Collaborative Research: NeTS—FIND: Privacy-Preserving Attribution and Provenence

Alex C. Snoeren, Tadayoshi Kohno[†]*, Stefan Savage, Amin Vahdat, and Geoffrey M. Voelker*

Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0404

[†]Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350

September 2007–August 2010

1 Project description

The Internet architecture was developed to support a number of key goals. Security was not among them. Indeed, in David Clark's classic paper, "The Design Philosophy of the DARPA Internet Protocols," the word *security* is not used once. By any accounting, security mechanisms have been added to the Internet in a fashion both post hoc and ad hoc, with minimal accommodations from the surrounding communications framework. Inevitably, these mechanisms have provided only an approximation to the security properties motivating their creation and have frequently conflicted with the existing network architecture in which they operate. The network firewall represents a classic example of this tension. A firewall is expected to help enforce an access control policy on traffic traversing its links and yet is unable to make any strong statements about the sender of a piece of traffic or the import of the content it contains. Moreover, in enforcing crude controls, firewalls routinely violate the end-to-end properties of protocols that traverse them.

We contend that many of these problems result from a mismatch between the level of abstraction provided by today's network architecture and the level necessary to describe real security properties. Real-world security policies are invariably about "who" and "what," while the Internet's architecture answers "where" and "how." For example, Internet addresses describe topological endpoints that are inherently virtual. Due to hot spots, spoofing, route hijacking, etc., an IP address in a packet may have only a transient relationship with the physical machine that sent it. Thus, an IP address has poor value for implementing either access control or accountability. Similarly, in the Internet architecture packet data is opaque and untyped by design. Thus, Internet datagrams carrying the proprietary source code of Cisco Systems are in no way distinguished from those carrying advertising for McDonalds. A security manager seeking to defend against data exfiltration must choose between extreme measures (e.g., an air gap) and half-measures (e.g., scanning for strings in network traffic).

We argue that a next-generation Internet architecture could bridge this gap by making the provenance of data a first-class design goal. However, it should be self-evident that the network alone cannot provide such a capability since data is created and manipulated on end-hosts. Hence, this proposal explores the architectural requirements and ramifications of both network and end-host support for data attribution and provenance. We are particularly interested in an architecture that addresses dual problems of *infiltration* and *exfiltration*: keeping unwanted traffic out and keeping sensitive data in. We propose mechanisms to both preserve data provenance as it traverses network routers and end hosts, and to make intelligent decisions within the network based upon the data's origin.

Our approach is based upon a mechanism for accurate, robust, and tamper-proof packet attribution that—critically—preserves appropriate levels of host and user privacy. Balancing privacy, identity and accountability requires a fundamental redesign of the Internet architecture: in particular, we propose that every packet be self-identifying. The contents of every packet, including this identifying information, is independently verifiable and non-repudiatable; every router or host is able to verify that any given packet is both attributable and unmodified. The confidentiality of possibly private information, however, including details concerning the packet's origin or route through the network, is preserved unless an appropriate third party is engaged (e.g., law enforcement agencies).

With privacy-preserving, per-packet attribution as a basic primitive, the next generation Internet would have the needed level of accountability to defend against denial-of-service attacks, botnets, and other forms of malware. The threat of accurate forensic analysis and legally enforceable liability provides a level of deterrence unimaginable in today's Internet. Furthermore, the inherent immutability of attributable packets significantly enhances the power and scope of the Internet defense arsenal: it becomes straightforward to effectively cordon off various networks and regions of the network, blocking unwelcome traffic from specified hosts and regions of the network.

Additionally, our attribution mechanism can be used to annotate network packets with information about the type of data contained within as a type of network capability, enabling the deployment of intelligent filters and reverse firewalls. State-of-the-art firewalls attempt to prevent attacks or data leakage through content inspection, but such techniques are easily thwarted by motivated adversaries or, frequently, innocent misconfigurations. Our group-based packet attribution mechanism elegantly addresses several shortcomings of previous capability-based approaches by preserving the privacy of the sender.

1.1 Objectives and significance

The goal of our work is to develop a new architecture for privacy-preserving attribution and provenance on the Internet. In particular, we propose to design and evaluate a key fundamental component of a secure next-generation Internet architecture: privacy-preserving, per-packet attribution. We propose a mechanism that not only enables non-repudiatable traceback and data exfiltration by empowering individual network elements to determine the provenance and authenticity of individual packets, but also preserves sender privacy through the use of shared-secret key escrow.

A key question we hope to address is whether our proposed attribution primitive is the appropriate architectural keystone for a new Internet security architecture. While we firmly believe it to be, we intend to put our hypothesis to the test by simultaneously developing an example application to both help define the architectural requirements and evaluate the effectiveness of our solutions. One of the chief benefits of this exercise will be to define the appropriate end-host interface to our new in-network functionality. Furthermore, we propose to make modest extensions to end-host functionality to track data provenance on the host itself. While we expect our work will innovate in this domain, the main objective is to prototype all three necessary components—network support, end-host support, and an example application—at a sufficient level of functionality to validate our proposed architecture.

Our work will combine new cryptographic developments with experience gained through the PIs' previous work on Internet traceback, secure routing, and attack mitigation technologies. We summarize the individual components of our proposed security architecture below:

- Privacy-preserving Per-packet Attribution: We argue that accountability should not be bolted-on from above, and instead requires first-class support from the network. In particular, we propose that every packet be self-identifying, but also disclosure-controlled. To demonstrate this approach, we will develop a packet attribution mechanism based on group signatures that allows any network element to verify post-hoc that a packet was sent by a member of a given group. Importantly, however, while any party can detect a forgery (a packet that cannot be properly attributed), actually attributing the packet to a particular member requires the participation of a set of trusted authorities from that group, thereby ensuring the privacy of individual senders.

- Data Exfiltration: We propose to explore the architectural ramifications of packet attribution on a system that limits network transmissions to authorized data. That is, only data that has been explicitly identified as safe to share (either manually or via an automated policy) is allowed to cross some network boundary; specialized firewalls drop and alert on any exceptions. We will adapt an existing content firewall prototype to use our attributions as a form of capability and evaluate the extent to which it simplifies the problem.

- Data Provenance: Finally, we plan to develop host-level support for data provenance. We will use a lightweight, VM-based approach to conservatively associate each outgoing data packet with any files or network input that it may depend upon. Specifically, when a network packet arrives, we will use its attribution information to taint all subsequent data derived from its contents, and transfer that attribution to any outgoing packets whose contents are dependent on the tainted data. Thus a record of each packet's provenance can be tracked transparently. We will further extend the content firewall described above to limit disclosure not only of explicitly protected data, but also of any data derived from protected data.

Taken together, we believe these components represent a substantial step towards a comprehensive security architecture for the next generation Internet. This collaborative proposal leverages the unique strengths of PIs at UC San Diego and the University of Washington: the requirements are informed by the UCSD PIs' deep knowledge of today's security threats through their involvement with the NSF-funded Collaborative Center for Internet Epidemiology and Defenses. PI Kohno recently received his Ph.D. from UCSD and is now on the faculty of the University of Washington. Kohno has a broad research record in practical cryptography and has also pioneered opportunistic mechanisms for identifying the physical origin of machines [61]. His ability to translate between the formal world of cryptographic primitives and the practical engineering of real-world protocols is instrumental to this project's success.

2 Results from prior NSF support

Our initial exploration of the proposed architecture was funded by the NSF through a one-year grant entitled "NeTS-FIND: Enabling Defense and Deterrence through Private Attribution" (September 2006–August 2007, $400,000).

Alex C. Snoeren. Snoeren's research has focused on support for mobile, secure [88, 100, 101], and flexible [4, 83, 84, 98, 99] wide-area routing. He is currently supported in part by a 2004 NSF CAREER award (CNS-0347949 March 2004–March 2009, $474,000) and serves as co-PI on two on-going NSF grants "Framework for Designing, Evaluating, and Deploying Global-scale Adaptive Networked Systems" (CNS-0411307 August 2004–July 2007, $345,636) and "NeTS-NBD: Algorithms and Infrastructure for Shared Mesh-based Broadcast" (October 2005–September 2009, $500,000). He was recently awarded two new NSF grants, "NeTS-NBD: Distributed Rate Limiting" (September 2006–August 2009, $360,000) and "CSR-PDOS: Harnessing Virtualized Resources in Cluster Computing," August 2006–July 2009, $450,000). In addition, Snoeren serves as senior staff in the NSF-funded CCIED center described below.

Snoeren's CAREER award, "Decoupling Policy from Mechanism in Internet Routing," supports work centering around the notion of a network capability [102] that empowers end hosts and ISPs alike to specify routes on a per-flow basis in a secure and accountable fashion. The project is initially exploring the use of network capabilities to implement an authenticated source-routing infrastructure called Platypus [87]. Platypus delivers the full power of AS-level source routing while addressing both the traffic engineering and accounting concerns of ISPs, reducing the barriers to deployment of a flexible, fine-grained wide-area routing system. This proposal leverages the insights gained from the design of network capabilities.

Snoeren currently advises six Ph.D. students and supervised three Masters theses. Through his students, Snoeren maintains ongoing collaborations with researchers at Google, HP Labs, and AT&T Labs–Research.

Tadayoshi Kohno. Kohno's research has focused on applied cryptography [1, 13, 14, 15, 16, 47, 49, 54, 57, 59, 60, 64, 66, 75], secure systems [65, 89, 111, 112], and information leakage [62, 63]. Within his applied cryptography research, Kohno's goal is to help further lift the reduction-based provable security approach of Goldwasser and Micali [51], and its practice-oriented variant [12], closer to the needs and constraints of real systems. For example, after discovering a security vulnerability in a portion of the Secure Shell (SSH) protocol, Kohno developed provably secure fixes that are not only compatible with existing artifacts of the SSH protocol, like the internal packet format, but that exploit the presence of these artifacts for security [15, 16]. Kohno has also developed methods for inferring forensics information about a TCP stream's physical device of origin [62, 63]. Following his original analysis of the Diebold AccuVote-TS electronic voting machines [65], Kohno has also developed new methods for improving the security of electronic voting [75, 89].

Kohno currently serves as a sub-contractor on the exploratory one-year FIND grant mentioned earlier. He is particularly focused on improving the cryptographic operations required for packet attribution.

Stefan Savage. Savage's research focuses on the security [72, 113, 74, 97, 11, 76, 79, 44, 80, 95, 94, 25, 2, 92], availability [22, 107, 56, 21], and measurement and manageability [36, 46, 10, 43, 90, 93, 91] of wide-area distributed systems. Savage's research is currently supported in part by an ongoing 2004 NSF CyberTrust grant, "Collaborative Center for Internet Epidemiology and Defenses" (CNS-0433668, October 2004 – September 2009, $3,100,000), and is co-PI on an NSF Infrastructure Grant, "FWGrid:A Research Infrastructure for Next Generation Systems and Applications" (EIA-0303622, September, 2003 – August, 2008, $1,800,000).

The goals of the first effort are to better understand the behavior and limitations of large-scale Internet epidemics [70, 72] using high-fidelity, active responders [113], and to develop systems that can automatically detect [71] and defend against Internet attacks in real-time [55, 56]. The critical distinction between this proposal and the CCIED mission is that CCIED focuses on today's Internet, while this proposal considers the potential to design the next-generation Internet. Moreover, the CCIED effort is focused primarily on defenses, while a significant portion of this proposal concerns the forensic support required for deterrence. The FWGrid infrastructure effort is focused on building and supporting a large computation and storage infrastructure, coupled with high-bandwidth wireless (>400Mbps) and wired (10Gbps) communications, and high performance video input and rendering output devices. It is focused on a number of key motivating applications including an omnipresent video-diary, a "day-in-the-life" of trace of enterprise-wide network and computation activity and passive video lab monitoring and analysis. Savage's involvement is particularly focused on the second of these activities.

Savage has supervised two Ph.D. students to completion: Ranjita Bhagwan, now at Microsoft Research and John Bellardo, now at Cal State SLO, and three M.S. students: Douglas Brown (currently completing the law program at NYU), Ishwar Ramani (now at Juniper Networks), Ryan Sit (founder of startup, Dropshots Inc. based on thesis work), and Christopher Tuttle (now at Google). Savage is currently advising eight Ph.D. students. The results of his previously funded efforts have led to collaborations with ICIR, AT&T Research, Google and Microsoft.

Amin Vahdat. Vahdat's research focuses on system support for scalable, high-performance network services, supported in part by a 2000 NSF CAREER award (CCR-9984328 June 2000-May 2004, $200,000), a 2000 NSF ITR award (ITR-0082912 September 2000-August 2003, $362,000), an ongoing NSF grant (CCR-0306490 September 2003-August 2005, $260,000) supporting the development of a large-scale emulation environment and a recently awarded NSF grant (CCR-0411307, $345,636) focusing on programming language and runtime support for large-scale distributed systems.

Vahdat's CAREER award on "Balancing Performance, Security, and Resource Utilization in Wide-Area Distributed Systems" supported a body of work on informed transcoding [28, 29, 30, 31, 32], overlay networks for federations of mutually distrustful autonomous systems [24, 68, 69, 109], and secure resource allocation [39, 50]. Vahdat's ITR award on "System Support for Automatic and Consistent Replication of Internet Services" focuses on consistency in replicated systems, and the inherent tradeoffs between various types of availability [119, 120, 121, 122, 123, 124]. Finally, an NSF CCR grant on "Evaluating Global-scale Distributed Systems using Scalable Network Emulation" supports the development and deployment of ModelNet [110, 118], ta scalable and accurate network emulation environment.

Through support from these grants, Vahdat has supervised four Ph.D. theses: Haifeng Yu, now at Intel Research Pittsburgh; Adolfo Rodriguez, now at IBM; Yun Fu, now at Yahoo! Corporation; and Dejan Kostic, now an Assistant Professor at EPFL; 12 M.S. theses; and 15 undergraduate research projects over the past five years. Vahdat is currently advising 9 Ph.D. students. The results of this work have led to continuing collaborations with Hewlett Packard, IBM, Intel, and Microsoft.

Geoffrey M. Voelker. Voelker's research has focused on wide-area distributed systems [17, 18, 19, 20, 21, 22, 37, 67, 86, 105, 106], computer networking [36, 52, 107, 108], and mobile and wireless computing [6, 7, 8, 9, 27, 35, 73]. This work has been supported in part by a 2003 NSF Trusted Computing grant (CCR-0311690, "Quantitative Network Security Analysis," August 2003–July 2005, $208,786), the a 2004 NSF CyberTrust Center grant (CNS-0433668, "Collaborative Center for Internet Epidemiology and Defenses," October 2004–September 2009, $3,100,000), and two recent grants through NSF NeTS-NBD (CCR-0411307, "Generating Realistic Network Traffic and Topologies," September 2006–August 2009, $345,636) and NSF CSR-PDOS (CNS-0615392, "Harnessing Virtualized Resources in Cluster Computing," August 2006–July 2009, $450,000).

Voelker's recent project "Quantitative Network Security Analysis" developed a combination of network analysis techniques and network measurement infrastructure to passively analyze large-scale Internet security threats such as denial-of-service attacks [77, 80], Internet worms [76, 78, 79], and port scans. Using a large "network telescope" we have developed at UCSD in combination with smaller monitoring platforms on other networks, we are measuring the vast majority of large-scale Internet attacks and capture global DoS, worm, and port scan activity on an ongoing basis.

Voelker has supervised seven Ph.D. theses (Song Cen, now at NextWave; Anand Balachandran, now at Microsoft; Ranjita Bhagwan, now at Microsoft Research India; Leeann Bent, now at Google; Renata Teixeira, now at Laboratoire d'informatique de Paris 6; Flavio Junqueira, now at Yahoo! Research, Barcelona; and Kiran Tati, now at VmWare) and six M.S. theses. He is currently advising 8 Ph.D. students and four M.S. students. The results of these efforts have led to close collaborations with AT&T Labs – Research, Microsoft Research, ICSI Center for Internet Research (ICIR), Google, and Intel.

3 General plan of research

Our proposed research rests on establishing two principal capabilities: a privacy-preserving mechanism that permits forensic attribution of individual network packets and a data provenance tracking mechanism that allows a data item's security-related attributes to be tracked across the hosts it traverses. We motivate each of these capabilities in turn and show how such an architecture can be used to help provide detterence and precise data confinement.

3.1 Attribution

Research in network security has traditionally focused on defenses—mechanisms that impede the activities of an adversary. However, paraphrasing Butler Lampson, practical security requires a balance between defenses and deterrence. While defenses may block an adversary's current attacks, only an effective deterrent can prevent the adversary from choosing to attack in the first place. However, creating such a deterrent is usually predicated on an effective means of attribution—tying an individual to an action. In the physical world this is achieved through physical forensic evidence—DNA, fingerprints, writing samples, etc.—but attribution can be uniquely challenging in the digital domain.

As Peter Steiner's famous New Yorker cartoon states, "On the Internet, nobody knows you're a dog." Indeed, a functional anonymity is *implicit* in the Internet's architecture since the lowest level identifiers—network addresses—are inherently virtual and insecure. An IP address only specifies a topological location—not a physical machine—and is easily forged on demand. Thus, it can be extremely challenging to attribute an on-line action to a particular physical origin (let alone to a particular individual).

The total absence of meaningful deterrence in today's Internet has profound implications on on-line criminality. First, it reduces the barrier to entry for new Internet crimes. To illustrate, consider Clark & Davis' cost-benefit model for criminal behavior [40]:

$$M_b + P_b > O_c p + O_c m P_a P_c$$

M_b and P_b are the monetary and psychological benefits of a crime, O_cp is the cost (overhead) of committing the crime, and O_cm is the monetary cost of a conviction, P_a the probability of getting caught and P_b the dependent probability of a conviction. With P_a approaching zero in the Internet domain, even low-margin crimes can offer significant value (hence SPAM). Second, without any meaningful risk of being caught, attackers are able to act repeatedly with impunity. It is this property that underlies the asymmetric nature of the modern computer security arms race. Attackers are free to improve their methods until they can break our defenses — leaving defenders forever in the role of catch-up.

It is our position that any future Internet architecture must move past this limitation and provide a post-hoc means to attribute the physical origin of any individual packet, message or flow. In particular, we argue that network traffic should be *self-identifying*: each packet is tagged with a unique non-forgeable signature identifying the physical machine that sent it. While such attribution may not definitively identify the *individual* originating a packet, it is the critical building block for subsequent forensic analysis, investigation and correlation; it provides a beachhead onto the physical scene of the crime.

3.1.1 Privacy preservation

However, there is a natural tension between this need for attribution and user desire (or legal rights) to privacy. Solutions that do not balance these interests have faced critical challenges to deployment. Thus, we believe that while the origin of every packet should be attributable, this origin must be opaque to all but properly authorized parties (e.g., under warrant). Specifically, we propose two key properties necessary for privacy-preserving attribution:

- *Post-hoc Authentication.* A properly empowered and authorized agency should be able examine a packet signature – even months after the packet was sent – and unambiguously determine the physical machine that generated it. This requirement also implies that signatures be non-forgeable (and hence non-replayable).

- *Privacy.* Packet signatures must be non-identifying to a normal observer. Thus, the encoded identifying information in these signatures must be opaque, in a strong sense, to an unprivileged observer. Moreover, the signatures must not serve as an identifier (even an opaque one) so different packets from the same source must carry distinct signatures. Overall, a user should have at least the same expectation of anonymity that they have in today's Internet excepting authorized investigations.

While these contrasting requirements appear quite challenging to satisfy, surprisingly there is a well-known cryptographic tool — the *group signature* — that neatly unties this particular Gordian knot. The group signature, first introduced by Chaum and van Heyst [33], is a public-key signature scheme in which a group manager creates per-sender signing keys. Only the group manager can reveal the signer from a signature, but anyone knowing the group manger's public key can verify that a signature was generated by a member of the group.

We apply group signatures to this problem as follows. Each machine is a member of some group and is provided with a secret signing key. Exactly how groups are constructed is very much a policy issue, but one pragmatic approach is that each computer manufacturer defines a group across the set of machines they sell. This is a particularly appealing approach because it side-steps the key distribution problem, as manufacturers are now commonly including Trusted Platform Modules (TPM) that encode unique cryptographic information in each of their machines (e.g., the IBM Thinkpad being used to type this proposal has such a capability).

Given a secret signing key, each machine uses it to sign the packets that they send. This signature covers all non-variant protocol fields and payload data as well as a random nonce generated on a per-packet basis. The nonce, the name of the group and the per-packet signature are all included in a special packet header field that is part of the network layer. Any recipient can then examine this header, and verify the signature to ensure that the packet was correctly signed by a member of the group (and hence *could* be authenticated post-hoc by the group manager).

We envision verification as not the sole province of the recipient, but as being a responsibility of the network as well. Thus, a network provider could block all un-attributable packets from being transited—providing a blanket "protection" for all of its clients. This service is similar in motivation to the source-address filtering that is used to mitigate spoofing in today's Internet [48], but provides a stronger guarantee. Moreover, this approach does not impose any restriction on the relationship between provider and customer: It allows considerable flexibility for innovation at the network layer (i.e., it is compatible with open "hotspot" access, mesh networks, overlays, etc.) and does not impose any bi-lateral administrative cost on providers.

We have built an initial prototype of this architecture, called Clue, that uses the short group signatures of Boneh, Boyen and Shacham as its central building block [23]. Written as an element in the Click router infrastructure, Clue is able to sign and verify each IP datagram, albeit with significant overhead. We propose to extend our system and continue our exploration of this architecture in two ways. First, we will develop and evaluate a number of performance optimizations that exploit synergies between the structure of group key cryptography, protocol dynamics and how networks are provisioned. Second, we will explore how more sophisticated packet signing schemas can provide additional capabilities (e.g., access control, support for tracking data provenance, physical location, etc) and try to understand the policy ramifications of these approaches. We outline these efforts in turn.

3.1.2 Performance optimizations

In our prototype software implementation, the overhead to sign (30 ms) and verify (50 ms) a packet are both significant. Since this is an architectural proposal it is fair to assume that technology will improve considerably in the time-frame over which adoption might take place. However, experience suggests that present-day plausibility helps overcome the natural psychological barriers that face all significant departures from existing practice. Thus, we propose to explore system-oriented optimizations to dramatically reduce or hide the overhead of the group signature operations in our baseline scheme.

Signature precomputation. Counterintuitively, much of the computation in signing a packet may have little dependence on the packet content at all. For example in the short group signature scheme of Boneh *et al.,* based on bilinear pairings, the most expensive operations are independent of the data being signed. Thus, for senders it is reasonable to compute signature precursors in advance during idle time or I/O. Our early experiments suggest that this could reduce the packet-serialized overhead by up to three orders of magnitude (rendering signing an inconsequential operation even in software). Thus, we believe that for many classes of use our architecture will be bottlenecked by the verification and not signing. Thus, most of our additional optimizations plans focus on this operation.

Windowed signatures. Again, counterintuitively, the size of the data being signed or verified is not a significant factor in the overhead of each operation. It is possible to create a *windowed* signature scheme in which the i-th signature is over the last k packets, $P_{i-k+1}, \ldots, P_i$. This allows the receiver to verify the last n packets in a single step, amortizing the cost of verification over a number of packets in a flow.

Unfortunately, a windowed signature optimization creates a number of conflicts with the existing Internet architecture: it cannot natively accommodate lost or reordered packets and it does not allow for verification of a single packet in isolation (e.g., such as necessary for an interactive protocol). However, we believe this full generality can be achieved by modifying the windowed signature algorithm slightly to encode both a hash over the packet and a hash over the window (an approach that, for technical reasons, we refer to as "flat hash trees"). A verifier may choose to verify any individual packet in isolation or, at their discretion, the previous k packets in the window. The verifier could switch modes dynamically to amortize costs when a window of packets is available, or a single packet when there is a desire to reduce latency.

Latency hiding. Many protocols decompose the request-response nature of the transport protocol from the application-level delivery semantics. This difference provides significant opportunities to hide overhead. For example, in the TCP protocol, there is no reason to serialize the generation of acknowledgements on the verification of signatures on incoming data packets. Instead, this verification can be overlapped in the round-trip time to receive the sender's next packet and amortized (as described above) across a full window of data before it is delivered to the user's socket buffer. For many client workloads we believe that the verification overhead may be largely if not completely hidden in this fashion.

Selective verification. A receiver may always shed load by randomly selecting which packets (or packet windows) to verify in exchange for the risk of unintentionally accepting an unattributable packet. Additionally, a receiver may choose to selectively verify packets if they are *bound* to an earlier verification event by some other protocol. For example, if the key-exchange portion of an SSL transaction is verified then a reciever might choose not to verify subsequent packets within the session (although this does require trusting the sender's ability to keep the session key secret).

Incremental verification. We believe it is possible, via a minor modification, to create an *incrementally verifiable* version of the signature scheme we have described. This construction allows unverifiable packets to be rejected very quickly on average with negligible impact on the cost of verifying well-formed packets (although at the cost of a larger per-packet signature). This tradeoff, perhaps combined with selective validation, is potentially valuable for tolerating denial-of-service attacks against the verification mechanism.

Parallel signing and verification. Our current software prototype uses a serial implementation of group signatures. However, many of the computations on both the signing and verification step are independent and could easily be exploited by either thread-level parallelism in a software implementation, or the natural data parallelism available in an FPGA or native hardware implementation. As well, both tasks are amenable to pipelining when the processing of multiple packets is overlapped.

Network load balancing. Some of the optimizations we have proposed above exploit the knowledge or buffered state available to a receiver. Fewer of these opportunities are available to intermediate network routers that have far less knowledge or state about a particular traffic flow. However, the network infrastructure provides its own opportunity for optimization resulting from the serial nature of its store and forward architecture. In particular, along a particular path a set of routers will each forward a packet in turn. Thus, if all unverified packets are dropped, any verification performed after the first router is effectively redundant. Even assuming that ISPs wish to verify their traffic independently, there are frequently three or more hops over which verification work could be load balanced. We plan to explore the ways in which this load balancing could take place, including explicit schedules and probabilistic schemes.

Overall, while we do not believe we can make privacy-preserving attribution "free," we believe its costs can be made acceptable on today's client systems. We believe that parallel hardware implementation will be appropriate for router-level implementation and for heavily loaded servers.

3.1.3 Design explorations

Our baseline architecture provides a single packet signature whose semantics are undefined beyond group membership. However, the general approach offers considerably richer design opportunities. For example, our architecture allows the possibility of a *set* of labels for a given packet. For example, a packet might be signed by several groups — any or all of which might be necessary for admission into a given network. Effectively, each signature is not only a capability to attribute a packet but also a declaration of jurisdiction. This allows a number of problems to be addressed directly. For example, while our existing scheme provides a mechanism to identify the sending machine, this value may be limited for public access hosts (although it may still offer some forensic value if there is additional evidence or surveillance material at the site). However, such a system could be designed to require a personally-held smartcard that contributed its own signature. Thus a packet would reflect *both* the machine that sent it and the smartcard that was used to enable the transmission. Alternatively, ISPs might tag first-hop packets with the country of origin. Thus a host could express reception policies such as "Accept packets whose source signatures have acceptable legal standards for opening AND are attached to networks within a compatible system of jurisprudence." More simply, a company might simply use a local group manager to provide access control for its employees (independent of their IP address or location). Finally, a packet might be annotated with a signature indicating the value of the data being carried. For example, confidential information might belong to one group – allowing network firewalls to prevent it from leaving the organization (unless perhaps the sender belonged to a particular authorized group as indicated by yet another signature).

We have developed our approach to be a network-layer capability strictly compatible with the Internet datagram model. Hence, we have not assumed any kind of negotiation or connection setup. However, if we relax this restriction it allows a number of other opportunities. For example, a recipient makes it clear what "capabilities" are required for access and thereby directs the sender to choose among signing keys available to it. Moreover, it might request permission to have the first-hop router add its own signature to identify the sender's physical location.

Finally, ours is a fundamental architectural exploration that opens up significant degrees of freedom in how it is applied. To wit, the policy implications of a robust packet attribution mechanism are many and varied. For example, it is not at all clear who should be able to authorize the "opening" of a packet. The Internet is an international entity and one without any overarching controlling legal authority. What then should IBM do when served with a warrant to open a packet? Whose laws should apply? Those in the country the requester resides, those where IBM resides, where the packet was found, or those of the host country (or nationality) of the owner of the machine that sent the packet? There are compelling cases for each. Will U.S. government networks be willing to receive packets signed by groups in Iran? Will they be willing to buy Thinkpad laptops manufactured by the Lenovo group in China? At this time the PIs do not have a position on the social value of these policies, but we believe that understanding them is critical to ultimately designing a mechanism that is both flexible and broadly acceptable.

3.2 Provenance

Attribution addresses the problem of associating traffic with a particular machine, and perhaps ultimately user, thereby providing the foundation for deterrence. It is also potentially usful for stating some policies about attributes of the sender, perhaps their group affiliation or authorization. However, attributing the origin of a packet helps little in enforcing policies about what *kinds* of data may traverse a network.

It has become increasingly desirable, for reasons of both competition and compliance, to limit which information may cross aministrative network boundaries, but this requirement is a poor match to today's network architecture. For example, software companies do not want unauthorized source code to be sent out to the Internet either from accidental exposure (e.g., attachment on email sent to the wrong address, or incorrect permissions on files accessible via a Web server) or malicious intent (e.g., a disgruntled employee, or hosts compromised by malware or hackers). Thus, while developers should be able to share source code unhindered within the internal company network, unauthorized transmission to the outside wolrd should be under explicit policy control. Of course, the network boundary may be internal as well. A company may need to sandbox different projects from each other because of contractual arrangements with customers, and therefore prevent leakage of intellectual property between internal groups. Still further complicating this problem, an organization's interest in its data is not restricted to a single format or expression, but to *derived* data as well (e.g., object files, libraries, and executables). Simply operations such as extracting the contents of a file and placing it into an e-mail message or placing the file in a compressed or encrypted attachment, should not be sufficient to defeat any exfiltration controls.

The key challenge facing any such approach is determining whether a particular packet is safe to distribute across a network boundary. The answer, of course, depends on the origin(s) of the contents of the packet. Ideally, if the system knew the source of the data, and the sources of any data upon which that data depends, it could make an informed decision. Here, origin does not necessarily refer a particular physical machine or location, but rather an object's provenance and the attributes associated with its ancestors (thus, a packet whose contents are ultimately derived from confidential material should itself be confidential).

To meet this challenge, we propose an exfiltration approach that uses our attribution architecture to combine data provenance tracking on the host with filtering in the network. We will use group signatures to label data objects; objects in the same group are equivalent with respect to policy. However, simply enshrining the capability to associate group attributes with packet data is not sufficient to meaningfully enforce data flow policies. The end hosts must also maintain these attributes as data is manipulated to prevent the "laundering" of a data's provenance. In a sense, this issue reflects a mismatch between end-to-end focus of Internet protocols and the data-oriented requirements of confinment. While the network is ideally situated for observing and controlling the flow of traffic between two points, only the end-host can can interpret the effect of its own processing once a packet is delivered (at best, the network may be able to correlate—as per Paxson's "stepping-stone" analysis [125]). Thus, it is our contention that the behavior of end-hosts must be included as part and parcel of any network architecture redesign.

To this end, we plan to provide mechanisms to label base objects, such as source code files, on each host directly. We will then use "tainting" techniques to automatically label new data objects, such as object files, that are derived from the base objects. We envision performing taint at the lowest level of the system (virtual machine monitor), and at the smallest data granularity (memory words), to transitively track data as it flows through the system. Our goal is to encompass any form of data derivation, whether it is via file input and output of a process (a compiler creating object files from source) or via cut and paste in the windowing system (copying code from source in an editor and pasting into an email message).

When transmitting packets, the host will use the attribution framework to sign those packets using the appropriate signature. If a Web server is sending a source code file tracked and labeled using our architecture, for example, the networking stack will sign the outgoing HTTP packets with the group signature associated with the data in the labeled and tracked file. In effect, the packets have been signed with a network capability that determines whether the outgoing packet can cross a network boundary.

Logically, the network component enforces policy by verifying the signatures on outgoing packets. Depending upon the policy desired, the network will drop unauthorized packets (either without signatures, or signatures that fail to verify). In implementation, the network component can be distributed across all machines (e.g., as a network filter at the virtual machine monitor layer) or centralized into a "reverse firewall" situated on the network boundary.

At a high-level, host attribution enables networks to express and implement policies about source and location, and treats packet contents as opaque data when applying policies. It is not what is in the packet that matters, but where in the network and from which machine a packet originated. Data provenance, in contrast, enables organizations to express and implement policies about packet contents: it is precisely what is in the packet payload that determines what policies apply. Combining host attribution and data provenance unifies the attribution architecture and enables organizations to express and implement a wide range of flexible exfiltration policies, including "Drop all outgoing unauthorized data" as well as "Drop all unauthorized data unless from these authorized hosts."

We recongize that ours is a significant departure from traditional network architectures – which traditionally abdicate responsibility for data upon end-point delivery. However, we do not believe that this traditional model offers the power necessary to implement real-world security policies. It is exactly this mismatch between meaning and function that results in inperfect and incomplete solutions such as today's network firewalls.

We develop our approach to exfiltration in two stages. In the next section, we first present a design of a high-performance centralized network verifier called Glavlit. Glavlit can be deployed independently of the attribution architecture, requiring no changes to or support on the host, and can mitigate covert channels. It is, however, limited to verifying explicitly identified data objects, not derived data, and requires knowledge of the transport protocol. Then, in Section 3.2.2, we discuss the challenges of evolving exfiltration to track data provenance and take advantage of our attribution architecture. With this integration, we extend exfiltration to encompass derived data, and simultaneously generalize and simplify the network verifier by pushing verification down into the IP layer.

3.2.1 Glavlit design

As part of our initial NSF FIND seed funding, we have designed an initial exfiltration system called Glavlit. Glavlit targets preventing information leaks through HTTP flows using a combination of content control and protocol channel mitigation. Glavlit does not require any changes on the host, but does require protocol parsers to pull out the protocol chatter from actual object transfer (for example, HTTP request and response data).

The Glavlit system consists of three components: Warden, Client, and Guard. Users use the Client to import objects into the system for Glavlit vetting. It provides an interface to users of the protected network to manage content control. A system user authenticates to the Client software so that the Warden can enforce additional vetting policy. For example, the Warden can enforce that only project leaders may vet files of a certain type.

The Warden is a central server that vets objects, determining whether the object has the ability to leave the network. The Warden can implement any type of digital and/or human reviews to determine if the object is fit for release. This process can be as simple as keyword search, or as rigorous as requiring approval from a committee of human analysts. Once approved, the object is partitioned into 1024-byte chunks that are then hashed. The resulting collection of hashes determines what content is allowed to leave the network, and enables high-speed verification at the Guard. We assume access to the Warden is controlled through appropriate authentication.

The Guard verifies objects as they traverse a network boundary. It operates as a transparent network gateway at the perimeter of the protected network. The Warden and Guard share hashes and meta-data of vetted objects. As packets pass over the outgoing network boundary, the Guard detects HTTP protocol covert channels and verifies that all file content has been previously vetted. It is a high-speed bidirectional network bridge that can actively stop data exfiltration as it occurs.

Verification consists of ensuring that data has been previously vetted. As the Warden receives packets, it locates data within the network stream and compares the hash of individual chunks to its pre-existing collection of hashes. To identify an object, we perform a lookup hash on the first 256 bytes of the file content. Once identified, the Guard hashes each chunk of object data and compares the result with the known hash. If any chunk does not match, we close the connection by injecting a TCP RESET.

To perform verification transparently in the network, the Guard must be able to reconstruct the network communications on the fly. Since Glavlit performs hashes on a chunk granularity, the packets associated with a chunk can be transmitted as soon as all chunks within a packet are fully verified. Because chunks may cross packet boundaries and packets may arrive out of order at the gateway, Glavlit may have to perform some packet buffering for further analysis before forwarding the packets. If the amount of required state becomes too large (e.g., as a result of some internally-mounted denial-of-service attack), it is always safe for Glavlit to drop buffered packets or to reset connections.

With this design, Glavlit separates vetting from verification. With this separation, our goal here is to enable generic and powerful vetting techniques to be employed. Since these techniques perform deep object analysis, they can be very time-intensive. Once vetting is complete, verification is done on all data leaving the network at the boundary to ensure that it was previously vetted.

Of course, there are a number of potential attacks against such a system. Sensitive data may be encoded and transmitted through simple dictionary exchanges, timing attacks in the protocol exchange, etc. Our ongoing work on Glavlit is to identify these attacks and prevent them to the extent possible. For instance, we are identifying known good communication patterns for particular protocols to prevent encoding attacks from taking place. Similarly, we are collecting recommendations for small modifications to protocols such as HTTP that might render them more robust against certain types of such attacks. While Glavlit takes steps to eliminate unauthorized channels, this problem is generally intractable. However, Glavlit is able to impose limits on the capacity of such channels and essentially "raise the bar" for attackers.

3.2.2 Tracking derivations

Glavlit is an exfiltration system for verifying outgoing traffic using contemporary capabilities. As we discussed above, we can greatly enhance network-based exfiltration by combining it with host-based data provenance tracking and our proposed packet attribution architecture. In this more powerful model, the host associates group signatures with data objects and any objects derived from that data. When the host transmits packet data taken from tagged objects, it signs the packets using the group signature. As packets traverse a network boundary, the network-based gateway verifies that the packets have a valid signature and have authorization to cross that boundary. Furthermore, an organization can combine host attribution with attribution based on data provenance to express policies that are a function of both the machine sending the data and the data being sent. In this section we describe our approach to tracking data provenance on the host and the challenges that we must address in more detail.

While it would be possible to make data provenance a first-class feature of an operating system [42], this approach would invariable impact a range of operating systems' APIs and require rewriting existing applications. Instead, we plan to explore a virtual machine monitor (VMM) approach, which has the advantage of being able to accommodate legacy operating systems and applications. However, because a VM abstracts a computer system at the level of individual memory accesses and instructions it is forced to infer higher-level abstractions such as sockets, transport-level connections, and so forth. Thus, the dual challenge and opportunity of this approach is to exploit the benefits of precise monitoring of data accesses while managing the added complexity of inferring and understanding higher-level state.

We intend to base our effort on a technique known as *tainting*. Thus, in principle, the execution state of a VM carries with it a set of implicit attribute values (e.g., current user, executing application) that apply to every write (either to memory or I/O device). In turn, a read from memory also carries with it the set of attributes that were used to write the location previously. Thus, a memory location is said to be *tainted* by the attributes that its last write was causally dependent on. By tracking the set of these attributes through the dynamic data flow of a program, one can maintain the complete set of causally dependent attributes for each memory object. Since this information is implicit in the data flow of the operating system and those applications using it, this causality can be inferred dynamically rather than requiring changes to existing legacy environments.

For our purposes, we do not have to solve the general problem of tracking the complete set of execution attributes. Instead, we associate data objects with signatures and track this mapping as the taint attribute as applications and the operating system use the data. The foundation for tracking begins when users or automated tools associate a policy with data objects, such as a file, thereby associating with that object a signature that corresponds to a group representing the policy. Any subsequent execution using that data as input will propagate the mapping to any outputs, such as network packets, based on the data.

Building on our previous experience with the Xen VMM [52, 113], we plan to produce a modified VMM that provides and tracks low-level data tainting services. Our initial prototype will be based on an approach from Cambridge University that dynamically switches between direct virtualization and machine emulation to track derived memory accesses [53]. In this approach, buffers holding tainted data (e.g., input) are unmapped in the TLB and subsequent faults switch the VMM into a full machine emulator (e.g., such as Qemu). The emulator carefully tracks instructions and register contents and marks any writes directly or indirectly dependent on input data as tainted by the associated attributes. This approach has been used extensively to find control flow violations, such as buffer overflow attacks, but we propose to use VMM taint to track data provenance via signature association.

There are several clear challenges with this approach. First, an obvious concern is efficiency. Since tainting requires maintaining state for each object being monitored, this can both create significant storage overhead as well as execution overhead to track the dataflow of tainted objects. Minimizing these overheads will require several optimizations. Execution overhead can be minimized by using a conservative method for tainting attributes that are in scope for large periods of time. For example, it is not necessary to perform dataflow analysis to determine that the current userid should taint all writes in the address spaces of a user's programs. Less obvious, if a Word document causes the Normal.doc template to change, it will affect all subsequent documents created by the same user. Some of these optimizations can be special-cased, but in general we will explore performing them automatically by monitoring how long a particular value causes dynamic data flow to be activated and adjusting our policy accordingly. Note that such optimizations may also cause objects to be unnecessarily tainted due to their conservativism.

A similar set of optimizations can be used to reduce the amount of state that must be maintained. Thus, rather than track data provenance for each byte, some attributes may implicitly apply to all memory allocated by an application and others may have variable granularity (e.g., a list of attributes mapping to memory ranges will be compact if the mapping is sparse as we would expect). However, we expect that initial implementations will require significant resource requirements and may cause perceptible degradation of some applications. We believe this is an acceptable tradeoff given the demands of this application and the relative change in the demands of normal data processing applications when compared to the capability of emerging multi-core processors.

The final pieces of this exfiltration architecture are at the network layer. On the host, the taint tracking system propagates signature associations on data from application memory down into the socket buffers of the operating system. When the networking stack on the host transmits a packet, the IP layer resolves the signature associated with the data in the packet payload. It then uses the per-packet attribution library to sign the packet. Packets that traverse an administrative network boundary, such as external links to the Internet, pass through a network reverse firewall evolved from the Glavlit Guard. For each packet, this firewall verifies the packet signature to vet whether the packet can leave the network. Depending upon the policy associated with the signature group, it can drop packets based upon a lack of signature, signatures that fail to verify, whether signed packets were sent from authorized hosts, etc.

As discussed earlier in this proposal, current implementations of the attribution operations have significant overhead. Fortunately, using attribution to track and verify data provenance uses the same operations as privacy-preserving per-packet attribution. As a result, they immediately share the benefit of optimizing the attribution methods. In particular, optimized hardware implementations of the verify operation for use in routers can be used by the exfiltration firewall as well. An alternative implementation of the network firewall is to distribute its functionality to every host, having the VMM perform the verification and filter duties. Although a distributed implementation scales as the number of hosts in the network grows, it also relies upon every host attached to the network being configured and operating properly.

4 Research plan

We now present a year-by-year account of research topics that we will pursue as part of this proposal.

Year 1

- Improve performance of prototype group-signature-based packet-attribution mechanism by implementing signature precomputation and windowed signatures.

- Complete implementation of Glavlit, considering implications for taint tracker and packet attribution.

Year 2

- Demonstrate the effectiveness of per-packet attribution technology by developing a traceback tool that uses key-escrow to reveal non-repudiatable identifying information about a packet's source.
- Develop VM-based tainting mechanism to track data provenance on standard operating systems, leveraging packet attribution to track network communications.
- Continue to improve performance of packet attribution by incorporating selective and incremental verification.

Year 3

- Deploy prototype attribution infrastructure and traceback mechanism on PlanetLab, DETER, or GENI facility if available.
- Integrate taint-based provenance tracking with Glavlit implementation.
- Refine network stack interface for packet attribution based on experience gained linking attribution with host-based provenance in Glavlit.

5 Related work

Early, high-profile distributed denial-of-service attacks, mounted largely with spoofed source addresses, spurred the development of a number of traceback techniques that could enable operators to determine the true source of attack traffic [115, 94, 101]. The availability of these tools, along with increased vigilance and deployment of reverse-path filtering, has dramatically decreased the power of such attacks. Unfortunately, spoofed flooding attacks were merely the first salvo in an arms race between motivated cybercriminals and network operators. In an effort to simultaneously obscure their identity and increase the number of potential attack sources available to them, attackers are now recruiting third-party end hosts to construct large-scale botnets, reported to number in the hundreds of thousands. Because these networks are so numerous, various researchers have proposed to fundamentally restrict the Internet's default best-effort any-to-any service model using a capability-based model [3, 5, 116, 87, 26]. The key differentiator between our proposal and the previous capability-based schemes is the inability of routers, firewalls, or even destinations to determine the packet's origin without the consent of either the sender or a (group of) trusted third party(s). We believe this distinction is critical to ensuring privacy in the next generation Internet.

There are commercial content-control solutions that perform vetting and verification simultaneously on the gateway [82, 85]. The primary drawback to this approach is that it is unable to perform whole-object analysis, critical for supporting modern transfer protocols. Our approach, on the other hand, performs only object vetting at the gateway, removing any time constraints on verification. Other approaches to preventing unauthorized data from leaving the network include scrubbing files offline [45, 104], explicit communication between clients and the network gateway [96], and redundant filter nodes in back-end fault-tolerant services [117].

A variety of taint-based data provenance mechanisms have been explored in the context of malware defenses. In particular, researchers have proposed a number of systems that taint memory regions to prevent control transfers to untrusted code [81, 58, 114]. More aggressive proposals have argued for first-class operating system support [42] and processor support for byte-level tainting [34, 41, 103]. Aside from execution prevention, there is a large body of work on dataflow analysis that attempts to follow the propagation of particular pieces of data through a program's execution. We do not propose to attempt anything as sophisticated; our approach is essentially a more efficient implementation of the TaintBochs project, which logs taint propagation in the Bochs x86 simulator [38].

6 Broader impacts

While this work is high risk, if this approach is adopted as a core component of the next-generation Internet, it has the potential to effect a qualitative change in the level of accountability, security, and privacy provided by the network. As our nation's and indeed the world's infrastructure comes to increasingly rely on an intertwined web of network services, it is critical that we advance the state-of-the-art for securing these distributed services. Moreover, many of the core tensions in this proposal—such as between attribution and privacy—are common across a range of distributed Internet services. We expect the insight we gather to be reflected in distributed applications beyond merely the network layer services we intend to study. Finally, we believe that the introduction of more sophisticated cryptographic concepts to the networking community has merit in itself. Networking researchers are driven by the tools they have available and creating a better appreciation between the networking and cryptography communities can only benefit both.

In addition, it is our intention that the resources from this grant and the context of this work will create educational opportunities for students at a variety of levels. Undergraduates working with the PIs have co-authored a number of conference publications and several have gone on to pursue graduate study at universities including Princeton, Cambridge University, University of Michigan, UC San Diego, CMU, and University of Washington. One such student, Ethan Eade, was awarded a Marshall fellowship and was runner up for the highly competitive CRA Outstanding Undergraduate Award in 2004. As well, PI Savage has been involved in teaching the concepts of network security to high school students as part of the University of California's summer COSMOS program. COSMOS brings teams of highly motivated students between the 9th and 12th grades to UCSD for a four-week intensive residential program. As part of this effort they select clusters in sub-disciplines of science and engineering and complete hands-on projects in partnership with faculty and graduate students.

All the PIs are committed to education and curricular development. Student evaluations routinely rank their courses among the top in the department. PI Vahdat, in particular, was awarded the 2003 Duke University David and Janet Vaughn Distinguished Teaching Award. PI Voelker received the 2001 School of Engineering Teaching Award and the 2006 Chancellor's Associates Award for Excellence in Undergraduate Teaching. The effectiveness of thier courses rests partly on incorporating research materials into the course work. All the graduate and some of the advanced undergraduate courses offered by the PIs are project-oriented, involving an original research effort culminating in a project report and presentation during an end of term "mini-conference." A number of these course efforts have led to subsequent conference publications.

References

[1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In V. Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 205–222. Springer-Verlag, Aug. 2005.

[2] A. Aggarwal, S. Savage, and T. Anderson. Understanding the performance of tcp pacing. In *Proceedings of IEEE Infocom Conference*, pages 1157–1165, Tel-Aviv, Israel, Mar. 2000.

[3] D. G. Andersen. Mayday: Distributed filtering for Internet services. In *Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems (USITS)*, pages 31–42, Seattle, WA, Mar. 2003.

[4] D. G. Andersen, A. C. Snoeren, and H. Balakrishnan. Best-path vs. multi-path overlay routing. In *Proceedings of the USENIX/ACM Internet Measurement Conference*, pages 91–100, Miami, FL, Oct. 2003.

[5] T. Anderson, T. Roscoe, and D. Wetherall. Preventing Internet denial-of-service with capabilities. In *Proceedings of the 2nd ACM Workshop on Hot Topics in Networks (HotNets-II)*, Cambridge, MA, Nov. 2003.

[6] P. Bahl, W. Russell, Y.-M. Wang, A. Balachandran, G. M. Voelker, and A. Miu. Pawns: Satisfying the need for ubiquitous secure connectivity and location services. *IEEE Wireless Communications*, 9(1):40–48, Feb. 2002.

[7] A. Balachandran, G. M. Voelker, and P. Bahl. Hot-spot congestion relief in public-area wireless networks. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, Callicoon, NY, June 2002.

[8] A. Balachandran, G. M. Voelker, and P. Bahl. Wireless hotspots: Current challenges and future directions. *Mobile Networks and Applications*, 10(3):265–274, Jan. 2005.

[9] A. Balachandran, G. M. Voelker, P. Bahl, and V. Rangan. Characterizing user behavior and network performance in a public wireless lan. In *Proceedings of the ACM SIGMETRICS Conference*, Marina Del Rey, CA, June 2002.

[10] J. Bellardo and S. Savage. Measuring packet reordering. In *Proceedings of the ACM/USENIX Internet Measurement Workshop (IMW)*, pages 97–105, Marseille, France, Nov. 2002.

[11] J. Bellardo and S. Savage. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *Proceedings of the USENIX Security Symposium*, pages 15–28, Washington, D.C., Aug. 2003.

[12] M. Bellare, J. Kilian, and P. Rogaway. The security of cipher block chaining. In Y. Desmedt, editor, *Advances in Cryptology – CRYPTO '94*, volume 839, pages 341–358. Springer-Verlag, Aug. 1994.

[13] M. Bellare and T. Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 491–506. Springer-Verlag, May 2003.

[14] M. Bellare and T. Kohno. Hash function balance and its impact on birthday attacks. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 401–418. Springer-Verlag, May 2004.

[15] M. Bellare, T. Kohno, and C. Namprempre. Authenticated encryption in SSH: Provably fixing the SSH binary packet protocol. In V. Atluri, editor, *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 1–11. ACM Press, Nov. 2002.

[16] M. Bellare, T. Kohno, and C. Namprempre. Breaking and provably repairing the SSH authenticated encryption scheme: A case study of the Encode-then-Encrypt-and-MAC paradigm. *ACM Transactions on Information and System Security*, 7(2):206–241, May 2004.

[17] L. Bent, M. Rabinovich, G. M. Voelker, and Z. Xiao. Characterization of a large web site population with implications for content delivery. In *Proceedings of the International World Wide Web Conference (WWW)*, New York, NY, May 2004.

[18] L. Bent, M. Rabinovich, G. M. Voelker, and Z. Xiao. Towards informed web content delivery. In *Proceedings of the 9th International Workshop on Web Content Caching and Distribution (WCW)*, Beijing, China, Oct. 2004. Best student paper.

[19] L. Bent and G. M. Voelker. Whole page performance. In *Proceedings of the 7th International Web Caching Workshop (WCW)*, Boulder, CO, Aug. 2002.

[20] R. Bhagwan, D. Moore, S. Savage, and G. M. Voelker. Replication strategies for highly available peer-to-peer storage. In *Proceedings of the International Workshop on the Future Directions in Distributed Computing (FuDiCo)*, Bertinoro, Italy, June 2002.

[21] R. Bhagwan, S. Savage, and G. M. Voelker. Understanding availability. In *Proceedings of the International Workshop on Peer To Peer Systems (IPTPS)*, pages 256–267, Berkeley, CA, Feb. 2003.

[22] R. Bhagwan, K. Tati, Y.-C. Cheng, S. Savage, and G. M. Voelker. Totalrecall: System support for automated availability management. In *Proceedings of the 1st ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 337–350, San Francisco, CA, Mar. 2004.

[23] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. pages 41–55, 2004.

[24] R. Braynard, D. Kostić, A. Rodriguez, J. Chase, and A. Vahdat. Opus: an Overlay Peer Utility Service. In *Proceedings of the 5th International Conference on Open Architectures and Network Programming (OPENARCH)*, June 2002.

[25] N. Cardwell, S. Savage, and T. Anderson. Modeling tcp latency. In *Proceedings of IEEE Infocom Conference*, pages 1742–1751, Tel-Aviv, Israel, Mar. 2000.

[26] M. Casado, T. Garfinkel, A. Akella, D. Boneh, N. McKeowon, and S. Shenker. Sane: A protection architecture for enterprise networks. In *Proceedings of the 3rd ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, San Jose, CA, May 2006.

[27] S. Cen, P. C. Cosman, and G. M. Voelker. End-to-end differentiation of congestion and wireless losses. *IEEE/ACM Transactions on Networking*, 11(5):703–717, Oct. 2003.

[28] S. Chandra, C. Ellis, and A. Vahdat. Multimedia Web Services for Mobile Clients Using Quality Aware Transcoding. In *Proceedings of the Second ACM/IEEE International Conference on Wireless and Mobile Multimedia*, August 1999.

[29] S. Chandra, C. Ellis, and A. Vahdat. Application-Level Differentiated Multimedia Web Services Using Quality Aware Transcoding. *IEEE Journal on Selected Areas of Communications (JSAC)*, 18(12), December 2000.

[30] S. Chandra, C. S. Ellis, and A. Vahdat. Differentiated Multimedia Web Services Using Quality Aware Transcoding. In *INFOCOM 2000 - Nineteenth Annual Joint Conference of the IEEE Computer And Communications Societies*, March 2000.

[31] S. Chandra, C. S. Ellis, and A. Vahdat. Managing the Storage and Battery Resources in an Image Capture Device (Digital Camera) using Dynamic Transcoding. In *Third ACM International Workshop on Wireless and Mobile Multimedia (WoWMoM'00)*, August 2000.

[32] S. Chandra, A. Gehani, C. S. Ellis, and A. Vahdat. Transcoding Characteristics of Web Images. In *Multimedia Computing and Networking 2001 (MMCN'01)*, January 2001.

[33] D. Chaum and E. van Heyst. Group signatures. pages 257–265, 1991.

[34] S. Chen, J. Xu, N. Nakka, A. Kalbarczyk, and R. Iyer. Defeating memory corruption attacks via pointer taintedness detection. In *Proceedings of IEEE International Conference on Dependable Systems and Networks*, 2005.

[35] Y.-C. Cheng, J. Bellardo, P. Benko, A. C. Snoeren, G. M. Voelker, and S. Savage. Jigsaw: Solving the puzzle of enterprise 802.11 analysis. In *Proceedings of the ACM SIGCOMM Conference*, Pisa, Italy, Sept. 2006.

[36] Y.-C. Cheng, U. Hoelzle, N. Cardwell, S. Savage, and G. M. Voelker. Monkey see, monkey do: A tool for tcp tracing and replaying. In *Proceedings of the USENIX Annual Technical Conference*, pages 87–98, Boston, MA, June 2004.

[37] M. Chesire, A. Wolman, G. M. Voelker, and H. M. Levy. Measurement and analysis of a streaming media workload. In *Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems (USITS)*, San Francisco, CA, Mar. 2001. Best paper.

[38] J. Chown, B. Pfaff, T. Garfinkel, K. Christopher, and M. Rosenblum. Understanding data lifetime in whole system simulation. In *Proceedings of the USENIX Security Symposium*, Aug. 2004.

[39] B. N. Chun, Y. Fu, and A. Vahdat. Bootstrapping a distributed computational economy with peer-to-peer bartering. In *Proceedings of the First Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, June 2003.

[40] J. R. Clark and W. L. Davis. A human capital pespective on criminal careers. *Journal of Applied Business Research*, 11(3):58–64, 1995.

[41] J. R. Crandall and F. T. Chong. Minos: Control data attack prevention orthogonal to memory model. In *Proceedings of the International Symposium on Microarchitecture*, Dec. 2004.

[42] P. Efstathopoulos, M. Krohn, S. VanDeBogart, C. Frey, D. Ziegler, E. Kohler, D. Mazières, F. Kaashoek, and R. Morris. Labels and event processes in the asbestos operating system. In *Proceedings of the ACM Symposium on Operating Systems Principles*, Oct. 2005.

[43] D. Ely, S. Savage, and D. Wetherall. Alpine: A user-level infrastructure for network protocol development. In *Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems (USITS)*, pages 171–183, San Francisco, CA, Mar. 2001.

[44] D. Ely, N. Spring, D. Wetherall, S. Savage, and T. Anderson. Robust congestion signaling. In *Proceedings of the 9th International Conference on Network Protocols (ICNP)*, pages 332–341, Riverside, CA, Nov. 2001.

[45] Entelligence^{TM}Content Control Server. Entrust. www.entrust.com/content-control.

[46] C. Estan, S. Savage, and G. Varghese. Automatically inferring patterns of resource consumption in network traffic. In *Proceedings of the ACM SIGCOMM Conference*, pages 137–148, Karlsruhe, Germany, Aug. 2003.

[47] N. Ferguson, D. Whiting, B. Schneier, J. Kelsey, S. Lucks, and T. Kohno. Helix: Fast encryption and authentication in a single cryptographic primitive. In T. Johansson, editor, *Fast Software Encryption*, volume 2887 of *Lecture Notes in Computer Science*, pages 330–346. Springer-Verlag, Feb. 2003.

[48] P. Ferguson and D. Senie. Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. RFC 2267, Jan. 1998.

[49] K. Fu, S. Kamara, and T. Kohno. Key regression: Enabling efficient key distribution for secure distributed storage. In *ISOC Network and Distributed System Security Symposium*, Feb. 2006.

[50] Y. Fu, J. S. Chase, B. Chun, S. Schwab, and A. Vahdat. Sharp: An architecture for secure resource peering. In *Proceedings of the 19th ACM Symposium on Operating System Principles (SOSP)*, Bolton Landing, NY, Oct. 2003.

[51] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Science*, 28:270–299, 1984.

[52] D. Gupta, K. Yocum, M. McNett, A. C. Snoeren, A. Vahdat, and G. M. Voelker. To infinity and beyond: Time-warped network emulation. In *Proceedings of the 3rd ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, San Jose, CA, May 2006.

[53] A. Hon, M. Fetterman, C. Clark, A. Warfield, and S. Hand. Practical taint-based protection using demand emulation. In *Proceedings of ACM EuroSys Conference*, Apr. 2006.

[54] T. Iwata and T. Kohno. New security proofs for the 3GPP confidentiality and integrity algorithms. In B. Roy and W. Meier, editors, *Fast Software Encryption*, volume 3017 of *Lecture Notes in Computer Science*, pages 427–445. Springer-Verlag, Feb. 2004.

[55] F. Junqueira, R. Bhagwan, A. Hevia, K. Marzullo, and G. M. Voelker. Surviving Internet Catastrophes. In *Proceedings of the USENIX Annual Technical Conference*, Anaheim, CA, Apr. 2005.

[56] F. Junqueira, R. Bhagwan, K. Marzullo, S. Savage, and G. M. Voelker. The phoenix recovery system: Rebuilding from the ashes of an internet catastrophe. In *Proceedings of the 9th USENIX Workshop on Hot Topics in Operating Systems (HotOS-IX)*, pages 73–78, Lihue, HI, May 2003.

[57] J. Kelsey, T. Kohno, and B. Schneier. Amplified boomerang attacks against reduced-round MARS and Serpent. In B. Schneier, editor, *Fast Software Encryption*, volume 1978 of *Lecture Notes in Computer Science*, pages 75–93. Springer-Verlag, Apr. 2000.

[58] V. Kiriansky, D. Bruening, and S. Amarasinghe. Secure execution via program shepherding, Aug. 2002.

[59] L. R. Knudsen and T. Kohno. Analysis of RMAC. In T. Johansson, editor, *Fast Software Encryption*, volume 2887 of *Lecture Notes in Computer Science*, pages 182–191. Springer-Verlag, Feb. 2003.

[60] T. Kohno. Attacking and repairing the WinZip encryption scheme. In B. Pfitzmann, editor, *Proceedings of the 11th ACM Conference on Computer and Communications Security*, pages 72–81. ACM Press, Oct. 2004.

[61] T. Kohno, A. Brodio, and kc claffy. Remote Physical Device Fingerprinting. In *Proceedings of the IEEE Symposium and Security and Privacy*, Oakland, CA, May 2005. Award paper.

[62] T. Kohno, A. Broido, and K. Claffy. Remote physical device fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, 2(2):93–108, April–June 2005.

[63] T. Kohno, A. Broido, and k. claffy. Remote physical device fingerprinting. In *IEEE Symposium on Security and Privacy*, pages 211–225. IEEE Computer Society, May 2005.

[64] T. Kohno, J. Kelsey, and B. Schneier. Preliminary cryptanalysis of reduced-round Serpent. In *Third AES Candidate Conference*, pages 195–211, Apr. 2000.

[65] T. Kohno, A. Stubblefield, A. D. Rubin, and D. S. Wallach. Analysis of an electronic voting system. In *IEEE Symposium on Security and Privacy*, pages 27–40. IEEE Computer Society, May 2004.

[66] T. Kohno, J. Viega, and D. Whiting. CWC: A high-performance conventional authenticated encryption mode. In B. Roy and W. Meier, editors, *Fast Software Encryption*, volume 3017 of *Lecture Notes in Computer Science*, pages 408–426. Springer-Verlag, Feb. 2004.

[67] M. Koletsou and G. M. Voelker. The medusa proxy: A tool for exploring user-perceived web performance. In *Proceedings of the 6th International Web Caching Workshop (WCW)*, Boston, MA, June 2001.

[68] D. Kostić, A. Rodriguez, J. Albrecht, A. Bhirud, and A. Vahdat. Using random subsets to build scalable network services. In *Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems (USITS)*, Seattle, WA, Mar. 2003.

[69] D. Kostić, A. Rodriguez, J. Albrecht, and A. Vahdat. Bullet: High bandwidth data dissemination using an overlay mesh. In *Proceedings of the 19th ACM Symposium on Operating System Principles (SOSP)*, Bolton Landing, NY, Oct. 2003.

[70] J. Ma, J. Dunagan, H. J. Wang, S. Savage, and G. M. Voelker. Finding diversity in remote code injection exploits. In *Proceedings of the ACM Internet Measurement Conference*, Rio de Janeiro, Brazil, Oct. 2006.

[71] J. Ma, K. Levchenko, C. Kriebich, S. Savage, and G. M. Voelker. Automated protocol inference: Unexpected means of identifying protocols. In *Proceedings of the ACM Internet Measurement Conference*, Rio de Janeiro, Brazil, Oct. 2006.

[72] J. Ma, G. M. Voelker, and S. Savage. Self-stopping worms. In *Proceedings of the ACM Workshop on Rapid Malcode (WORM)*, pages 12–21, Washington D.C., Nov. 2005.

[73] M. McNett and G. M. Voelker. Access and mobility of wireless pda users. *Mobile Computing and Communications Review (MC2R)*, 9(2), Apr. 2005.

[74] A. Mizrak, Y.-C. Cheng, K. Marzullo, and S. Savage. Fatih: Detecting and isolating malicious routers. In *Proceedings of the IEEE Conference on Dependable Systems and Networks (DSN)*, pages 538–547, Yokohama, Japan, June 2005. Award paper.

[75] D. Molnar, T. Kohno, N. Sastry, and D. Wagner. Tamper-evident, history-independent, subliminal-free data structures on EPROM storage -or- how to store ballots on a voting machine. In *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2006.

[76] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the slammer worm. *IEEE Security and Privacy*, 1(4):33–39, July 2003.

[77] D. Moore, C. Shannon, D. Brown, G. M. Voelker, and S. Savage. Inferring internet denial-of-service activity. *ACM Transactions on Computer Systems*, 24(2):115–139, May 2006.

[78] D. Moore, C. Shannon, and J. Brown. Code-red: a case study on the spread and victims of an internet worm. In *Proceedings of the ACM/USENIX Internet Measurement Workshop (IMW)*, Marseille, France, Nov. 2002.

[79] D. Moore, C. Shannon, G. M. Voelker, and S. Savage. Internet quarantine: Requirements for containing self-propagating code. In *Proceedings of the IEEE Infocom Conference*, pages 1901–1910, San Francisco, CA, Apr. 2003.

[80] D. Moore, G. M. Voelker, and S. Savage. Inferring Internet denial of service activity. In *Proceedings of the USENIX Security Symposium*, pages 9–22, Washington, D.C., Aug. 2001. Best paper.

[81] J. Newsome and D. Song. Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software. In *Proceedings of the Annual Network and Distributed Systems Security Symposium (NDSS)*, Feb. 2005.

[82] SureView$^{\text{TM}}$. Oakley Networks. www.oakleynetworks.com.

[83] C. Partridge, A. C. Snoeren, W. T. Strayer, B. Schwartz, M. Condell, and I. Castiñeyra. FIRE: Flexible intra-AS routing environment. In *Proceedings of the ACM SIGCOMM Conference*, pages 191–203, Stockholm, Sweden, Aug. 2000.

[84] C. Partridge, A. C. Snoeren, W. T. Strayer, B. Schwartz, M. Condell, and I. Castiñeyra. FIRE: Flexible intra-AS routing environment. *IEEE Journal on Selected Areas in Communication*, 19(3), Mar. 2001.

[85] Portauthority. PortAuthority Technologies. www.portauthoritytech.com.

[86] L. Qiu, V. N. Padmanabhan, and G. M. Voelker. On the placement of web server replicas. In *Proceedings of IEEE Infocom Conference*, Anchorage, AK, Apr. 2001.

[87] B. Raghavan and A. C. Snoeren. A system for authenticated policy-compliant routing. In *Proceedings of the ACM SIGCOMM Conference*, pages 167–178, Portland, OR, Sept. 2004.

[88] L. A. Sanchez, W. C. Milliken, A. C. Snoeren, F. Tchakountio, C. E. Jones, S. T. Kent, C. Partridge, and W. T. Strayer. Hardware support for a hash-based IP traceback. In *Proceedings of DARPA Information Survivability Conference and Exposition (DISCEX)*, Anaheim, CA, June 2001.

[89] N. Sastry, T. Kohno, and D. Wagner. Designing voting machines for verification. In *Usenix Security*, 2006.

[90] S. Savage. Sting: a tcp-based network measurement tool. In *Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems (USITS)*, pages 71–79, Boulder, CO, Oct. 1999. Best student paper.

[91] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. M. Voelker, and J. Zahorjan. Detour: a case for informed internet routing and transport. *IEEE Micro*, 19(1):50–59, Jan. 1999.

[92] S. Savage, N. Cardwell, D. Wetherall, and T. Anderson. Tcp congestion control with a misbehaving receiver. *ACM Computer Communications Review*, 29(5):71–78, Oct. 1999.

[93] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson. The end-to-end effects of internet path selection. In *Proceedings of the ACM SIGCOMM Conference*, pages 289–299, Cambridge, MA, Sept. 1999.

[94] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for IP traceback. In *Proceedings of the ACM SIGCOMM Conference*, pages 295–306, Stockholm, Sweden, Aug. 2000.

[95] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Network support for IP traceback. *IEEE/ACM Transactions on Networking*, 9(3):226–237, June 2001.

[96] A. Singh. Eraser: An Exploit-Specific Monitor to Prevent Malicious Communication Channels. Technical Report 04-28, Georgia Tech CERCS, 2004.

[97] S. Singh, C. Estan, G. Varghese, and S. Savage. Automated worm fingerprinting. In *Proceedings of the 6th ACM/USENIX Symposium on Operating System Design and Implementation (OSDI)*, pages 45–60, San Francisco, CA, Dec. 2004.

[98] A. C. Snoeren. Adaptive inverse multiplexing for wide-area wireless networks. In *Proceedings of the 4th IEEE Global Internet Symposium (GlobeCom)*, pages 1665–1672, Rio de Janiero, Brazil, Dec. 1999.

[99] A. C. Snoeren, K. Conley, and D. K. Gifford. Mesh based content routing using XML. In *Proceedings of the 18th ACM Symposium on Operating System Principles (SOSP)*, pages 160–173, Banff, Canada, Oct. 2001.

[100] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer. Hash-based IP traceback. In *Proceedings of the ACM SIGCOMM Conference*, pages 3–14, San Diego, CA, Aug. 2001. Best student paper.

[101] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, and W. T. Strayer. Single-packet IP traceback. *IEEE/ACM Transactions on Networking*, 10(6):721–734, Dec. 2002.

[102] A. C. Snoeren and B. Raghavan. Decoupling policy from mechanism in Internet routing. In *Proceedings of the 2nd ACM Workshop on Hot Topics in Networks (HotNets-II)*, pages 81–86, Cambridge, MA, Nov. 2003.

[103] G. E. Suh, J. W. Lee, D. Zhang, and S. Devadas. Secure program execution via dynamic information flow tracking. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2004.

[104] Content Alarm NW. Tablus Inc. www.tablus.com.

[105] K. Tati and G. M. Voelker. Shortcuts: Using soft state to improve dht routing. In *Proceedings of the 9th International Workshop on Web Content Caching and Distribution (WCW)*, Beijing, China, Oct. 2004.

[106] K. Tati and G. M. Voelker. On object maintenance in peer-to-peer systems. In *Proceedings of the International Workshop on Peer To Peer Systems (IPTPS)*, Santa Barbara, CA, Feb. 2006.

[107] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker. In search of path diversity in isp networks. In *Proceedings of the USENIX/ACM Internet Measurement Conference*, pages 313–318, Miami, FL, Oct. 2003.

[108] R. Teixeira, A. Shaikh, T. Griffin, and G. M. Voelker. Network sensitivity to hot-potato disruptions. In *Proceedings of the ACM SIGCOMM Conference*, Portland, OR, Sept. 2004.

[109] A. Vahdat, J. S. Chase, R. Braynard, D. Kostić, P. Reynolds, and A. Rodriguez. Self-organizing subsets: From each according to his abilities, to each according to his needs. In *Proceedings of the International Workshop on Peer To Peer Systems (IPTPS)*, Cambridge, MA, Mar. 2002.

[110] A. Vahdat, K. Yocum, K. Walsh, P. Mahadevan, D. Kostić, J. Chase, and D. Becker. Scalability and accuracy in a large-scale network emulator. In *Proceedings of the 5th ACM/USENIX Symposium on Operating System Design and Implementation (OSDI)*, Boston, MA, Dec. 2002.

[111] J. Viega, J. T. Bloch, T. Kohno, and G. McGraw. Token-based scanning for source code security problems. *ACM Transactions on Information and System Security*, 5(3):238–261, Aug. 2002.

[112] J. Viega, J. T. Bloch, Y. Kohno, and G. McGraw. ITS4: A static vulnerability scanner for C and C++ code. In *Sixteenth Annual Computer Security Applications Conference*, pages 257–267, Dec. 2000.

[113] M. Vrable, J. Ma, J. Chen, D. Moore, E. VandeKieft, A. C. Snoeren, G. M. Voelker, and S. Savage. Scalability, fidelity and containment in the potemkin virtual honeyfarm. In *Proceedings of the 20th ACM Symposium on Operating System Principles (SOSP)*, pages 148–162, Brighton, UK, Oct. 2005.

[114] W. Xu, S. Bhatkar, and R. Sekar. A unified approach for preventing attacks exploiting a range of software vulnerabilities. Technical report, SUNY Stony Brook, Aug. 2005.

[115] A. Yaar, A. Perrig, and D. Song. An endhost capability mechanism to mitigate DDoS flooding attacks. Proceedings of the IEEE Symposium on Security and Privacy, May 2004.

[116] X. Yang, D. Wetherall, and T. Anderson. A dos-limiting network architecture. In *Proceedings of the ACM SIGCOMM Conference*, Philadelphia, PA, Aug 2005.

[117] J. Yin, J.-P. Martin, A. Venkataramani, L. Alvisi, and M. Dahlin. Separating Agreement from Execution for Byzantine Fault Tolerant Services. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 253–267, New York, NY, USA, 2003. ACM Press.

[118] K. Yocum, E. Eade, J. Degesys, D. Becker, J. Chase, and A. Vahdat. Toward Scaling Network Emulation using Topology Partitioning. In *Proceedings of the 11th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, October 2003.

[119] H. Yu and A. Vahdat. Building Replicated Internet Services Using TACT: A Toolkit for Tunable Availability and Consistency Tradeoffs. In *Proceedings of the Second International Workshop on Advanced Issues of E-Commerce and Web-based Information Systems*, June 2000.

[120] H. Yu and A. Vahdat. Efficient Numerical Error Bounding for Replicated Network Services. In *Proceedings of the 26th International Conference on Very Large Databases (VLDB)*, September 2000.

[121] H. Yu and A. Vahdat. Combining Generality and Practicality in a Conit-Based Continuous Consistency Model for Wide-Area Replication. In *The 21st IEEE International Conference on Distributed Computing Systems (ICDCS)*, April 2001.

[122] H. Yu and A. Vahdat. The Costs and Limits of Availability for Replicated Services. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP)*, October 2001.

[123] H. Yu and A. Vahdat. Minimal Replication Cost for Availability. In *Proceedings of the ACM Principles of Distributed Computing*, July 2002.

[124] H. Yu and A. Vahdat. Consistent and automatic service regeneration. In *Proceedings of the 1st ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco, CA, Mar. 2004.

[125] Y. Zhang and V. Paxson. Detecting stepping stones. In *Proc. 9th USENIX Security Symposium*, Aug. 2000.