Wiretapping the Internet

C.J. Antonelli, P. Honeyman

Center for Information Technology Integration, University of Michigan, Ann Arbor†

ABSTRACT

With network security threats and vulnerabilities increasing, solutions based on online detection remain attractive. A complete, durable record of all activity on a network can be used to evaluate and train intrusion detection algorithms, assist in responding to an intrusion in progress, and, if properly constructed, serve as evidence in legal proceedings.

This paper describes the Advanced Packet Vault, a technology for creating such a record by collecting and securely storing all packets observed on a network, with a scalable architecture intended to support network speeds in excess of 100 Mbps. Encryption is used to preserve users' security and privacy, permitting selected traffic to be made available without revealing other traffic. The Vault implementation, based on Linux and OpenBSD, is open-source.

A Vault attached to a heavily loaded 100 Mbps network must capture, process, and store about a terabyte each day, so we have to be very sensitive to the recurring cost of operation and the reliability issues of 24x7 operation. We must also be sensitive to the admissibility of information collected by the Vault in support of legal proceedings; the legal ramifications of operating a vault, particularly at a public institution; and the public perception of its use.

Keywords: security, privacy, evidentiary record, intrusion detection, packet sniffing

1. INTRODUCTION

With network security threats and vulnerabilities increasing, solutions based on online detection remain attractive. A complete, durable record of all activity on a network can be used to evaluate and train intrusion detection algorithms, assist in responding to an intrusion in progress, and, if properly constructed, serve as evidence in legal proceedings.

This paper describes the Advanced Packet Vault (APV), a technology for creating such a record by collecting and securely storing all packets observed on a network, with a scalable architecture intended to support network speeds in excess of 100 Mbps. Encryption is used to preserve users' security and privacy. The cryptographic organization of the Vault permits selected traffic to be made available without revealing other traffic, by encrypting each packet with a key dependent on its source and destination IP addresses. The Vault implementation, based on Linux and OpenBSD, is open-source.

In the remainder of this paper, we define our goals and summarize potential solutions, and describe our previous work, a prototype implementation of the Vault, in some detail. We discuss the hardware, software, and cryptographic organization planned for the Advanced Packet Vault, and close with a discussion of legal, regulatory, and evidence-handling issues surrounding the APV.

2. GOALS

The architecture of the packet Vault reflects the following goals, which are mostly unchanged from those used for our prototype:

---

† Correspondence: `{cja,honey}@citi.umich.edu`, `http://www.citi.umich.edu/`

- **Commodity.** We want to build a packet Vault from commodity hardware and software, notwithstanding the attraction of expensive machines with fast buses and I/O devices. In the three years since our prototype was built, commodity processor speeds have risen from 133 to 750 Mhz, fairly inexpensive dual-PCI and 532 Mhz single-PCI motherboards are available, and striping IDE disk controllers provide high speed throughput to arrays of inexpensive disks[†]. With a Vault built from cheap parts in hand, we feel we can trade both money and time for speed by buying faster parts next year.
- **Completeness.** It remains vital to capture and store every packet in order to create a complete record. Because an adversary can exploit any form of packet triage, the only way to defend against all such attacks is to build a Vault that stores packets at the maximum rate the network delivers them.
- **Permanency.** While our prototype used recordable CD-ROM storage for creating the permanent record, that storage medium is surpassed by current magnetic tape's storage density and write speed. A single Mammoth2 tape drive delivers 12 MBps and stores 60 GB on an 8mm tape cartridge consuming 5.3 cubic inches. We therefore look to magnetic tape for long-term storage of the APV's permanent record.
- **Security.** We retain the requirement that the APV's permanent record be protected with strong cryptography, to guard against unsupervised inspection of Vault data. Accordingly, our design goals acknowledge the possibility of loss of physical control by assuming the worst, anticipating potential disclosure of the encrypted data. It is also vital that the data be organized in a way that allows restricted subsets of the traffic to be revealed. Simply put, we would like to publish keys that unlock certain data on a given tape, without the possession of those keys exposing other data on it.

3. POTENTIAL SOLUTIONS

There are many commercial packet sniffers on the market, in the form of intrusion detection products that search through the received packet stream for sequences of packets whose contents match predetermined "attack signatures." Suspicious packet traffic is remembered and reported to an administrator.

Such devices are good at performing triage on the packet stream, i.e. winnowing the wheat. They work best when they need only read a few bytes of each packet, and when the wheat to chaff ratio is fairly low, as the output medium is typically a magnetic disk. The APV, on the other hand, must read every byte of every packet and store all such bytes permanently. The former concern can be addressed only through careful systems integration; the latter requires mass storage. Moreover, the Vault must operate continuously.

Most importantly, no packet archiving system of which we are aware possesses a cryptographic organization that secures the data being archived against accidental or malicious disclosure, or permits selective disclosure of archived data. Considering the use to which the Vault will be put, such considerations are extremely important.

4. PROTOTYPE PACKET VAULT

We present a brief summary of our prototype Packet Vault below. The reader is referred to Antonelli et al[1] for a complete discussion.

4.1. Prototype Architecture

When we built the prototype we did not believe a single 133 MHz commodity machine could accept packets from the network, encrypt them, and write them to CD-ROM without becoming overloaded, so the Packet Vault prototype is composed of two 133 MHz PCI-bus Pentium machines connected via a private 100 Mbps Ethernet. One machine (the "listener") is connected to the network being monitored and is used to capture and encrypt packets, which are then sent over the private network. The listener never stores packets on permanent media, such as disk or tape.

The other machine (the "writer") receives encrypted packets and assembles them on magnetic disk for subsequent storage on CD-ROM. The two magnetic disks on the writer are attached to a common SCSI bus. A second SCSI bus dedicated to the CD-ROM recorder (CD-R) prevents bus contention; early recorders responded to data under-runs with recording failures. Figure 1 shows the hardware architecture of the Packet Vault prototype.

---

† A colleague reports 90 MBps sequential write performance to two sets of four 7200 RPM ATA/66 disks, each set with its own striping controller on its own PCI bus, and using Linux-based software striping across both sets.

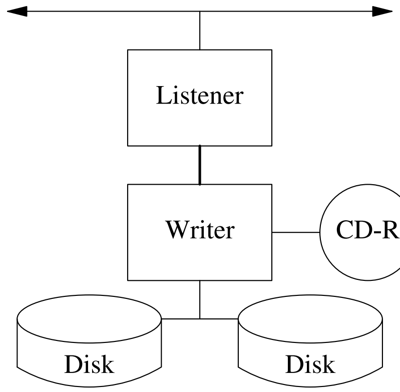

Figure 1 Architecture of the Packet Vault prototype.

We chose UNIX for both listener and writer for its familiarity and flexibility. OpenBSD was chosen for the listener for its kernel packet filtering support; early availability of CD-R drivers dictated the choice of Linux for the writer.

We use BPF[2] on the listener to capture all packets seen on the 10 Mbps network being monitored and write them to an *accumulator* file in an MFS[3] memory file system. We modified the BPF code to pass packets directly from the kernel network buffers to MFS, obviating two copies between user and kernel space. A listener process monitors the size of the accumulator file and renames it when it reaches 4 MB in size or after 1 minute has elapsed, which keeps the size of the MFS packet files manageable. The names of the packet files reflect the time of day they were created.

Another process on the listener polls the MFS for new packet files, encrypts their contents, and uses `rcp` to copy the files over the private 100 Mbps link to the writer. Unencrypted data are stored only in the MFS, so in the event of a system failure no unencrypted data remain.†

When enough packet files have accumulated on the writer to fill a CD-ROM, a background process is spawned on the writer. The writer process generates an ISO-9660-compliant image on magnetic disk containing the packet files and the cryptographic material necessary to permit later recovery of the packet data. The image is written, then purged from magnetic disk. A double-buffering scheme avoids disk contention between image generation writes and subsequent packet file writes on the same physical disk.

4.2. Prototype Cryptographic Organization

The cryptographic organization of the prototype Vault follows from our requirement that Vault media accommodate accidental disclosure. We thus anticipate the possibility of unrestricted access to a mass storage device filled with Vault data. We also endeavor to provide access to individual packet contents with fine granularity.

Our basic strategy is to encrypt all packet payloads; the challenge is to devise a means of associating different keys with different packets at some level of granularity. The ends of the spectrum are unattractive: one key per CD-ROM risks a serious breach if lost, while managing a different key for each packet clearly becomes unmanageable.

Our unit of granularity for associating packets with keys is the *conversation*, defined as a set of packets with the same pair of source and destination IP addresses. Including port numbers would offer finer control, but would also require special treatment for non-TCP streams and create problems with port-agile applications.

Each CD-ROM *volume* holds sufficient information to reconstruct the packet traffic it stores, thus no ancillary information need be managed. We use a multi-level encryption scheme. Symmetric key encryption is used to seal packet payloads and any additional information necessary to reconstruct the packets (explained below). Asymmetric key encryption is used to encrypt the symmetric keys. A trusted third party (such as the Regents of the University of Michigan) holds the private key. Figure 2 shows the cryptographic organization on CD-ROM.

---

† We run the listener with swapping disabled, but acknowledge potential attacks on RAM hardware.[4]

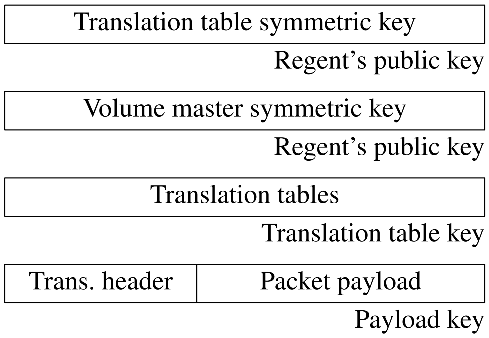

Figure 2 Cryptographic organization of the Packet Vault prototype.

Our implementation uses 1024 bit PGP[5] for asymmetric key and DESX[6] for symmetric key encryption. Starting with Karn's DES implementation[7] we added both pre- and post-whitening steps for each block:

$$DESX_{k.k_1.k_2}(x) = k_2 \oplus DES_k(k_1 \oplus x)$$

DES encrypts 64-bit blocks, so this requires 64 + 56 + 64 = 184 bits of key material, and conservatively extends the effective key length of DES in our environment to at least 95 bits with respect to key search (in the sense of Kilian and Rogaway[6]), while adding a trivial amount of computation to each block encryption.† To hinder traffic analysis, we obscure source and destination addresses by substitution. A translation table mapping real to substituted addresses is encrypted with DESX using a *translation table key* K_T unique to each volume. To speed up searches for specific conversations, a second table holds all pairs of translated addresses for which at least one conversation exists on the CD-ROM. The absence of a given pair of addresses in the second table means the CD-ROM contains no packets of that conversation, obviating an exhaustive search to establish this fact. Both translation tables are written to CD-ROM.

A key is constructed for a given conversation by combining the concatenated, untranslated source and destination IP addresses with a 192-bit *volume master key* K_V using exclusive-or, and then using DESX in CBC mode to encrypt a 192-bit constant with the combined value:

$$K_{C_i} = DESX_{K_V \oplus (SA_i \| DA_i)}(CONST)$$

The resulting 192-bit *conversation key* K_{c_i} is used to encrypt packet payloads of the conversation:

$$C_i = DESX_{K_{C_i}}(P_i)$$

A new volume master key and translation table key are generated for each volume. In the prototype, they are computed from previous keys:

$$K_{V_{i+1}} = DESX_{K_{V_i}}(K_{V_i})$$

$$K_{T_{i+1}} = DESX_{K_{T_i}}(K_{T_i})$$

where K_{V_0} and K_{T_0} were randomly generated. This scheme does not exhibit good long-term randomness; in the APV, we replace this with a practically strong random data generator.[10]

A new pair of PGP keys are generated per Vault instance. The public key is used to seal the volume master and translation table keys before they are written to CD-ROM.

Finally, we have built a rudimentary decryption engine that reconstructs all packet traffic stored on a CD-ROM given the private PGP key of the Vault that created it. We use the engine to verify the implementation of our cryptographic organization.

† If an attacker could obtain all the plaintexts for all encrypted packets on a volume, and if the average packet length is 100 bytes, this would yield 6 million plaintext/ciphertext pairs. Rogaway's effective key length expression becomes $55 + 64 - 1 - \log_2(6 \times 10^6) = 95$ bits[8]; the analysis of Blaze et al[9] makes the same recommendation via a different argument.

4.3. Prototype Experiences

The prototype Packet Vault was operational for several months in 1998, irregularly collecting packets from a 10 Mbps Ethernet that was usually lightly loaded but had periods when experimental video work caused traffic to exceed 70%. During the period 12-21 August 1998 we operated the Vault continuously, collecting about 7.7 GB on 15 CDs. Figure 3 shows a throughput trace; there were four interruptions of significant duration caused by Vault failures during this period.

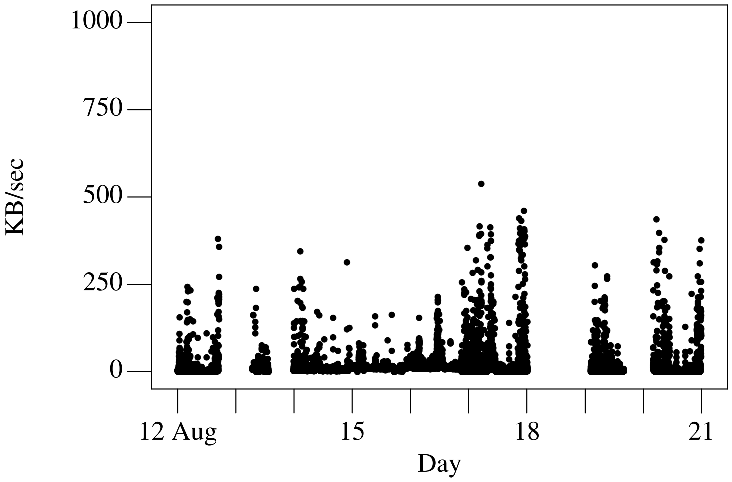

Figure 3 This graph shows Vault throughput measured in kilobytes per second during the period 12–21 August 1998. Because values are averaged in 30-second intervals there is some peak clipping; the maximum observed value is 1.2 MB/sec.

Systems engineering and integration were the major challenges in the construction and operation of the prototype. Bottlenecks discovered along the way were removed until the Vault could handle the incoming network traffic. For example, it was discovered that passing packets in and out of the kernel from BPF to MFS was too slow, so we modified the listener's kernel to skip the kernel/user space copies.

The other obvious target for performance optimization was the encryption code. We used a machine-specific implementation of the DES code compiled with full optimization and aggressively cached the DES key schedules. These changes sped up the encryption task by over 80%, but this also opens the door to a denial of service attack by an adversary who manufactures packets that defeat the caching.

5. ADVANCED PACKET VAULT

We divide our task of constructing an Advanced Packet Vault into two phases. In the first phase, we are developing a production 10-100 Mbps Vault, leveraging directly our experiences with the prototype. Improvements in performance and reliability will be achieved by porting the prototype to high-capacity hardware, including available hardware encryption devices and mass storage output devices, and by concentrating the Vault architecture onto a single host. The goal is to permit the Vault to archive all traffic found on a fully-loaded 10 Mbps network segment, and to permit the Vault to run for extended periods of time without supervision.

In the second phase of work, we will extend the design of the Vault to support operation in 100 Mbps environments and beyond. At such speeds, simply scaling up the current architecture is not feasible, requiring a parallel architecture instead. In this approach, groups of Vault engines cooperate to cover a high-speed network, possibly using either a round-robin technique or a frame check sequence, CRC, or message digest of a packet's contents to distribute packets among the available engines. In addition, the use of robust mass storage technologies are required to deal with data volumes well exceeding a terabyte per day.

In this section, drawing on our experiences with the Vault prototype, we discuss the issues in constructing the first phase of an Advanced Packet Vault, including cryptographic organization, systems engineering, terabyte storage technologies, and legal and regulatory constraints.

5.1. Cryptographic Organization

We continue to believe that our basic organization is sound, and will retain it for the APV. In the prototype, our unit of protected storage was a single CD-ROM, with unique, strong volume keys for each unit. We plan to keep roughly the same volume size in transitioning to magnetic tape, recording a number of volumes per tape, as we do not want to secure 50 or 100 GB of data with the same volume key.

Some time ago the Electronic Frontier Foundation's DES cracker and a worldwide network of personal computers jointly obtained the encryption key to a DES-encrypted message in 22 hours via a brute-force search of the key space.[11] Since the prototype Vault uses DES at the core of its encryption strategy, this calls the security of the data stored on the CD-ROMs into question. We believe our use of DESX inhibits the use of brute-force DES crackers, because it is difficult for an attacker to derive a plaintext/DES-ciphertext pair from a set of plaintext/DESX-ciphertext pairs obtained by a chosen-plaintext attack.[8] However, while we continue to believe in the strength of DESX, the effect of DES's rapidly declining strength on our cryptographic organization cannot be ignored.

For this reason, we will investigate the triple-DES (TDES) and Rijndael symmetric ciphers. TDES increases the key length of all symmetric keys as recommended by Blaze et al[9] and is roughly three times as slow as DES. Even though processors today are more than three times faster than they were when we started the prototype project, we will evaluate a PowerCrypt TDES crypto accelerator card[12], possibly against a dual-processor motherboard software implementation. Rijndael, recently selected for the AES, has the advantage that key schedule generation time is about a third of that required by DES. However, we know of no cheap Rijndael crypto card at present.

In any event, recent developments have shown that ciphers once considered secure are rapidly being broken as technologies and analysis techniques mature. It is not reasonable to assume that any cipher will be strong enough to withstand decades of determined attacks, which implies that loss of physical control of the Vault media will lead to exposure of the data they carry.

5.2. Systems Engineering

To scale the Advanced Packet Vault to higher network speeds and continuous operation requires careful systems analysis all along the processing pipeline, from the data bus to the media store.

Imagine a Vault capturing and storing the traffic on a heavily loaded 100 Mbps network; our challenge is to capture, process, and store about a terabyte each day, processing more than 12 MB every second.

Considering the throughput issue first, the system PCI bus will have to process 12 MBps inbound from the network card to the memory, 26 MBps from the memory to the magnetic disk, and 20 MBps outbound to the tape drives. The 58 MBps aggregate throughput is perilously close to the sustained throughput limits of a standard PCI bus (133 MBps maximum, typical throughput is about 1/3 to 2/3 of that). For this reason we are considering either a dual-PCI bus motherboard or a Revision 2 PCI motherboard rated at 532 MBps.

The PCI bus can make no real time guarantees as it admits asynchronous delays, so sufficient buffering must be designed into the data pipeline. This requires us to add magnetic disk; eliminating it is attractive in terms of overall throughput requirements, but 128 MB of memory buys only 10 s of buffering time. In addition, buffering to disk provides us the ability to rearrange the data — such as placing the translation tables at the beginning of the tape. In any event, migrating from CD-ROM to magnetic tape obviates the need for ISO-9660 image creation, which was a significant performance headache in the prototype.

We will replace the key generator we hastily constructed for the prototype with a practically strong random data accumulator and generator.[10]

5.3. Terabyte Storage Technologies

With respect to the storage issue, we have to be very sensitive to the recurring cost of operation, which includes personnel costs for system operation and maintenance, storage costs for media, and the cost of the media itself. Our target is the ability to store a year's worth of data in a cubic meter, at a cost of $50,000 for physical media. According to Gray and Shenoy[13] storage cost is improving by a factor of four every three years, so this target will soon be realized.

Terabyte Storage Technologies					
Type	MB/s	meas MB/s	GB/vol	$/GB	cc/GB
AIT2	6		50	1.80	1.9
DLT 8000	6	4.6	40	1.62	11
Mammoth2	12	9.6	60	1.50	1.5
AIT3	12		100		0.95
LTO	10		100		2.3
DVD-R	5/10		4.7	3.20-6.40	3.2

Table 4 This table compares speeds, capacities, cost, and storage volume for several magnetic tape storage devices. AIT2, DLT, and M2 are available today; AIT3 and LTO are promised for 1Q01. Writable DVD storage is shown for comparison.

We evaluate candidate mass storage technologies according to media write speed, storage density, cost per GB, and physical space consumed, measured in cubic centimeters per GB. Table 4 compares several popular commodity technologies, including AIT2 and AIT3 (Sony), LTO (Linear Tape Open), and Mammoth2 (Exabyte). Mammoth2 appears to have the best write speed, storage density, and space consumed, but is rumored to have reliability problems. AIT2 has good reliability but poor write speed; AIT3 promises to match Mammoth2 in speed, but is not available yet. LTO has the lowest risk in terms of architecture, but is not available yet either. AIT2, LTO and especially DLT are inferior to Mammoth2 in terms of physical space consumed per GB; AIT3 promises to be better. All technology lines have similar tape library offerings; we will select a small library with 7 to 20 media slots for our project.

Finally, there will be significant volume management and retrieval issues when dealing with petabytes of accumulated data. Storing tape metadata at the beginning of each tape will help, but the Vault must maintain a separate directory or tape catalog allowing retrievals specified for a given time interval to be mapped to a set of tapes. Other mappings may be useful as well. This will be even more important in the second phase of work, where data being recorded by multiple parallel Vaults must be located and accessed. We are investigating the use of a directory for this purpose, but have not yet fully investigated the security requirements here.

5.4. Legal and Regulatory issues

Technological issues aside, we must be sensitive to the admissibility of information collected by the Vault in support of legal proceedings; the legal ramifications of operating a Vault, particularly at a public university; and the public perception of its use. While the Vault is capable of casting a broad net, this practice is forbidden by established wiretap legislation. Public institutions are prevented by numerous federal- and institutional-level laws and policies from recording certain user data, e.g. student transcripts, without permission.

5.4.1. Legal Issues

Conducted as part of the prototype Vault work, a study by the Office of Policy Development and Education at the University of Michigan identifies a number of thorny legal issues connected with operation of the Vault.[14]

University of Michigan Policy forbids the interception of electronic mail without consent or a court order. Even in the absence of this policy, it is conceivable that a court would find that the Vault is intercepting electronic mail under Title I of the Federal Electronic Communications Privacy Act (ECPA). Interception for research purposes might not fall under the so-called "system administrator exemption," which permits interception in the normal course of business or as necessary to protect the rights or property of the service provider, and could therefore be unlawful. In a similar vein, the Family Educational Rights and Privacy Act (FERPA) prohibits disclosing student records to anyone who does not have a specific need to see them.

More generally, the courts have have read the First Amendment of the United States Constitution to prohibit government action that would tend to discourage citizens from speaking their minds. This "chilling effect" applies here, as awareness of the Vault's presence might tend to limit free speech by those whose subnets are being monitored.

The Vault is a research instrument, so its use is under the purview of the University's regulations on research involving human subjects. While most such research requires informed consent, and the signing of a consent form, it is possible to get an exemption from a University Institutional Review Board. Exemptions can be granted to projects that involve little risk to subjects or where informing the subjects of the nature of the experiment could bias results. As anyone who sends email to a user on a monitored subnet would arguably be a research subject, obtaining consent from all of them would be problematic.

Other issues potentially raised by use of the Vault include increasing the likelihood of copyright violations, bypassing an institution's policies with respect to creating permanent records subject to Freedom of Information Act (FOIA) requests, increasing the likelihood of civil discovery "fishing expeditions" against the material contained in the Vault, weakening users' Fourth Amendment protections against search and seizure, and increased exposure of the Vault's operators to civil liability.

The question of whether encrypted text is distinguishable from clear text in the eyes of the law has no definitive answer. In some cases, the purported protection offered by encryption is irrelevant; for example, it is not irrational to hold that copyright infringement takes place when a work is copied, not when it is read.

The study recommends that, at a minimum, all users be notified of the Vault's existence. This resolves some of the legal issues. However, notification does not cure First Amendment "chilling," nor does it address the FOIA or FERPA issues. In addition, it is not clear how to obtain consent from remote correspondents of local users. A recommended stronger form of consent would allow users to volunteer to be monitored, but requires us to separate on different sets of subnets those users who consent to monitoring and those who do not or to modify the Vault to discard certain packets. Both approaches are problematic.

Other recommendations include physically securing all archival materials, maintaining an access audit trail, capturing fewer types of packets, and using the Vault only for investigation of specific, ongoing security incidents.

For these reasons, we chose not to attach the prototype Vault to any production subnets at Michigan, nor to gather many packets. The data we do have were collected on a semi-private CITI subnet for a limited period after all users of the subnet were notified in advance.

In corporate environments, by contrast, the repeated refusal by the Congress to pass laws restricting workplace monitoring suggests that a business is free to monitor workers' communications on its computer systems without consent or knowledge. In fact, in securities trading environments, Wall Street regulations require such monitoring. Use of our Vault is less controversial in these environments (at least for now).

In private communications held since our study was concluded, we have been told that our analysis is overly conservative, and that operating a Vault should not be as problematic from a legal and privacy standpoint as we claim. The recent reaction to Carnivore[15] however, gives us no cause to be optimistic here.

5.4.2. Evidence handling

Sommer outlines general principles for the production of reliable, computer-derived evidence:[16]

- the scene of the crime must be "frozen"
- there must be continuity of evidence
- all procedures used in examination should be auditable

The prototype Packet Vault recorded onto CD-ROM — an immutable medium that effectively "freezes" the evidence. The APV records onto magnetic media which is not immutable. Including a digital signature with the tape contents will help prove the authenticity of any tape that purports to have been generated by a Vault. Periodically generating and publishing a digest of Vault output will help prove the point in time at which Vault data were actually recorded, as suggested by Haber and Stornetta.[17] Finally, operating a second Vault on the same subnet — or simply a machine that periodically creates a digest of the input traffic it sees, for comparison with a similar digest generated by a Vault — can serve to corroborate evidence generated by that Vault.

While it is possible that some packets traversing the network during periods of peak load are not seen by the Vault, its architecture precludes the generation of spurious packets, i.e. the Vault does not manufacture evidence. The Vault thus provides evidence that can be used to support other materials, such as audit logs. Continuity of evidence is indicated by the data handling architecture of the Vault, which is available for public inspection. The monotonically increasing time-stamped sequence of stored packets lends further support for continuity of evidence.

The Vault source code and, potentially, the contents of the Vault media are available for public inspection, which allows procedures to be audited.

6. CONCLUSION

We have discussed the Advanced Packet Vault, a technology for creating a permanent record of all traffic observed on a network. The cryptographic organization of the Vault preserves security and privacy by permitting selected traffic to be observed without revealing other traffic.

While operational and engineering issues abound, we observe that legal, regulatory, and evidence-handling issues surround the Vault. At the time of writing many of these issues have no definitive answers, auguring interesting times ahead.

7. ACKNOWLEDGEMENTS

We thank Mike Stolarchuk for his contributions to the architecture of the prototype Packet Vault. He also wrote the BPF layer modifications, and provided invaluable systems engineering assistance. Dan Boneh suggested the conversation key mechanism. The prototype Vault was partially supported by Bellcore (now Telcordia).

The work of the first author is supported in part by the U.S. Department of Justice under Award no. 2000-DT-CX-K001. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of the U.S. Department of Justice or the U.S. Government.

REFERENCES

1. C.J. Antonelli, M. Undy, and P. Honeyman, "The Packet Vault: Secure Storage of Network Data" in *Proc. USENIX Workshop on Intrusion Detection and Network Monitoring,* p. 103−109, Santa Clara (April, 1999).
2. Steven McCanne and Van Jacobson, "The BSD Packet Filter: A New Architecture for User-level Packet Capture" in *Proc. Winter USENIX Conf.,* p. 259−269, San Diego (January, 1993).
3. Marshall Kirk McKusick, Michael J. Karels, and Keith Bostic, "A Pageable Memory Based Filesystem" in *Proc. Summer USENIX Conf.,* p. 137−143, Anaheim (June, 1990).
4. Peter Gutmann, "Secure Deletion of Data from Magnetic and Solid-State Memory" in *Proc. Sixth USENIX Security Symp.,* p. 77−89, San Jose (July, 1996).
5. William Stallings, "Protect Your Privacy: The PGP User's Guide," *Prentice-Hall,* New Jersey (1995).
6. Joe Kilian and Phillip Rogaway, "How to Protect DES Against Exhaustive Key Search" in *Advances in Cryptology - Crypto '96, Lecture Notes in Computer Science,* ed. N. Koblitz, 1109, p. 252−267, Springer-Verlag (1996).
7. Phil Karn, `karn@unix.ka9q.ampr.org` (December, 1995).
8. Phillip Rogaway, *RSA Laboratories' CryptoBytes,* 2, 2 (Summer, 1996).
9. Matt Blaze, Whitfield Diffie, Ronald L. Rivest, Bruce Schneier, Tsutomu Shimomura, Eric Thompson, and Michael Wiener, "Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security" in `www.counterpane. com/keylength.html (January, 1996).`
10. Peter Gutmann, "Software Generation of Cryptographically Strong Random Numbers" in *Proc. Seventh USENIX Security Symp.,* p. 243−257, San Antonio (January, 1998).
11. Electronic Frontier Foundation in `www.eff.org/DesCracker/`.
12. Global Technologies Group, Inc. in `www.powercrypt.com/`.
13. Jim Gray and Prashant Shenoy, "Rules of Thumb in Data Engineering," Technical Report MS-TR-99-100, Microsoft Research, Redmond WA. (1999).
14. Joseph M. Saul, Peter Honeyman, and Virginia Rezmierski, "Policy Issues Related to Network Monitoring: The Secure Packet Vault," Unpublished, Ann Arbor (July, 1997).
15. United States Department of Justice in `www.usdoj.gov/jmd/pss/busopp.html`.

16. Peter Sommer, “Computer Forensics: an introduction” in `www.virtualcity.co.uk/vcaforens.htm (1997).`

17. Stuart Haber and W. Scott Stornetta, “How to Time-Stamp a Digital Document,” DIMACS Technical Report 90-80, Bellcore, Morristown (December, 1990).