Why Hardware?

Special-Purpose Hardware Prevents Cybersecurity Attacks

J. Cox

9 September 2019

Abstract

Cybersecurity attacks are increasingly prevalent and many products have become available for detecting such attacks. However, few products can be found that prevent these malicious attacks. In these pages we examine the advantages of using special-purpose hardware to protect the confidentiality and integrity of data. This protection includes encryption of data, authentication of the data source, verification that the data have not been corrupted and have been transmitted over explicitly authorized links. Only the cybersecurity functions of encryption, authentication, authorization, integrity and no other functions are built into the special-purpose hardware. It's impossible to successfully execute an attack from an untrusted endpoint because there's no memory that can be altered to store and execute malware. The same functions can be realized using a general-purpose computer, where its memory and CPU would store and execute the appropriate software. However, using a general-purpose computer risks harmful malware attacks that might compromise the computer, whole networks and ultimately even the nation's critical infrastructure. The universal nature of general-purpose computers makes them uniquely vulnerable to attack while special-purpose hardware remains safe, even from attacks by the most determined adversary operating from any untrusted endpoint.

To explain to the non–technical reader the difference between a general-purpose computer and special-purpose hardware, we employ a whimsical analogy between a medieval castle and a general-purpose computer. The castle walls do not thwart the clever brigands (malicious network packets) who enter through nooks and crannies to reach the castle's inner workings. Once inside they practice their mischief beyond the eyes of the castle's chamberlain (operating system). However, a knight (special-purpose hardware), whose only wish is to thwart the brigands before they reach the castle's drawbridge (network interface controller), provides trustworthy protection. The castle's chamberlain, no matter how clever, cannot provide the same protection while performing his many duties. In contrast, the knight does one thing and one thing only.

This analogy will not satisfy the more technically oriented reader. Accordingly an analysis of the cybersecurity protection offered by general-purpose computers and by special-purpose hardware is presented in a final section of this document. A general-purpose computer cannot be shown to be provably safe from an attack launched by an untrusted endpoint without enforcement of strict control over all its executable code. The placement of all cybersecurity functionality in separate, special-purpose hardware can be proven to be safe from such an attack. It has other advantages too. Cybersecurity protection can be dropped into existing fixed-endpoint networks with ease and without the need for added software or the replacement of existing hardware.

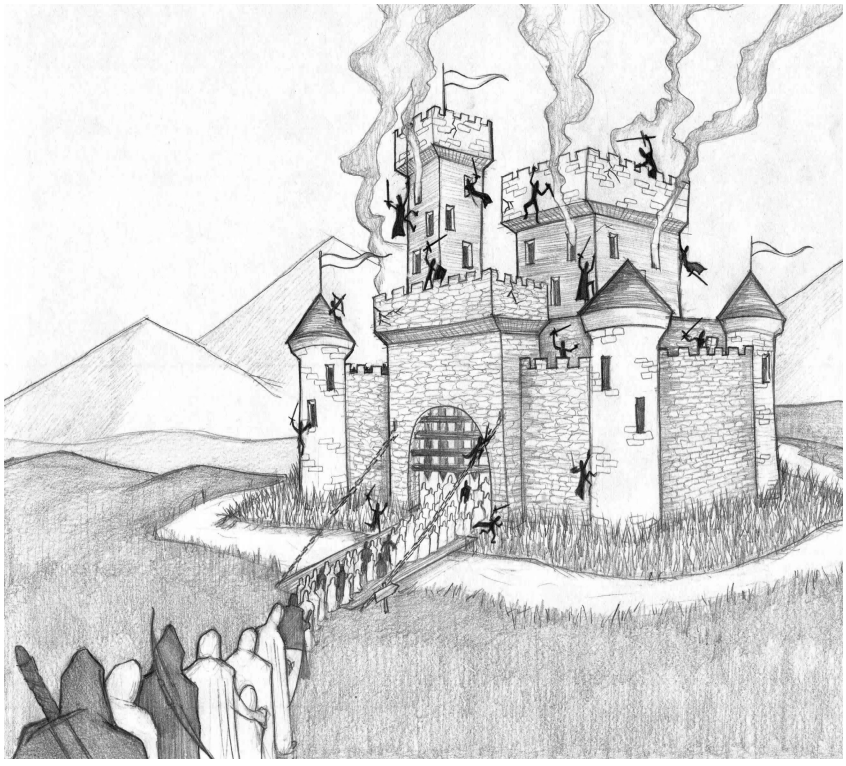
In summary, special-purpose hardware can be proven to be safe from attack by untrusted endpoints and, in addition, cybersecurity functionality can be separated from the many other tasks required of a general-purpose computer. This makes it much easier to understand the level of security exhibited by a computer and its network.

Analogy

It may be helpful to think of your computer as your castle, one that has a long line of visitors each carrying packets of information from other castles, near and far. Some of these visitors have harmful intent and your operations and security minister, the lord chamberlain, checks them all at his desk just inside the drawbridge. However, some of the attackers avoid this check and look for vulnerabilities throughout the castle and its walls. Inevitably, a window or door has been left ajar and the attacker enters masquerading as a steward (data). Castle servants, who receive instructions from such an imposter, may view him as a legitimate steward with the result that the attacker has effectively taken control of one or more functions of the castle.

The following sketch illustrates this situation in a whimsical manner, but it is a provocative mental analogy suggesting events that can happen in a general-purpose computer. In fact, imagine that the castle (general-purpose computer) controls a portion of the kingdom's critical infrastructure or transfers money to and from neighboring castles. Wherever cybersecurity is a high priority and activities are managed by a general-purpose computer, this analogy suggests that there is an opportunity for cybercriminals.

Your Computer is a Castle
Attackers are Everywhere

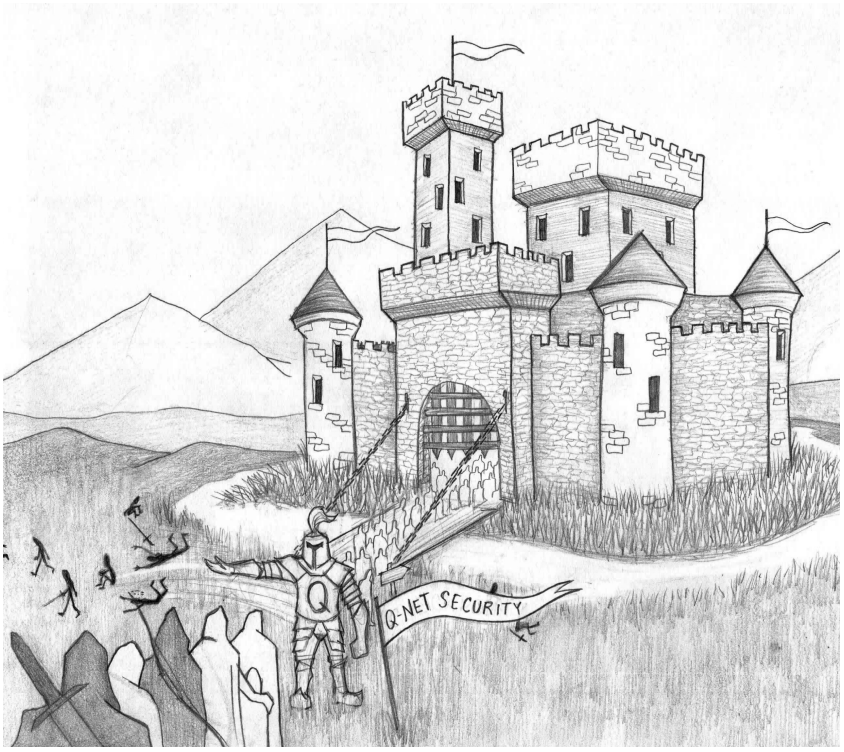


Art provided by Chris Rau Art

There is a moat around the castle, but it is of little value if the attackers are allowed across the drawbridge to be screened by the chamberlain. The drawbridge is analogous to your network interface controller (NIC) and once across the moat the attackers can find a poorly protected nook or cranny and corrupt, destroy or encrypt the castle's records. They can also

exit carrying valuable records via the rear gate marching along with all the benign visitors and emissaries that you have sent to do your bidding elsewhere in the kingdom.
Instead of allowing the visitors and their packets inside the castle walls, it is better to check them as they approach the moat. As the sketch below suggests, there is a Q-Knight stationed there that has been selected because of his obsessive, compulsive demeanor. He insists on a thorough check of all incoming travelers. The Q-Knight's duties are completely different from those of the chamberlain who performs a myriad of functions and acts as the castle's operating system. The Q-Knight does one thing and one thing only: authenticate and authorize visitors.

Your Computer is a Castle
Protect it with Q-Net Security



Art provided by Chris Rau Art

Thus, in our whimsical analogy the OCD-tainted Q-Knight stationed at the entrance to the drawbridge carries out these authentication and authorization checks. No attacker can outwit the Q-Knight and all visitors that fail the checks or arrive without credentials from a trusted castle are cast aside. The moat keeps the ejected attackers from reaching the castle wall.

In the real world, it is best to employ special-purpose hardware, whose only job is to authenticate all incoming traffic, make sure each packet is authorized and flawless. Only then will the payload be decrypted and transferred into your computer. Q-Net Security carries out this agenda outside your computer using special-purpose hardware that is designed for that function alone. It is called the Q-Net Input-Output Unit (QIO) and this smartphone-sized device performs a fixed sequence of steps swiftly without any possibility of compromise.

Analysis

The analogy described above may be helpful to some readers, but to others it will not be convincing. To satisfy these others, it is necessary to understand the pivotal difference between special-purpose hardware and general-purpose computers. It is also advantageous to consider a relevant theoretical computer science result.

There are many examples of early applications of special-purpose hardware: the Jacquard loom, the Babbage Difference Engine, punch card sorting machines and the code-breaking Colossus of WWII. These machines were all designed for a single job. The listed machines all predated Eniac, the first general-purpose computer, which was launched at the University of Pennsylvania in 1945. Since then, because of their computational potential, general-purpose computers have eclipsed special-purpose hardware. Their ability to interweave instructions and data introduces limitless flexibility and unbounded computational variety, a result that was proven theoretically by Alan Turing in 1936. However, this enormous computational blessing comes with the unforeseen cybersecurity curse: the potential for inadvertent execution of malware delivered by attackers located elsewhere on the computer's network.

To explain this fundamental vulnerability that resides in a general-purpose computer, we must understand that both the lists of instructions (software) and the data to be processed are stored in memory in a way that makes them intrinsically indistinguishable from each other. They both are really just numbers and the same number can serve as an instruction or as a data value depending only on the circumstances of its use. The processing engine (CPU) in a general-purpose computer has no foolproof way to tell the difference. If the CPU reads a data value when expecting an instruction, it will unfailingly execute the instruction that corresponds to that data value. This interweaving of control and data is the weakness that attackers employ to accomplish their mischief.

Careful design and coding of the computer's operating system (OS) can help to protect against this kind of attack. For example, a microkernel (seL.4) has been formally proven to satisfy the classic security properties of data integrity and confidentiality. However, trouble occurs as systems grow in complexity or evolve over time. For example, designers are challenged to maintain cybersecurity when their designs grow to include multiple threads, multiple levels of memory and multiple cores. Despite general improvements in the security of recent OS versions, many users have lingering doubts that may lead them to neglect an upgrade even when it is beneficial to security to do so. However, new code and new functionality will always lead to some increased risk, even though that risk may be decreasing with time as the industry gets better and better at writing more secure software. However, the more executable lines of code there are, the greater the potential for security lapses. Beyond such software flaws, multiple benign application programs can interact to provide the foothold that an attacker uses to execute harmful code. Recently long dormant hardware design lapses (SPECTRE and MELTDOWN) have been found to provide subtle access for attackers wishing to breach confidentiality. With tens of billions of transistors and tens of millions of lines of code in an endpoint computer, policing the trustworthiness of every possible interaction between every instruction and every data value is unattainable.

But why not put all the necessary cybersecurity protection into a program that is installed on your computer and then find a way to prove your computer is safe against a remote attack?

To explain why that is an impractical idea we need to return to Alan Turing's pivotal 1936 paper. That paper proved the existence of what we now call general-purpose computers, computers that were all equivalent to each other in the sense that they were universal in their ability to eventually calculate anything that is computable. Turing's 1936 paper also proved an important result known as the *Turing halting problem*, the problem of determining whether an arbitrary computer program with an arbitrary input will finish running (i.e., halt) or instead run forever. Turing showed that such a question was undecidable.

By building on the work of Turing, Kirkpatrick[1] has recently shown that computer security within a general-purpose computer is algorithmically intractable. Even by comparing a suspected operating system with an authentic copy, the task of locating the first byte of foreign machine code that might execute was shown to be undecidable. Thus, to prevent attacks with certainty requires meticulous control of all input data, web interactions and downloads. Normally, this degree of oversight is unacceptable in all, but the most sensitive of applications.

Instead of trying to patch the many vulnerabilities associated with a general-purpose computer, the cybersecurity manager should consider the use of special-purpose hardware. The Q-Net Security QIO product employs silicon that encrypts, decrypts, authenticates and authorizes all communication and, in addition, includes important functionality to generate and manage all packets and keys.

Special-purpose hardware, such as the QIO, that executes no other functions than those required for cybersecurity protection, can never be breached by packets sent from an untrusted location. The QIO cannot be programmed or reprogrammed to perform any actions other than the specified cybersecurity functions. A breach is impossible even when the attacker spoofs the packet's source-address because of the potent authentication provided. Furthermore, the QIO has no memory for the storage of malware instructions and the confidentiality of all exported data are automatically protected by strong encryption. Thus, protection against attacks from an untrusted endpoint can be proved and the confidentiality of exported data is always protected.

Contrast the ease of use and the provable safety against attack provided by the QIO with the absence of those qualities in a general-purpose computer. With the QIO the user can obtain cybersecurity protection without change to habits or the replacement of legacy equipment. The proof of safety against remote attack lies in the immutability of the microcircuit wires that determine the sequence of QIO cybersecurity operations including:

- True random key generation
- Frequent and reliable delivery of keys to authorized destinations
- Cryptographic wrapping of keys for delivery
- Payload encryption
- Payload decryption
- Authentication of sources
- Authorization of transactions

The QIO performs all these operation without significant impact on link bandwidth, link latency or endpoint throughput. Ideal applications for the QIO involve machine-to-machine communication between endpoints connected by IP networks such as an ICS/SCADA

[1] Brent Kirkpatrick; https://www.intrepidnetcomputing.com/security/whitepapers/undecidable.pdf

network. Endpoints can be located anywhere in the world. The transport medium can utilize private networks, public networks, copper, optical fibers, wireless links or any combination thereof. If the network successfully transported traffic between the fixed endpoints before installation of the protecting QIOs, it will do so after their installation. The QIO protection drops in without software updates, without any changes to existing network links and without modification of connected equipment.

The major advantages of Q-Net Security approach are:

- After installation of a QIO, an endpoint can be proven to be protected against malware attack.
- Distributed Denial Of Service (DDOS) attacks are blunted because the QIO deletes these troublesome packets and thereby protects the endpoint from harm.
- No changes or additions to an endpoint's legacy-code are required. No modification of existing equipment is needed.
- The external placement of the QIO protects, but does not intrude on the endpoint and so does not reduce the endpoint's processing capability.
- The cryptographic functions of key-generation, key-management, encryption, decryption, authentication and authorization are carried out faithfully and privately in the QIO. This is in contrast to a general-purpose computer, where a determined adversary can often find ways to access these essential cryptographic functions.

In summary, we propose that the current perimeter-based (firewall) approach to establishing cybersecurity protection be abandoned in favor of protection placed at each fixed endpoint. The Q-Net Security solution is a perfect way to protect connections between data-centers, between individual nodes in data-centers and between ICS/SCADA nodes in factories, offices and at the control points of our nation's critical infrastructure. Our approach takes advantage of the immutability of function provided by the QIO's special-purpose hardware. The inevitable vulnerability of a general-purpose computer is thereby completely avoided. Finally, the strong protection provided by the QIO drops into a properly functioning IP network smoothly without change to existing hardware and software.