Bionic Buffalo Tech Note #110:

Concept for a Secure Network Computer

last revised Tuesday 11 January 2000

©2000 Bionic Buffalo Corporation. All rights reserved.
Tatanka and *TOAD* are trademarks of Bionic Buffalo Corporation.

BACKGROUND

Proprietary or confidential information resources, delivered by network to a restricted audience, are highly vulnerable to interception and attack at the point of delivery.

Cryptography can protect data in transit and storage, but such data eventually must be decrypted to be useful. Potential points of attack include the identification and authentication process, the cryptographic mechanism itself (with attendant keys), display and entry operations, transient or temporary storage and cache locations, and the exploitation of hardware and software failures. Defence against attack requires a system known to be secure.

A secure system must be of verifiable design, and constructed of verifiable components in a verifiable process. If all secure systems were identical (except for configuration and keys), then economies of scale would allow reasonable costs in spite of the expenses of verification. However, useful systems ought to execute a variety of insecure, unverifiable application programs.

In order to allow these potentially dangerous programs to run, their operation must be contained in such a way that they cannot compromise the security of the system.

This Tech Note describes a Secure Network Computer (SNC), which attempts to control information access while allowing the limited use of unverified software.

MODEL & TERMINOLOGY

Sensitive, proprietary, secret, or other valuable information may be restricted to one or more *domains*. The domains are logical concepts used to contain the information. As examples, a domain may contain:

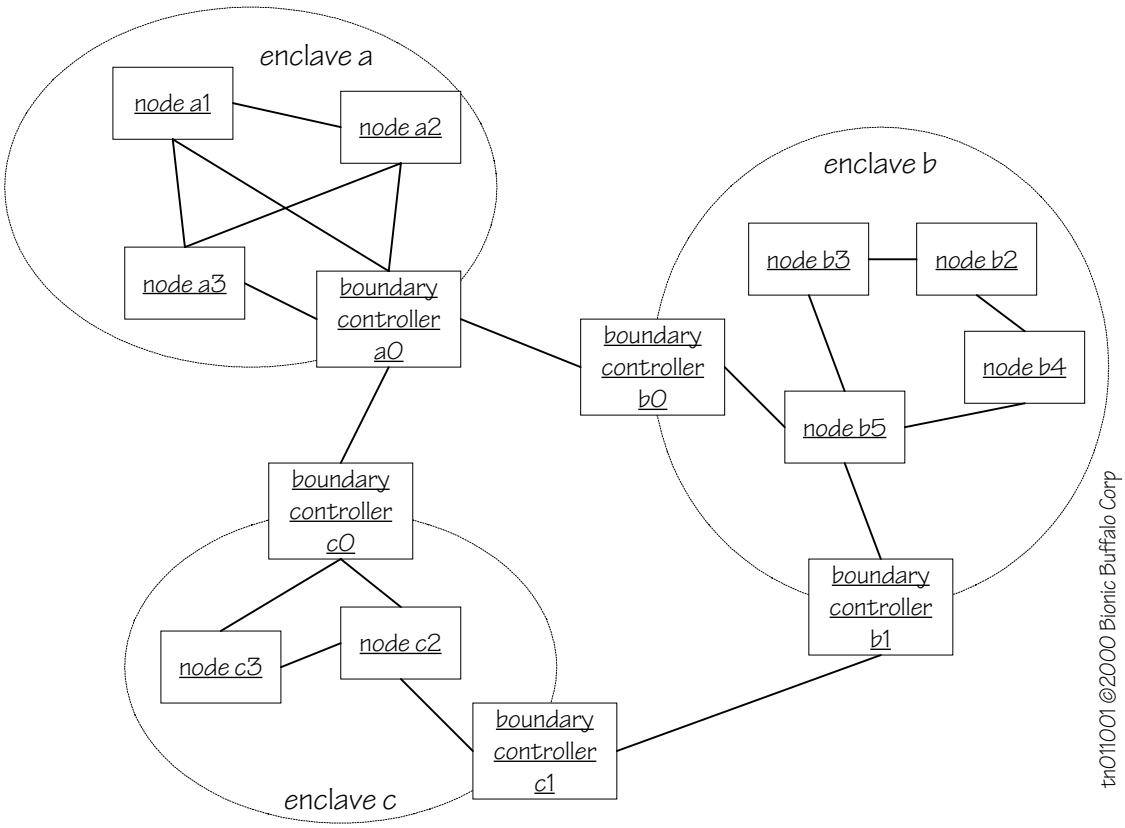
- military battle plans
- hospital patient records
- employee salary and benefit information

- citizens' tax returns

Domains may be contained within other domains. (Each department's salaries may be a separate domain, with all company salaries in a superdomain of these subdomains.) The primary goal of this organization is to protect the information in a domain from unauthorized access. To this end, each user must be authorized to have (possibly limited) rights to access specific domains, and must be prevented from inappropriate access to other domains. Furthermore, information leakage from one domain to another must be prevented.

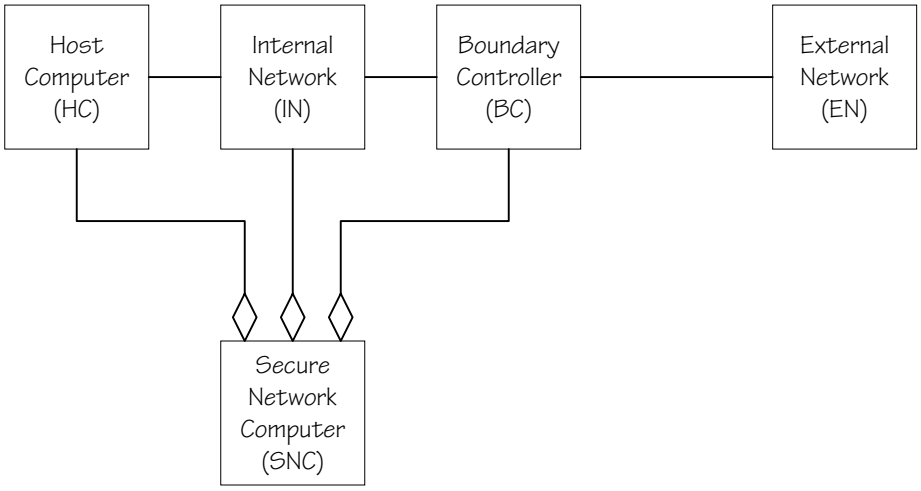
The physical realization of a domain is a set of *enclaves*. An enclave is an arena for information, protected by boundary controls, so that information cannot improperly enter or leave the enclave. In computer terms, an enclave might be a physically secure local network (LAN) protected by firewalls. An enclave may also consist only of a single node, which must have internal boundary controls if part of a larger domain.

The enclaves in a domain are connected to one another through their boundary controllers. The connection mechanism for a computer-based domain might be through private or public networks, with appropriate levels of physical and informational (cryptographic) security.

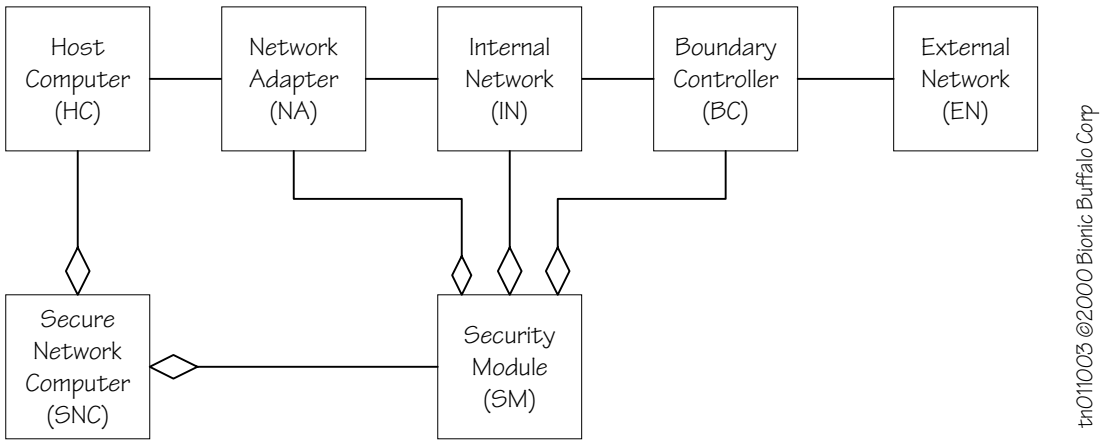


©2000 Bionic Buffalo Corporation; All Rights Reserved.

This Tech Note describes the SNC: a single-node enclave intended for connection to a larger information domain. As a minimum, the SNC contains three components: the Boundary Controller (BC), an Internal Network (IN), and a Host Computer (HC). The BC enforces the boundaries of the information domain. The HC provides a safe execution environment for unverified programs.



The IN may exist physically, or it may be only logical. If the IN and BC are constructed as a single physical module which connects into the HC's i/o bus (emulating a network adapter card), then the IN can be purely logical. Denote this single physical module (combining HC network adapter, IN, and BC) as the *Security Module* (SM).



The managers of a domain may restrict access only to qualified parties, who might be required to use only qualified equipment. For example, a qualified party might be one who has passed some background check, and qualified equipment might be required to meet certain standards for resistance against tampering or sabotage. The secure network computer described in this document is intended to meet such standards, allowing it to qualify for connection to secure domains.

©2000 Bionic Buffalo Corporation; All Rights Reserved.

An important architectural feature of the SNC described in this Tech Note is that no single failure will compromise the security of the system. The meaning of "failure" is taken broadly, to include hardware and software design and implementation errors, as well as hardware malfunctions.

Since the software running in the HC is not verified, it must be considered hostile. It must be presumed that HC applications and operating systems might be malicious, and may attempt to breach the security of the SNC by leaking information into or out of an information domain. The architecture of the SNC must prevent such leaks.

NETWORK ARCHITECTURE

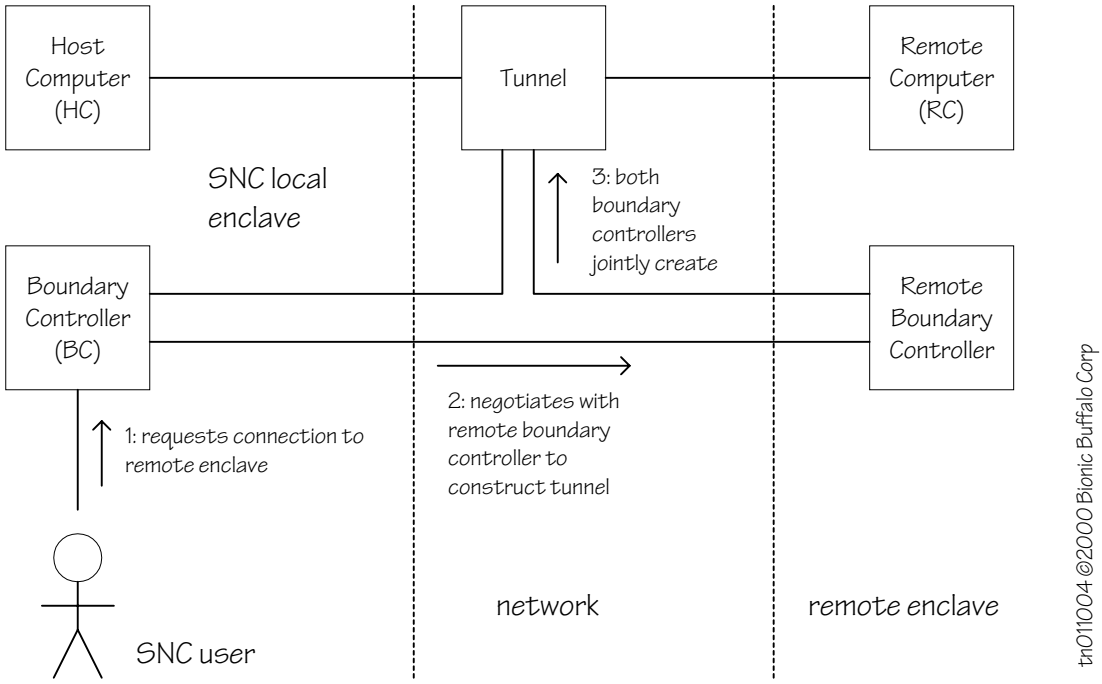
Even when there is no expectation that a network will be shared with other parties, secure connections among enclaves in a domain will typically use virtual private network (VPN) technologies. VPN protocols use certificates, cryptography, and signing processes to assure the integrity and impenetrability of traffic, whether or not the links are shared with other traffic.

The leading candidate protocol for a VPN of the sort required by the model described above is IPsec, which encapsulates ordinary IP traffic within secure connections. The details of IPsec are beyond the scope of this paper, but are well documented elsewhere. (Please refer to the Bibliography at the end of this paper.) IPsec allows the creation of a *tunnel* between the boundary controllers of different enclaves.

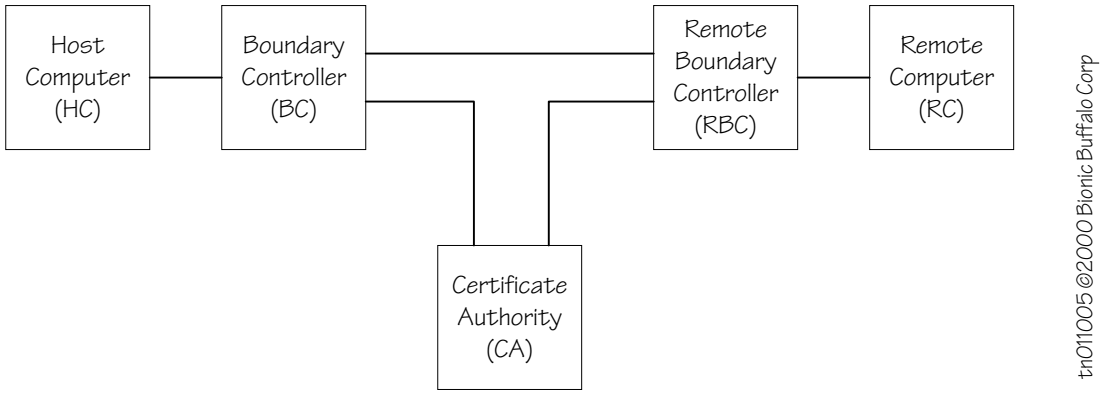
The BC permits the HC to send and receive network traffic only through the tunnel, enforcing the isolation of the information domain from other domains. The BC has access to and from the remote enclave only, and not to any other nodes of the network. To maximize the number of programs (and operating systems) which may run on the HC, operation of the network is transparent to the HC: at least one of the nodes within the remote enclave appears local to the HC. (The "local" node may be a gateway to the enclave's other nodes, or all of the enclave's nodes may be local, and on the same subnet with the HC.)

Prior to establishing a VPN connection, the boundary controllers must agree on which keys are to be used for the session. This process is known as *key exchange*, and requires that the parties identify themselves so that authorization to make the connections can be ascertained. (Certifying the identity of users is known as *authentication*.)

©2000 Bionic Buffalo Corporation; All Rights Reserved.



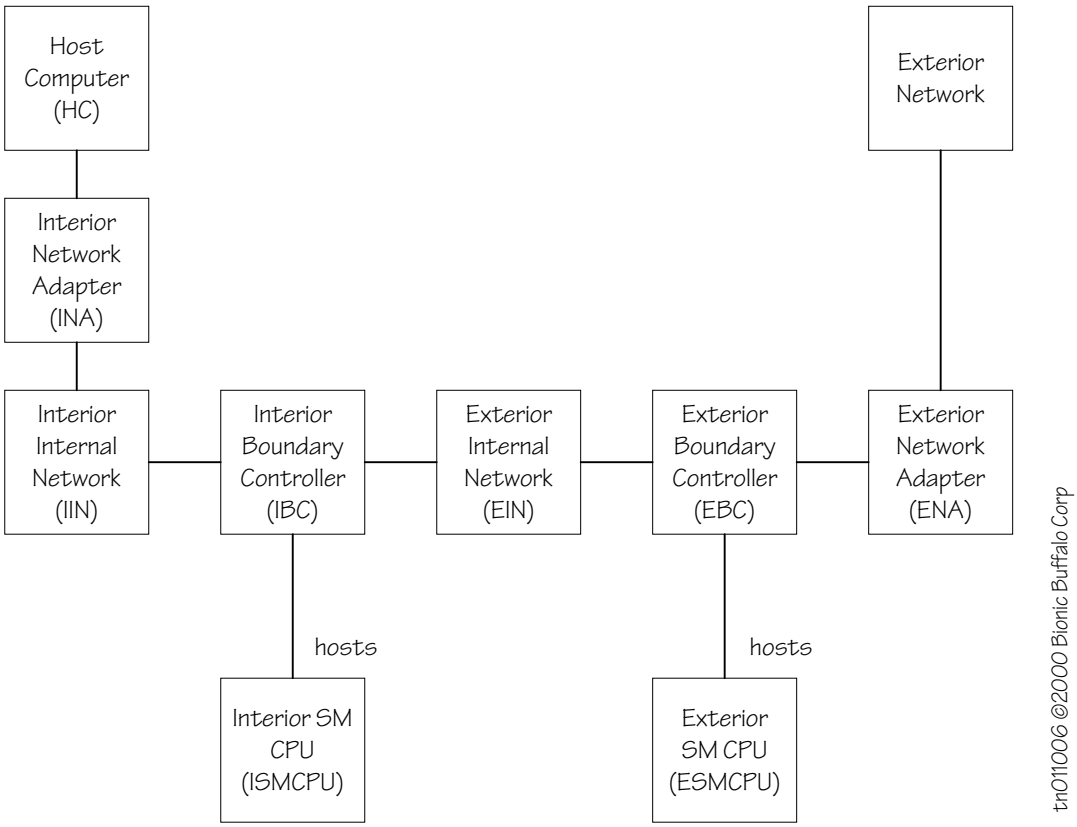
Authentication may involve dynamic reference to one or more *certificate authorities* (CAs). Various protocols (such as LDAP) are used to communicate with the CAs, and these protocols may themselves be encapsulated within IPsec. In general, a CA may be found within a boundary controller of the enclave to which connection is to be made, or within a separate node on the network.



In the SNC, all of these negotiations are handled by the BC. The HC has no role until after the tunnel is created.

To prevent a single failure from allowing a security breach, the BC is implemented as two separate BCs, each checking the work of the other. A separate CPU hosts each BC, and the two CPUs are connected in series. Even if one BC allows passage of inappropriate traffic, the second BC will block the inappropriate messages. This requires two CPUs in the SM, and two internal networks.

©2000 Bionic Buffalo Corporation; All Rights Reserved.



Protocol stacks are complex, and are rarely entirely free of errors. To reduce the possibility of software errors (or sabotage) causing security breaches, separate teams can program each BC, and each CPU can run a different RTOS and protocol stack. Using dissimilar CPUs can guard against mask or other implementation errors.

CLEARING THE HOST COMPUTER

A given SNC may be used to access more than one information domain. In order to prevent information from leaking inappropriately into or out of a domain, the memory in the HC must be cleared between connections to different domains. In this context, "memory" is taken to include all cache and register locations in the HC.

A simple procedure to zero memory, cache, and registers, or to write patterns to them, does not comply with the design goal that no single failure can be allowed to compromise the integrity of the system. No matter how elaborate, such a scheme is susceptible to implementation errors in the HC, and may also be vulnerable to hostile, unverified software running in the HC.

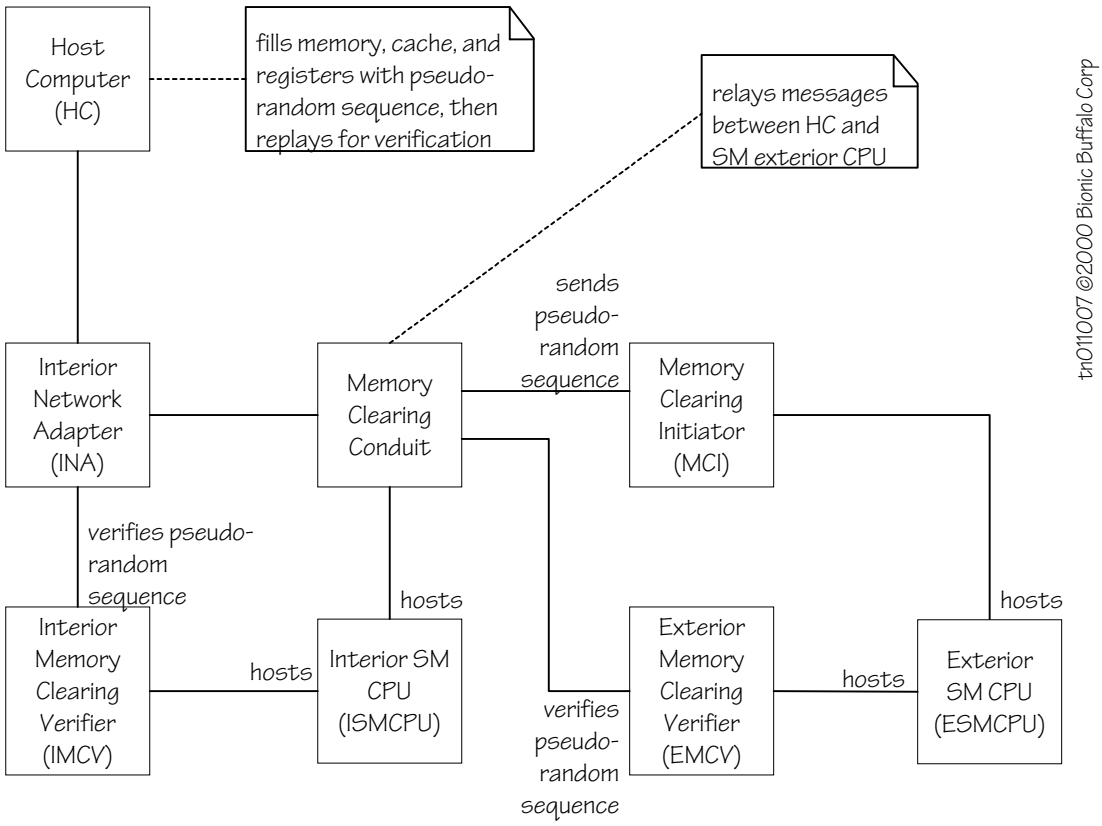
©2000 Bionic Buffalo Corporation; All Rights Reserved.

An effective procedure to verify that HC memory, cache, and registers are cleared, is to present a cryptographically strong, pseudo-random number sequence to the HC. This sequence must be long enough so that it will occupy all of the memory, cache and register storage capacity of the HC. Then the HC must be able to replay the identical sequence back to its originator, showing that the entire capacity of the HC was cleared by being filled with the pseudo-random number sequence.

The Memory Clearing Initiator (MCI) creates the pseudo-random sequence, and correct replay is verified by the Interior and Exterior Memory Clearing Verifiers (IMCV and EMCV). Both the MCI and both MCVs are external to the HC, and may conveniently be located within the SM. There are two MCVs, to prevent the failure of either one from compromising the security of the SNC. Each MCV is hosted on a different CPU. The ESMCPU hosts the MCI.

The two SM CPUs must be connected in series between the HC and the external network. Otherwise, either CPU might fail and pass messages between the HC and the external network, without the other CPU to act as a check. Therefore, the interior CPU must include a conduit between the HC and the exterior CPU.

The INA used for the HC's network traffic can also serve for communications related to memory clearing. If the INA is implemented with direct memory access (DMA) to the HC, then the ISMCPU can examine HC memory directly to verify clearing. The converse should be forbidden: it should be impossible for the HC to access SM memory, except perhaps for HC-ISMCPU message buffers whose contents must be modified or viewed by the HC in any case.



©2000 Bionic Buffalo Corporation; All Rights Reserved.

If the HC's memory clearing operation fails, one or both of the MCVs will detect the failure. If one of the MCVs fails, then the other MCV will detect the failure of its companion. If the memory capacity of ISMCPU exceeds that of HC, then the EMCV may be required to fill ISMCPU memory during the test, to guarantee that IMCV does not sabotage the test by spoofing (pretending to be the HC).

AUTHENTICATION AND USER INTERACTION

Before an enclave's boundary controller will grant the SNC admission, both the user and the SNC must be authenticated. Each must possess an appropriate private key and certificate. Similarly, the boundary controller itself must be authenticated by the SNC, else a connection might be made to a counterfeit enclave.

The SNC's own certificates and keys can be stored within the SM, since it is the SM's CPUs which will use them. The user's keys and certificates, on the other hand, are usually kept in an identification token which is carried by the user.

The most common types of identification token are smart cards and smart buttons. These devices contain their own CPUs and software, and (in some cases) their own power sources (batteries) and clock-calendars. Their main functions are:

- to sign documents, using their secret keys
- to verify signatures, using the certificates they contain

Identification tokens are designed to be resistant to tampering and observation. If they secret key can be read from such a device, then the device could be cloned and would not be a reliable form of identification. If the clock-calendar were to be changed, then the expiration dates on certificates could be avoided. Since the token's CPU must know the keys and certificates to sign and verify, the CPU itself must also be resistant to tampering and observation.

Personal identification tokens usually will refuse to perform their functions unless their owners give them additional information. This additional information might be a secret password or number, or it might be a digest of the owner's scanned retina or fingerprint. This precaution makes it more difficult to impersonate a user by stealing his or her identification token.

When a remote boundary controller wants to verify the identity of a user, it will send an arbitrary document to be signed by the user's token. The token, in turn, will expect the user to provide the additional secret information (password, number, or scan). Then the token will sign the document and return it to the remote boundary controller for signature verification.

©2000 Bionic Buffalo Corporation; All Rights Reserved.

In general, the SNC requires an input device to acquire the additional information required by the user's identification token. The input device might be a keypad, fingerprint scanner, or other such device, or may be some combination of these. In addition, the user must be prompted during operation, so a display device (such as a liquid-crystal display) must also be provided.

If the SNC is configured to offer a choice among enclaves for connection, then a keypad or scanner may also be required for enclave selection. Thus the input and display device (or devices) might serve several purposes.

A natural idea is to use the HC's keyboard and display for these purposes. There are several problems with this idea:

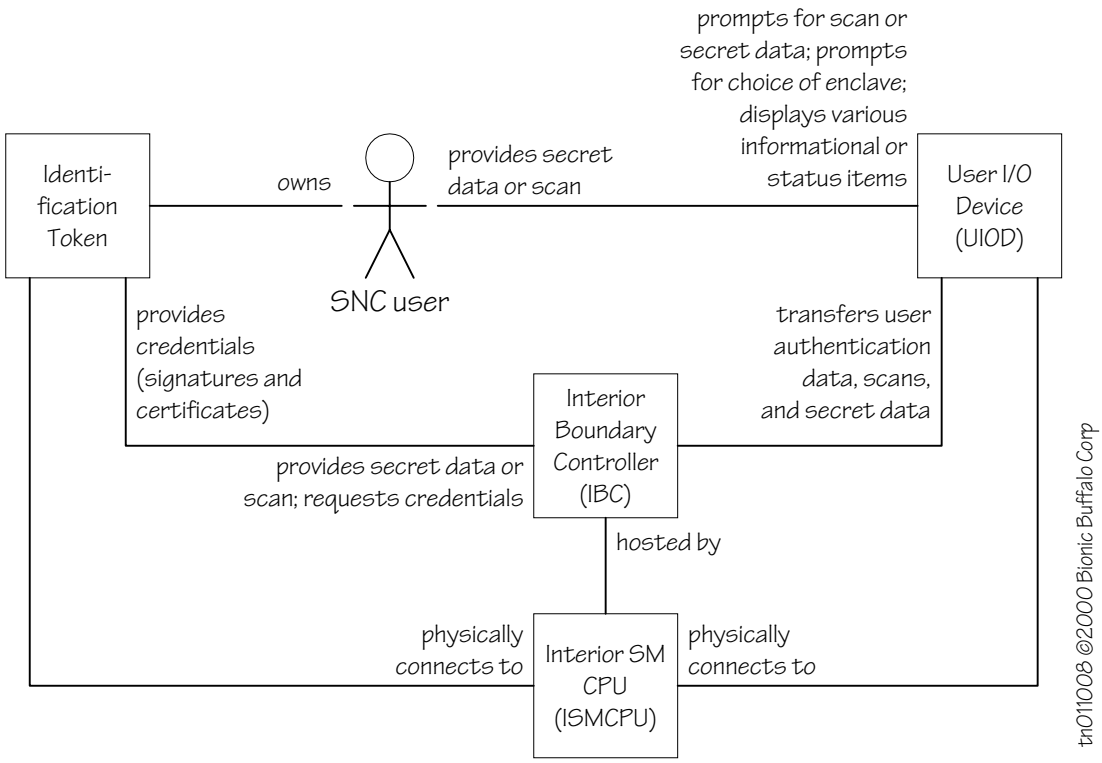
- to allow a variety of off-the-shelf programs to run on the HC, it cannot be presumed that such programs will be programmed to share their keyboard and display with other computers

- malicious software on the HC might be programmed to steal a user's secret password or number

- the required input and output devices may not be compatible with standard keyboards and displays used in the HC

Although these are not insurmountable problems, the architecture presented here assumes that a separate user i/o device is more practical and cost-effective than sharing the HC's peripherals. Of course, an implementation may decide to tackle these issues, and make a different design choice.

©2000 Bionic Buffalo Corporation; All Rights Reserved.



The user i/o device is not redundant in this architecture, nor is the identification token itself. If the i/o device were to fail, then it would be impossible to authenticate the user, and security of the SNC would not be breached. The identification token might fail in two ways:

- The token might incorrectly decide that the user's secret information was correct, and erroneously sign a document. If this were a single-point failure, it would be due to a shortcoming of the token's design or architecture, and is beyond the scope of this document. (One workaround might to require two separate tokens.)

- The token might issue an incorrect signature, or otherwise fail to operate incorrectly. Since this is by nature a cryptographic operation, any failure mode will probably result in a failure to authenticate the user, and can therefore be ignored

If the identification token is used for other purposes (such as to verify that the SNC itself is authentic), then this analysis is not complete. However, such problems involve the architecture of the entire system, and not only that of the SNC, and merit additional consideration beyond that presented here.

System-wide, protection against single-point failures in user authentication is a function of the external boundary controller, and not of the SNC itself. For redundancy, an enclave might elect to use two boundary controllers, serially connected, in the same way this SNC uses interior and exterior boundary controllers.

©2000 Bionic Buffalo Corporation; All Rights Reserved.

BOOTING THE HOST COMPUTER

Once the SNC connects to an information domain, the SNC enforces isolation from any other domain. The only source for the HC's programs (operating system and application) is from within the current domain. The next step after connection is to boot the HC by loading an operating system into it.

Based on configuration, and possibly on the user's identity and credentials, the user may be given a choice of operating systems, or may be restricted to a single possibility. If there is a choice, then the user may be prompted to select the operating system to boot. This may be done using a first-level boot program running in the HC, or may be done by the SM using the user i/o device.

In any case, some operating systems (such as Unix and its variants) are network-aware to the extent that they can be booted without a disk, while others (such as Windows) must have disk support. To accommodate both kinds of operating system, the HC's BIOS must offer two possibilities:

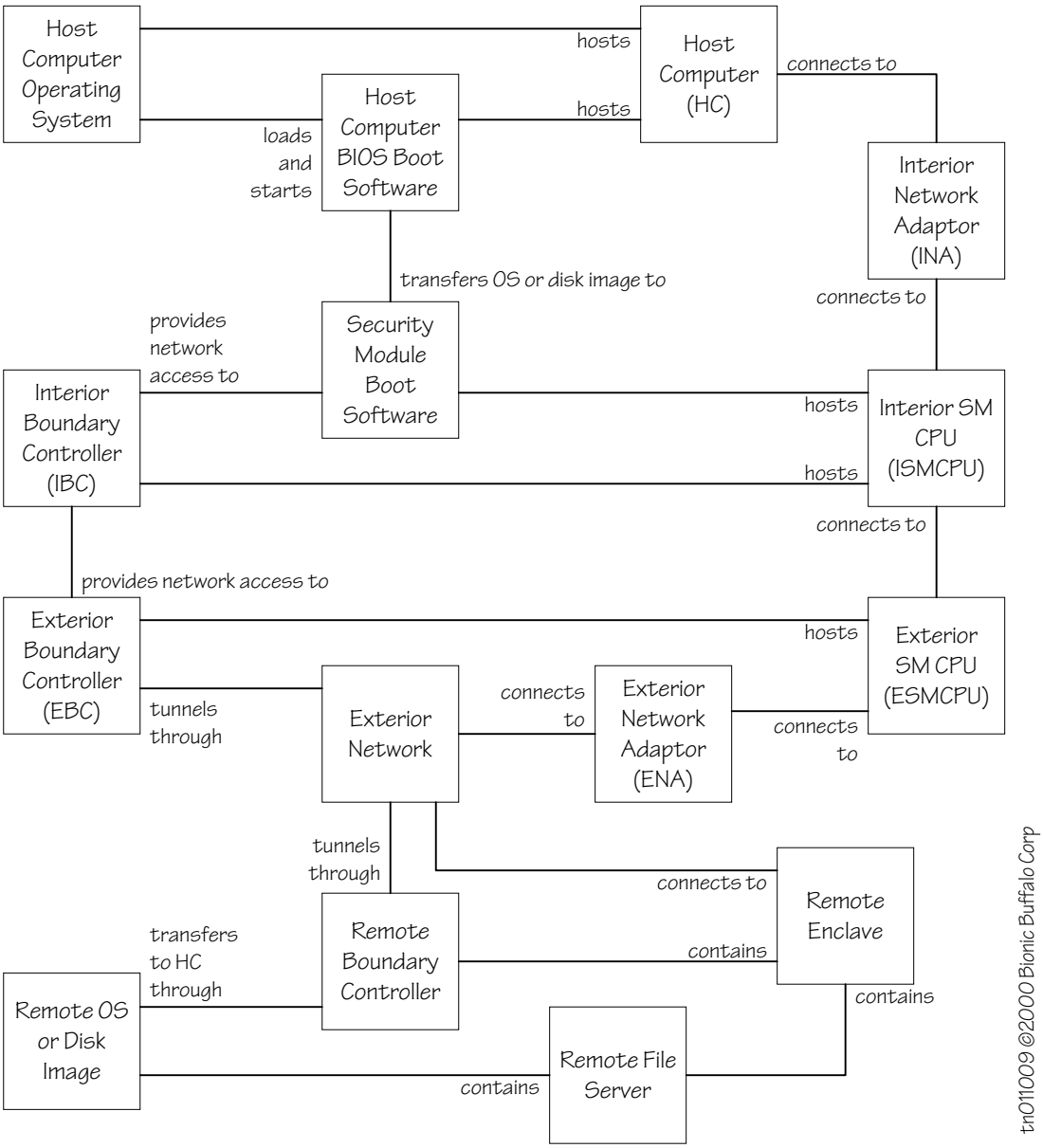
- Option 1: Using some combination of the HC BIOS and the SM software, emulate a disk drive. The virtual disk would be a file on a server in the remote enclave.

- Option 2: Provide a network boot facility, as is found on many workstations. (The usual protocols are those of tftp or nfs.)

Because the SM already contains a protocol stack, the ISMCPU hosts any network operations associated with the boot process.

Once to HC's operating system is booted, it will use its own network features, if any, to communicate with the other machines of the information domain. The HC's operating system will view the SM as a network card. The INA can be designed to emulate a standard network adapter, or it can have a novel interface and a custom driver can be written for the HC's operating system.

©2000 Bionic Buffalo Corporation; All Rights Reserved.



Thus, the INA does triple duty, and *may* have three distinct interfaces, for use in: (1) HC memory, cache, and register clearing, (2) the HC boot process, and (3) standard network adapter emulation.

PHYSICAL SECURITY

To achieve its purpose, the SNC must be physically secure. The level of physical security needed will vary depending on the application, but it is generally required that, except where intentionally allowed:

©2000 Bionic Buffalo Corporation; All Rights Reserved.

- the internal data and operation of the SNC cannot be observed from the outside
- the internal data and operation of the SNC cannot be modified
- communication between the user and the SNC cannot be intercepted
- attempts to compromise the security of the SNC will be detected

This is an architectural concept document, not a design description. Most of the problems of physical security are design problems, involving specific technologies, design parameters, and environmental and deployment considerations. Therefore, they are not addressed here. However, it is appropriate to note them briefly, since they will be crucial to effective implementation.

The concept of the SNC assumes that a legitimate user will not compromise the security of the information domain. In other words, he or she will not convey information from a domain to unauthorized parties, nor will he or she introduce inappropriate information into a domain.

Although it was not explicitly stated, the SNC is intended primarily for human interaction with other computers in an information domain. It is a secure version of a typical desktop, palmtop, or laptop computer. As such, it requires the usual peripherals: a keyboard, pointing device, and display. These peripherals constitute the first area of physical vulnerability.

The obvious problem with the peripherals is possible direct eavesdropping. For example, a camera might be placed so that the display and keyboard could be observed. Protecting against such threats involves design parameters and technologies beyond the scope of this concept document, but may involve devices such as alternative displays (narrow-focus displays) and VR equipment (goggles and gloves).

Another problem with peripherals is more indirect interception, and possible modification of, user-SNC interaction. For instance, the EMF generated by a display can be intercepted using an antenna placed remotely from the target, and the display reconstructed at the receiving site. Again, the defences are beyond the scope of this paper, but include devices such as shielding around the peripherals to prevent unwanted entry or exit of EMF.

The second area of physical vulnerability is the system itself, which includes the components described earlier in this Tech Note. Most computers, designed without consideration for physical security, are extremely vulnerable to a variety of physical attacks. Common attacks include:

- simple disassembly, with possible replacement of components and subsequent re-assembly
- interception of emitted EMF, to be analyzed to extract information about internal operation and data
- insertion of physical probes into the system, to observe or affect its behaviour (sometimes involving processes such as use of chemical solvents or abrasive techniques for entry)

©2000 Bionic Buffalo Corporation; All Rights Reserved.

- observation of power consumption patterns, or of signals leaking out through power supplies, which sometimes can provide insight into key values and algorithms

- observation using X-rays, ultrasonic devices, or similar techniques (it is possible to map the inside of a closed container the way seismography charts the inside of the earth, or as CAT scans and NMR map the inside of living organisms)

- modification of environmental factors (as temperature) or introduction of signals to alter the behaviour of the system

The most crucial danger is that the private keys of the SNC might be extracted, allowing some other device to impersonate the SNC. However, a great deal of other information from inside the SNC also has value, and must be protected.

Protection against all of these threats includes both design and manufacturing activities. A design must consider and counter them, but the manufacturing process itself is also subject to sabotage or simple mismanagement. For instance, if a component is substituted or a process altered during assembly, the security of the system can be compromised. This is especially significant, since complex devices such as the SNC and its subassemblies cannot be fully inspected and tested once they are assembled. It is imperative, for the highest level of security, to perform inspections and testing during the manufacturing operation, and not only at the end.

SOFTWARE SECURITY

Producing secure software is, in many ways, similar to the production of secure hardware. Security cannot be assured merely by inspection and testing at the end. Security comes from control of the entire programming operation, including design and coding reviews and inspection at all phases of development. It also requires the fundamental process of ascertaining that the assured software is, in fact, the software loaded into the SNC, and not a substitute program.

As with hardware security, the principles of software security are beyond the scope of this document. In general, they might be characterized by the following:

- the same good practices which characterize other good software development efforts (in the end, the software should do what is intended to do, and should not exhibit any form of undesirable, unintended behaviour or other errors)

- elimination of those unnecessary features which can be removed easily without increasing the likelihood of failure (every superfluous feature is an opportunity for failure or sabotage)

- a high level of paranoia, including a willingness to consider any individual component or team member as potentially hostile

©2000 Bionic Buffalo Corporation; All Rights Reserved.

- special attention to security engineering and management (which involves special knowledge, much of which must be studied and learned)

The architecture of the SNC allows relatively easy use of redundancy to improve the trustworthiness of the system. If the two SM CPUs, for example, are designed and programmed by separate teams, then it is difficult for either team, deliberately or accidentally, to compromise the security of the software.

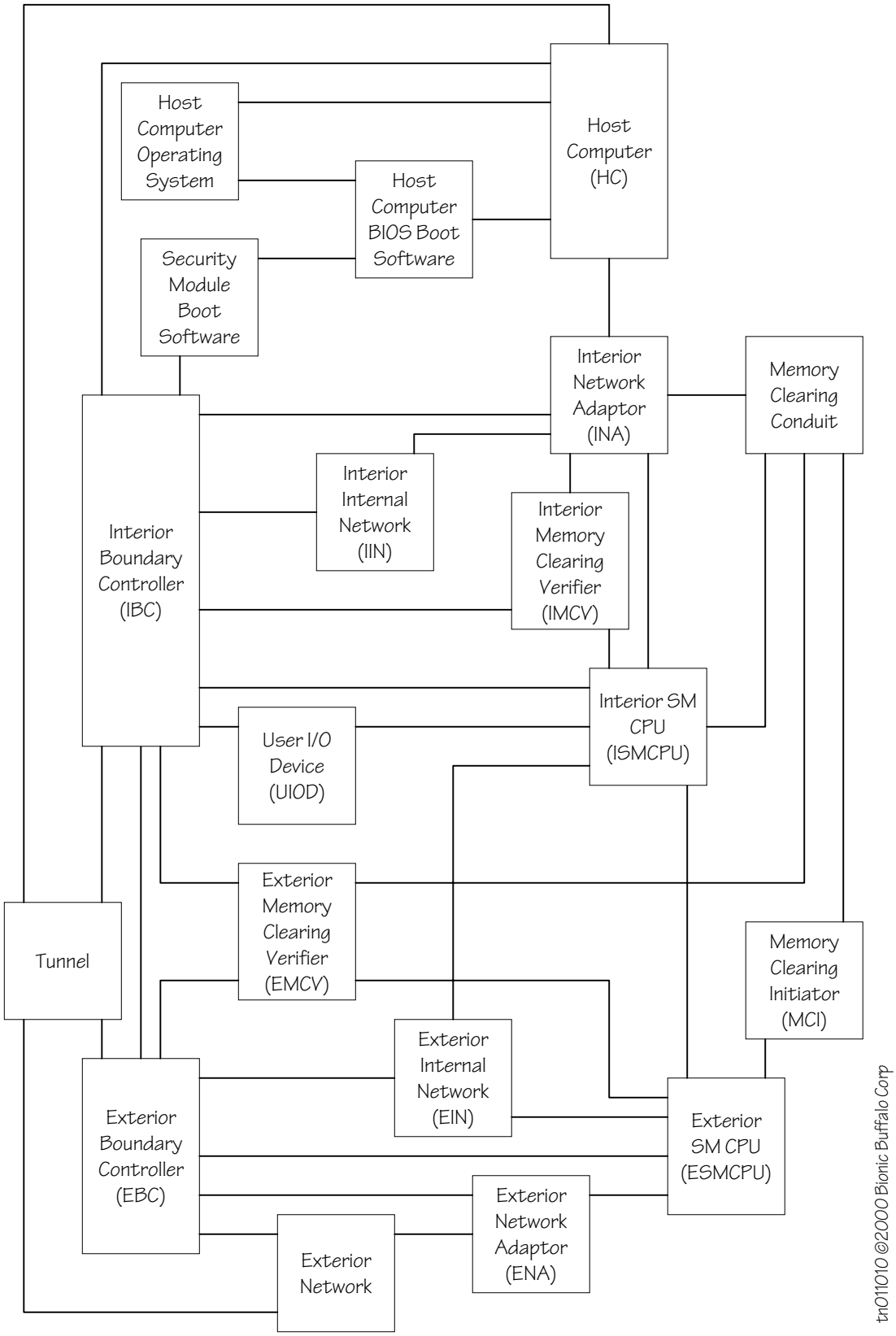
OVERALL ARCHITECTURE

This section presents a simplified entity-relationship diagram, which describes some of the components already discussed. It combines these components into a single diagram. Role annotation is omitted to make the drawing more compact.

Some major items have been omitted. For instance, the operating system for the two SM CPUs are not listed, and some relationships are also not included. However, the simplified drawing serves to illustrate an important reason to create such combined diagrams. With a diagram which gives an overall picture of the architecture, it is easier to do failure analysis on the design.

For instance, the combined diagram shows that any software path between the two SM CPUs must pass through either of two software components: the interior boundary controller or the memory clearing conduit. An analysis of the concept can focus on (a) the accuracy of the diagram, and (b) any possible security breaches in those two software components.

©2000 Bionic Buffalo Corporation; All Rights Reserved.



©2000 Bionic Buffalo Corporation; All Rights Reserved.

REDUCED-ASSURANCE VERSIONS

The concept presented here uses dual boundary controllers to assure very high probability that the security of an information domain will not be breached. In the overall lifecycle cost of the SNC, the extra cost of the redundancy will be relatively low.

However, there are two additional options to provide lower cost when extreme levels of security are not required.

The first such option eliminates the redundancy, while still providing the physical security. The main increase in risk comes from design and implementation failure, but there is also a possibility that single hardware failures might compromise security.

A second option can offer even lower cost. It restricts most of the physical security and tamper resistance to long-term key storage, and to session key generation. This option might be useful when the SNC is operated in a relatively safe physical environment, and the likelihood of physical attack is reduced. The design can employ the same technology used in identification tokens. Essentially, an embedded identity token carries the SNC's keys and certificates - its identity - and performs key generation and authentication for each session.

AUTONOMOUS REMOTE COMPUTING

In some applications, a device such as the SNC is useful without an operator or local human user. For example, a remote telemetry computer might process sensor data before forwarding it to a central location. It might be useful to run untrusted software on the remote computer, while still being assured that there would be no uncontrolled access to the information domain.

The same architecture as described above would apply, with some major simplifications. First, the tunnel to the remote enclave would require only a single authentication - for the computer itself, rather than for both computer and user. This eliminates the need for a user i/o device.

In addition, the HC's display, pointing device, and keyboard, which are significant potential vulnerabilities, are no longer required.

©2000 Bionic Buffalo Corporation; All Rights Reserved.

BIBLIOGRAPHY & CREDITS

This document presents ideas taken from many sources. It main contribution is to bring them together in a more complete form than the authors have heretofore seen.

The terminology and underlying architecture are based on those presented in *Security Architecture fore the AITS Reference Architecture*, Revision 1.0, 22 December 1997, by the DARPA Information Assurance Focus Group.

A good, general introduction to network security (including a discussion of IPsec and key exchange) is *Internet and Intranet Security*, by Rolf Oppliger, published by Artech House Inc in 1998. (ISBN 0-89006-829-1) The book *IPSec*, by Naganand Doraswamy and Dan Harkins, published by Prentice Hall PTR in 1999 (ISBN 0-13-011898-2) discusses IPsec and IKE specifically.

Physical security standards are outlined in *Security Requirements for Cryptographic Modules, FIPS 140*, published by the United States Department of Commerce. FIPS 140-1 was published in 1994, and its successor, FIPS 140-2 will probably be published sometime in 2000.

This Tech Note may be reproduced and distributed without payment of fees or without notification to Bionic Buffalo, as long as it is not changed, altered, or edited in any way. Any distribution or copy must include the entire Tech Note, with the original title, copyright notice, and this paragraph. For available Tech Notes, please see the Bionic Buffalo web site, at http://www.tatanka.com/doc/technote/index.htm, or email query@tatanka.com.