such sophisticated Silicon Trojans can be practically neutralised by using a form of location scramble (as used in 1980s bit slice processors) which make it difficult for each RAM chip to coordinate sufficiently with others.

5.4 A Second Application of the Silicon Security Harness

King (King, et al 2008) describes a method for tainting the VHDL of a processor design. This approach, if the preceding specific condition access monitor settings are

known, can be used to subvert the tags and condition access monitor. However, we now turn our attention to another harness approach utilised in the Annex Project (Grove, et al 2007) which neutralises King's Silicon Trojan for a number of useful applications where confidentiality is paramount.

The method uses a multiple CPU design where sensitive information of different levels are processed entirely within their own machine and memory space. Figure 2 shows the Minisec design for the Annex very High Security Communications system (Grove, et al 2007). As can be seen, conceptual Silicon Security Harness-like components encapsulate domain processing and inter-domain communications, limiting the ability of Silicon Trojans to exhibit cross-domain control. Not shown in the diagram but also included in the architecture is the ability to rigidly partition address spaces for certain high trust functions. The use of FPGA technology as the control mechanism makes it extremely difficult for it to partake sensibly in a large class of Trojan activity as the FPGA firmware load may not be known a priori by an attacker thus hiding system bus locations and other security critical details. The FPGA firmware is decoupled from the FPGA hardware allowing implementers to arbitrarily refactor the Security Harness providing a continually moving target for potential Silicon Trojan authors targeting the FPGA hardware. Additionally, testing vectors can exhaust many trigger conditions given the potential for its much simpler implementation.

The Minisec approach is more comprehensive than the previous method in that any Silicon Trojan resident in the CPU cannot interfere with other security levels without the cooperation of the more trusted hardware.

5.5 Further Applications of the Silicon Security Harness

The previous two applications of the Silicon Security Harness are characterised by their requirement for a custom baseboard to integrate the security harness components. Whilst this approach provides tighter integration and hence better security properties one of the advantages of the Silicon Security Harness approach is that it can be applied retrospectively to COTS products without substantial engineering or administrative effort. The following two applications, currently in development by DSTO, highlight the application of the Silicon Security Harness to commodity ICT hardware.

5.5.1 Hard Disk Guard

The Hard Disk Guard is an example of a combined Data Guard, Data Transformer, and Condition Monitor in a small form factor designed to reside on a host computer's ATA bus interposed between the motherboard and hard disk (Beaumont 2008a). Using standards-based connectors it can be quickly retrofitted to COTS systems thereby enforcing certain guarantees of the hard disk. Traditional disk encryption can be supported, however the true value of the Hard Disk Guard in the context of the Silicon Security Harness comes when it is used to support efficient administration and security policy enforcement through a large network. An enforced software rollback and software patch mechanism can be implemented that is controlled and administered remotely without recourse to proprietary

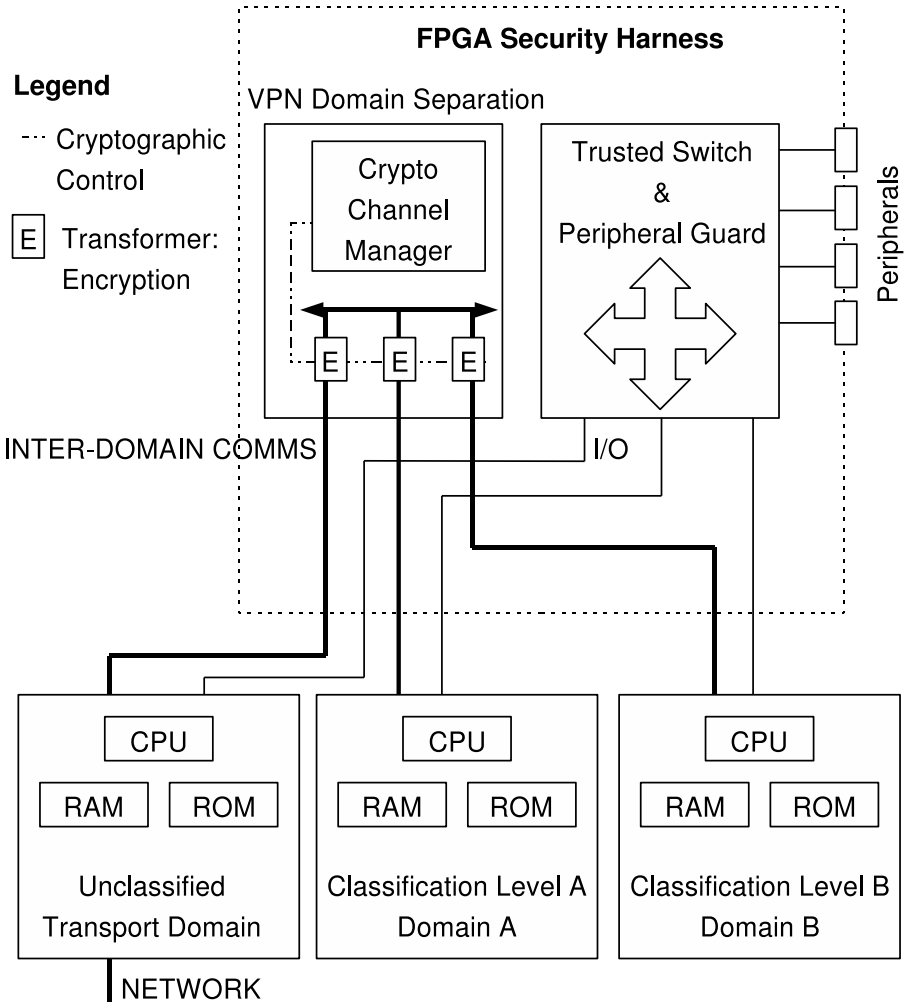

Figure 2: *Annex Minisec Security Harness*

software, or reliance on the integrity of any intervening software or hardware elements between the management terminal and the hard disk.

5.5.2 Digital Video Guard

The Digital Video Guard is an example of a Data Transformer designed to be retrofitted between a COTS computer system and its Digital Visual Interface (DVI) display (Beaumont 2008b). Whilst the Hard Disk Guard is able to protect software distributions and provide policy based management of data and software configurations the intent of the Digital Video Guard is to provide access to confidential data and applications over untrusted networks and hardware installations. Thus, whilst an enterprise computing system may be riddled with Silicon Trojans and software-based malware it is still possible to provide data confidentiality for certain applications by focussing security efforts on the maintenance and provision of a trusted application server and couple this with a cheap

ubiquitous Silicon Security Harness Data Transformer in the form of the Digital Video Guard. Figure 3 depicts a typical arrangement of the guard with a computer system. Interestingly, this guard raises the possibility of retrofitting to quite untrusted computer systems such as those in hotels and airport lounges while still retaining high assurance confidentiality despite the presence of an arbitrary trojan in the main computer system. The reader is referred to Beaumont (Beaumont 2008b) for a proper discourse on how the DVG can be employed to achieve the necessary confidentiality.

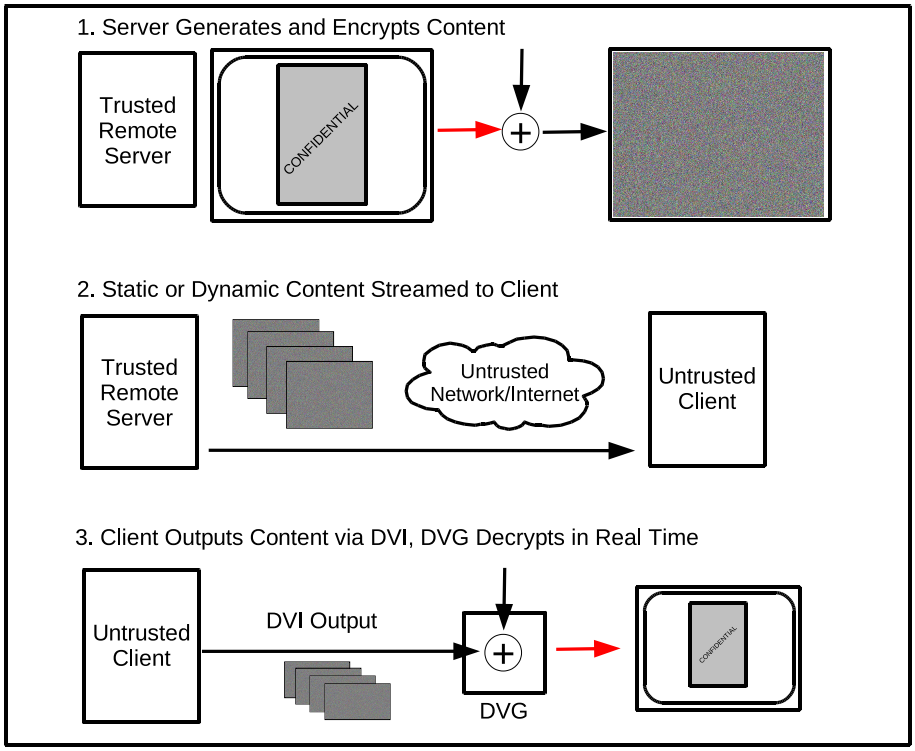

Figure 3: *Digital Video Guard Harness*

5.6 Related Approaches

A further example of harness-like components are some PCI based cards. There are a number of PCI cards that can analyse PCI transactions (VMETRO 2007), perform PCI bus snooping (Klingelhofer 2000), and independently inspect the host computer's system memory (Grand, Carrier 2007). Whilst these cards, and cards of similar nature, may inspect PCI transactions and system memory they are limited to the signature-based and heuristic-based methods employed by intrusion detection systems and virus checkers. In-line security harnesses as described in Anderson (Anderson 1996b) and Grove et al (Grove, et al 2007) have a further advantage of being able to provide some preventative protection measures by scrambling some bus traffic and by providing additional hardware safeguards to reinforce partitioning of memory spaces. However, to operate effectively they must be designed in rather than simply retrofitted. It is important that designs encompassing Silicon Harness safeguards be deployed on network edge devices, such as user peripherals,

and be located close as practical to the user interface. In this way the Silicon Harness components can intercept and protect the device's IO at the very edge of the network.

Unfortunately, none of the preceding discussion can guarantee detection or prevention of an arbitrary Silicon Trojan; McClean and Cohen have seen to that (Cohen 1987) (Cohen 1989) (McClean 1993). However, countermeasures can make the use of a Silicon Trojan fairly difficult for an exploiter.

The application of a Security Harness can target data confidentiality, data integrity, or system integrity and the Security Harness components and techniques will vary depending on the security context and desired security goals. We briefly touch upon some implementation considerations when adding a security harness.

The secrecy surrounding the architecture and placement of the harness system should be maximised as well as any specific parameter settings for the harness. This approach is likely to work best for infrastructure destined for relatively secure locations and since mobile devices are typically not physically secure relying on obfuscation carries more risk.

Parameter monitoring of harness components should include detecting deviations from the monitored component and its manufacturer's specification. A sophisticated monitor should use its own private network to fuse information from multiple components.

Harness components that can condition data streams and constrict transmissions to a canonical protocol representation should be used in a similar fashion to their macro counterparts - Firewalls.

Wherever possible data destined for off chip storage should be encrypted in order to prevent covert storage channel signalling via trusted components. Even simple encryption protocols, in order to enable very fast data throughput, can assist in disrupting a silicon Trojan in a Disk Drive. However, this will need to be weighed against needs which may involve RAID style configurations which may not be easily implementable if scrambling location storage is undertaken.

To ensure the integrity of harness components the secure assembly of components should be maximised.

6 Another Look at the Economics

At first glance, one could be forgiven for thinking that the Silicon Harness approach is not viable; since custom hardware is being used the expense could be enormous. However, let us consider this more carefully. Remember we contend that the main value is in the application and interoperability with other applications rather than the hardware per se, which is offering the same sort of functionality on the same sort of operating system with strong backward compatibility built in.

Commercial providers are now forced into this economic pocket, since any new PC, or mobile device, must be backward compatible with the network and previous versions of operating systems. In the commercial arena we are already seeing a shift away from rapid hardware upgrade cycles, particularly for desktop workstations. Enterprises have slowed their refresh rates for corporate workstations. Instead, they are focussing on constructing

centralised computing clusters, or blade servers, in an effort to virtualise their software environment and service delivery model. This has been done in part to better manage the fast moving software product cycles.

Hence if the National Security concern settles for a slower turnover rate of its user devices (especially its mobile devices), they will still tend to retain compatibility with new business applications. Of more interest should be whether they can run new applications and whether they retain acceptable performance levels. It is this last point which would be a stronger driver for hardware changeover. We contend that this hardware refresh lifecycle for the business enterprise can be slower than for the installed business services (with their never sated need for increasing interoperability with other applications and business functions). It is this latter business driver which results in a requirement to ensure that as much of the software as possible is commodity. New versions of software need to be continually updated and deployed to retain compatibility and support across the enterprise. Hence security has to take into account a plethora of rapidly changing applications, software updates etc. from a plethora of sources, some of which may include entities under the influence of adversaries. This makes security evaluation very difficult at this level. This then leads us to focussing security efforts and in-house investment into areas which we can choose to lower the lifecycle change rate. Mobile devices are one area which can be considered. While the rate of change in the commercial sphere is great, it is typically driven more by smaller form factors and energy issues, rather than significant advances in hardware functionality; new models still run Windows CE, Linux, and Symbian, however they tend to be smaller and run for longer periods than the superseded models. It is this very economic model that has driven the application of the Silicon Security Harness to personal mobile communication devices in the Annex project (Grove, et al 2007).

7 Conclusion

With the advent of outsourced componentry for critical infrastructure and the possibility even of sourcing components for weapon systems from emerging adversaries due to overwhelming economic drivers, new techniques will be needed to be able to build systems where there is a reasonable level of assurance they will operate as intended and in the expectation that the components may carry functionality not in the user's best interests.

Security Architectures will need to be Trojan tolerant, rather than Trojan averse. The Silicon Trojan introduces another layer of difficulty to the security practitioner and security regimes will be needed in areas of device assembly in order to provide a suitable combating methodology. A Silicon Security Harness added to selected hardware devices as part of an ensemble of security measures may be an approach to combating the inevitable rise of the Silicon Trojan.

Utilisation of the Silicon Security Harness implies a reversion to assembly in-country of selected products for high security needs. However, as explained in this paper, the rise in investment importance of the business application suite over user devices at the network edge, and longer "acceptable performance" life cycles of these devices may make

the additional initial investment of this approach feasible as a counter to the threat of the Silicon Trojan.

References

Adee, Sally, (2008) *The Hunt for the Kill Switch* , IEEE Spectrum May 2008

Anderson, M. S, (1996a) *Secure Computer Architecture* , US Patent 6,115,819, 1996.

Anderson, M., North, C., Griffin, J., Milner, R., Yesberg, J., Yiu, K., (1996b) *Starlight: Interactive Link,* acsac, p. 55, 12th Annual Computer Security Applications Conference (ACSAC '96), 1996

Beaumont, M. R., Hopkins, B. D., North, C. J., Yiu, K. K., (2008a) *Hard Disk Guard*, Defence Science and Technology Organisation, DSTO-TN-0827, 2008

Beaumont, M. R., North, C. J., Green, J. D, (2008b) *Digital Video Guard*, Defence Science and Technology Organisation, DSTO-TN-0863, 2008

Bell, D. E. and LaPadula, L. J. (1976) *Secure Computer Systems: Unified Exposition and Multics Interpretation* , MTR-2997 Rev. 1, MITRE Corp., Bedford, Mass., March 1976.

Adam Boileau, (2006) *Hit by a Bus: Physical Access Attacks with Firewire*, RUXCON 2006

Cohen, Fred, (1987) *Computer viruses: Theory and experiments* , Computers & Security, 6(1):22–35, February 1987.

Cohen, Fred, (1989) *Computational aspects of computer viruses* , Computers & Security, 8(4):325–344, June 1989.

DoD 5200.28-STD, (1983) *Trusted Computer System Evaluation Criteria (Orange Book)* , Library No. S225,711, August 1983.

Grand, J., Carrier, B., (2007) *Method and apparatus for preserving computer memory using expansion card* , US Patent, 7,181,560, 2007.

Green Hills Software Inc, (2008) *Integrity Real-Time Operating System* , (Retrieved 2008) <http://www.ghs.com/products/rtos/integrity.html>

D. A. Grove, T. C. Murray, C. A. Owen, C. J. North, J. A. Jones, M. R. Beaumont, B. D. Hopkins, (2007) *An Overview of the Annex System,* acsac, pp. 341–352, Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007), 2007

Govindavajhala, S., Appel, A.W., (2003) *Using memory errors to attack a virtual machine* , IEEE Symposium on Security and Privacy, 2003.

Intel Corporation, (1994) *FDIV Replacement Program* , (Retrieved 2008) <http://www.intel.com/support/processors/pentium/fdiv/>

Interlink Enterprise Computing, Spam Daily News, (2006) *Israeli hackers charged with developing industrial espionage spyware*, <http://www.spamdailynews.com/publish/Israeli_Trojan_horse_developers_indicted.asp>

King, Samuel T., Tucek, Joseph., Cozzie, Anthony., Grier, Chris., Jiang, Weihang., Zhoue, Yuanyuan., (2008) *Designing and implementing malicious hardware* , LEET 08 – First USENIX workshop on large-scale exploits and emergent threats.

Klingelhofer, Marc E., (2000) *Computer system employing a bus snooping multimedia subsystem for implementing video multicast transactions* ,US Patent 6,026,218, 2000.

KPMG LLP, (2003) Study for Alliance for Grey Market and Counterfeit Abatement (AGMA), *Claims the value of the Grey Market is $40 billion per year.* <http://www.agmaglobal.org/press_events/press_docs/KPMG_TheGreyMarket_Web.pdf>

Kumagai, Jean (2000) *Chip Detectives* IEEE Spectrum, November 2000 pp43 - 48

McClean, J, (1993) *Integrating Specifications, Integrating Assurances* in Proceedings, 16th National Computer Security Conference, September 20-23, 1993. pp 355-357

McCullough, D. (1987) *Specifications for Multi-Level Security and a Hook-up Property,* in Proc. 1987 IEEE Symposium on Security and Privacy, IEEE Computer Society Press, April 1987.

Markoff, J., (2007) *Chinese Seek to Buy a U.S. Maker of Disk Drives* , New York Times, 25 August 2007.

Metcalfe, Robert R., (circa 1980) *Metcalfe's Law.* The systemic value of compatibly communicating devices grows as the square of their number.

Microsystems Technology Office, Defense Advanced Research Projects Agency, (2007) *DARPA Request for Solicitations BAA 07-24: Trust in Integrated Circuits* , 7 March 2007. <http://www.darpa.mil/MTO/solicitations/baa07-24/index.html>

National Institute of Standards and Technology (NIST), Information Technology Laboratory, (1983) *Rainbow Series* , <http://csrc.ncsl.nist.gov/publications/secpubs/rainbow/>

Nicely, Thomas, (1994) *Pentium FDIV flaw FAQ*, 1994, <http://www.trnicely.net/pentbug/pentbug.html>

O'Dowd, Dan, (2004) *Linux Security: Unfit for Retrofit* , Green Hills Software Inc. White paper, EE Times April 19,2004

Office of the Under Secretary of Defense For Acquisition, Technology, and Logistics (OUSDATL), (2005) *High Performance Microchip Supply* – Defence Science Board Task Force, February 2005. <http://www.acq.osd.mil/dsb/reports/2005-02-HPMS_Report_Final.pdf>

Schwartau, Winn, (1994) *Information Warfare: Chaos on the Electronic Superhighway*, Thunders Mouth Press, New York, 1994.

Smith, Richard E., (2000) *Trends in Government Endorsed Security Product Evaluations*, Secure Computing Corporation. Proceedings of the 23rd National Information Systems Security Conference. A B1 evaluation adds US$2.5 million (1990s) to the cost of a product; CC EAL4/ITSEC EAL4 added about US$1 million at the time.

The National Information Assurance Partnership (NIAP), (2008) *The Common Criteria Evaluation and Validation Scheme* , (Retrieved 2008) <http://www.niap-ccevs.org/cc-scheme/in_evaluation/>

VMETRO Inc., (2007) *Vanguard PCI Bus Analyzer, Exerciser, Protocol Checker & Compliance Checker for PCI and PCI-X* , (Retrieved 2008) <http://www.vmetro.com/category3706.html?visitedlp=true>

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA	1. CAVEAT/PRIVACY MARKING

2. TITLE	3. SECURITY CLASSIFICATION
Towards Countering the Rise of the Silicon Trojan	Document (U) Title (U) Abstract (U)
4. AUTHORS	5. CORPORATE AUTHOR
M.S. Anderson, C.J.G. North and K.K. Yiu	Defence Science and Technology Organisation PO Box 1500 Edinburgh, South Australia 5111, Australia

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DSTO–TR–2220	AR–014–344	Technical Report	December, 2038

8. FILE NUMBER	9. TASK NUMBER	10. SPONSOR	11. No OF PAGES	12. No OF REFS
-	N/A	N/A	22	33

13. URL OF ELECTRONIC VERSION	14. RELEASE AUTHORITY
http://www.dsto.defence.gov.au/corporate/reports/DSTO–TR–2220.pdf	Chief, Command, Control, Communications and Intelligence Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved For Public Release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SOUTH AUSTRALIA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS

No Limitations

18. DSTO RESEARCH LIBRARY THESAURUS

Computer Security, Silicon Trojan, Annex, Computer Network Attack, Computer Network Exploitation, Implant

19. ABSTRACT

The Trojan Horse has a venerable if unwelcome history and it is still regarded by many as the primary component in Computer Network Attack. Trojans have been the direct cause of significant economic loss over the years, and a large industry has grown to counter this insidious threat. To date, Trojans have in the vast majority taken the form of malicious software. However, more recent times have seen the emergence of what has been dubbed by some as the "Silicon Trojan"; these trojans are embedded at the hardware level and can be designed directly into chips and devices. The complexity of the design of the device or chip in which they are embedded, coupled with the severe difficulty of evaluating increasingly dense, proprietary hardware designs, can make their discovery extremely difficult. This paper explores the possible effectiveness of a Silicon Trojan, whether they form a credible ongoing threat, and describes possible approaches which can be used as countermeasures.