ForNet: A Distributed Forensic Network

Kulesh Shanmugasundaram
Polytechnic University

1 Problem and Motivation

Security fails. Thousands of reported security breaches, worms, and viruses attest to this fact. A majority of security breaches are premeditated acts to subvert computing resources and use them to steal identities, share contraband materials, send Spam, or commit fraud across the Internet. With the wide-spread use of intrusion detection systems (IDS) and firewalls, information systems are more tightly guarded than ever before. Ironically, threats against the information systems are rapidly increasing. What makes materialization of a threat even worse is our inability to conduct a proper postmortem to identify the perpetrators. Most of the time we are alerted to a breach weeks or even months after it has taken place. By that time, valuable digital evidence we need is either subverted by the attackers or never logged by security mechanisms because they will not have realized that the attack has occurred. A postmortem helps us attribute an attack to its perpetrators and reveals how, why, and when the security mechanisms failed. Currently most perpetrators go unidentified! This is one of the reasons for the increase in attacks. Presence of a strong attribution mechanism will deter cyber crimes. Therefore, we need to complement the security mechanisms with forensic capabilities to monitor, capture, and store digital evidence. The question is: why aren't there any forensic capabilities in networks?

In order to build forensic capabilities into network infrastructure we must first overcome some difficult technical and socio-economic challenges.

Technical Challenges

1. *Data Collection:* Unlike an IDS, a forensic system must monitor and collect a wider range of data because we do not know what might be useful evidence a priori. Furthermore, collecting data at the entry points of a network is not sufficient as this will not help trace insider attacks. Therefore, one of the major challenges is in building a system that can monitor and collect a wide range of data from multiple points on networks and cope with ever increasing network speeds.

2. *Data Retention:* Data collected by the system must be retained for months as we do not know when the system will be called upon for a postmortem. Storage requirements for archiving network data can be overwhelming. For example, the daily network traffic for a moderately large network of $5,000$ hosts would run into Peta bytes! In addition, data collected from multiple points must be transfered to a data warehouse without flooding the network. Therefore, major challenge in data retention is to reduce the storage requirements to a feasible amount without sacrificing valuable forensic information.

3. *Data Retrieval:* During a postmortem the system must first locate the necessary data (or evidence) across wide area networks. We should note that the volume of evidence could be overwhelming. The major challenge here is in developing protocols to locate and transfer evidence across the Internet. We also need to

develop intuitive user interfaces for investigators to efficiently scour through large volumes of network data.

Socio-Economic Challenges

4. *Privacy:* Network monitoring and user privacy will always be at odds. To support better postmortems we need to monitor networks thoroughly which may make users uncomfortable. However, public approval is critical for wide spread deployment of such a system. Therefore, we must strike a good balance between supporting forensics and preserving user privacy.

5. *Economics:* Widespread deployment of the system also depends on the economic benefits to service providers. The challenge here is in justifying the cost to service providers.

Recent advances in data streaming algorithms, storage capacity, and computing power provides us the fundamental tools to address the technical challenges. Increase in cyber crimes, litigation where service providers are held liable for the damages, and strict compliance legislations create the economic incentives. We believe the lack of forensic capabilities in networks can no longer be ignored. This paper presents an overview of ForNet, a system developed to address this shortcoming. We refer the readers to [9, 10, 11] for an in-depth look at ForNet.

2 Background and Related Work

Distributed denial of service attacks create large uncorrelated network flows towards a set of hosts. Researchers have proposed some clever solutions to the problem of tracing IP packets back to their source in the presence of spoofed packets. Most of this work falls under two categories: one in which no extra network packets are generated [2, 3, 5, 6, 8, 12, 13] and the other in which a few extra network packets are generated [1, 7]. In both cases the source IPs of routers along the path of a set of packets are encoded (in the packets themselves or in separate packets) by the routers and decoded at the victims of denial of service attacks. A related problem is tracing connection chains. Attackers often obscure their identity and location by forming a connection chain by logging into a set of compromised systems or stepping stones before attacking a target. The method proposed in [14] creates "thumb-prints" of connections using packet content which can be compared to determine whether two connections contain the same text and are therefore likely to be part of the same connection chain. The method, however, fails when the connections are encrypted. To address this problem Zhang & Paxson [15] proposes an algorithm that does not rely on traffic content but instead relies on packet sizes and inter arrival times to identify stepping stones.

Currently network operators rely on IDS and firewall logs for forensics. However, IDS and firewalls do not monitor all network activities but only a limited subset deemed a security risk. This a priori assumption narrows the field of monitoring and limits the value of such logs in postmortems. For example, an IDS cannot be used to find the victims of a fast spreading worm. This is because the worm spread faster than we could update IDS signatures. Since the signature was not available at the time of initial infection the IDS would not have kept historical data necessary to identify the victims post facto. Another example is the simple activity of uploading a file, which could be a crime if the file contains trade secrets. Since IDS do not monitor such basic activities they would not log any information regarding this act, and not be useful for forensics.

Additionally, there are also packet capture tools that strives to capture raw network traffic in a LAN and store it for a few days [4]. This brute force method neither scales for monitoring wide area networks nor for storing potential evidence for weeks, months, or even years as required by most postmortems.

3 Approach and Uniqueness

ForNet is different from the previous systems in many ways. First of all, it can help with the postmortem of any security incident including insider attacks. It can also store potential evidence for months, which is much longer than any existing solution. This is because ForNet stores a succinct representation of raw network data, called *synopses*. In addition, ForNet uses a novel *cascading data collection* strategy that scales well across wide area networks. Finally, ForNet makes users part of the process by explicitly advertising the monitoring and privacy policies of networks.

3.1 Architecture of ForNet

ForNet has two major functional elements: *SynApp* and *Forensic Server*. SynApp is embedded into networking components, such as routers, switches, and gateways, to provide a secure, scalable, and modular environment for collecting data. A Forensic Server handles all postmortem queries for a particular domain, similar to a DNS server that handles a domain's name resolution. The collection of network components instrumented with SynApps are interconnected with Forensic Servers to form a hierarchy. In this hierarchy, all SynApps within a domain form a network and are associated with the Forensic Server for the domain. The Forensic Server functions as an archiver for data collected by the SynApps. In addition, it also advertises monitoring and privacy policies of the domain. Finally, the Forensic Server receives postmortem queries from outside the domain boundaries, authenticates them and either responds to them itself or passes them along to the appropriate SynApps. Responses to queries are evaluated against the privacy policy of the domain to ensure compliance. Then responses are certified by the Forensic Server and sent back to the originators.

3.2 Data Collection

Abstractly speaking, the Internet is a gigantic state machine. To support forensics we need to know the precise state of this machine at any given time. State transitions of the Internet can be characterized by links/connections between the nodes, content traversing the links, various protocol mappings, and the aggregates state transitions generate over time. These are the four fundamental types of information we need to support postmortems. Storing raw network data to infer this information would simply overwhelm any storage and the solution would not scale well either. ForNet employs two novel ideas to overcome the challenges in data collection. The first, *synopses* collect a compact summaries of network data instead of raw network data itself. The second, *cascading data collection* uses the ubiquitous presence of SynApps in networking components to distribute data collection.

3.2.1 Synopses A synopsis is a combination of data structures and algorithms that can represent a set of data in a succinct form and answer queries about the data with a predetermined confidence level using the succinct representation of the data. An example of a simple synopsis is a Bloom filter. It can represent a set of elements in a compact form and answer membership queries with a predetermined false positive rate. A good synopsis also allows for trade-offs between computation, memory, and accuracy. For example, in a Bloom filter increasing the number of hash functions increases the required computation per element but it also increases the accuracy of its answers. Similarly, memory requirements can also be traded off. This property of synopses allows SynApps to be tightly integrated into resource constrained network components. Note that synopses may not be able to recover raw data entirely, however they preserve enough information to carry out a postmortem.

3.2.2 Cascading Data Collection Data captured by SynApps should contain enough information to facilitate both macroscopic and microscopic views of security incidents. However, capturing data to create micro-

scopic views at the core of Internet is a resource intensive task due to the high volume of traffic. ForNet takes advantage of the hierarchical nature of the Internet and the ubiquity of SynApps to address this problem. More precisely, SynApps at the leafs of the hierarchy collect fine-grained data necessary for microscopic views whereas SynApps closer to the core collect more coarse information necessary for macroscopic views. For instance, a switch at a subnet collects packet payload digests (link-content) whereas the upstream router collects only connection records (links), and the edge router at the service provider simply collects traffic statistics (aggregates). We call this arrangement of distributed collection *cascading data collection*.

3.3 Data Retention

Use of synopses allows us to retain potential evidence at SynApps until it can be sent to a Forensic Server for archival. Synopses also allow us to transfer evidence via existing network infrastructure without flooding the network.

3.4 Data Retrieval

During a postmortem a forensic analyst uses ForNet's client component, Panorama, to construct appropriate queries. The queries are then sent to the Forensic Server of the network where the investigation begins. A query is a collection of one or more events in a set of networks within a time interval. A query may partially describe an event and request that the details be filled in by ForNet. A query may be sent to the Forensic Server of a domain or they can be propagated to Forensic Servers in neighboring networks for gathering additional information. In either case ForNet routes the queries to the appropriate Forensic Servers and returns the responses securely. The use of synopses makes query processing more complex than querying a relational database. Normally, an analyst should be aware of which synopsis stores what kind of data and how to query it. ForNet's query processor hides these details from analysts by a novel technique called *coalescing of synopses*. Each module exports an XML interface to ForNet's query processor. Given a query, the query processor automatically determines which synopses should be queried in what order. The query processor then generates a query plan and executes it. Therefore, an analyst can fill out a form with whatever information is available about an incident and ask ForNet to fill in the missing information.

4 Results and Contributions

A prototype of ForNet is currently deployed in our campus network monitoring thousands of hosts. Thus far we have developed two novel synopses to address content related queries.

Hierarchical Bloom Filter: A Bloom filter can answer queries about the exact payload in a packet. HBF generalizes a Bloom Filter by being able to answer not only about exact payloads of a packet but also any excerpts of payload or excerpts that span multiple packets. HBF also reduces the storage requirements by a factor of at least 20. We demonstrate the practicality and efficacy of HBF with ForNet by tracking the spread of MyDoom virus in our campus network [9].

Flow Content Characterization: We developed flow content characterization synopsis to robustly identify content types of network flows. Robust identification is achieved by characterizing content types based on the statistical properties of sampled payloads. Currently this synopsis can distinguish between the following popular content types: encrypted, compressed, text, audio, video, or JPEG and can easily be extended to other content types as well. Again we demonstrated the use of this synopsis in ForNet by identifying resource abusers in our campus network [10].

5 Conclusion and Future Work

We described a system, ForNet, that monitors, collects, and retains data to support network forensics on the Internet. Unlike

packet loggers, ForNet creates compact summaries of raw network data known as synopses. Synopses capture enough information to perform an effective postmortem while saving many orders of magnitude in storage space compared to raw network data. ForNet also implements a distributed collection strategy known as cascading collection which makes ForNet scale across wide area networks. Furthermore, we introduced the notion of advertising privacy and monitoring polices of a network domain. The forensic server of the domain enforces the monitoring policy by way of SynApps and complies with the domain's privacy policy when answering postmortem queries. Finally, XML based query routing protocols, coalescing of synopses, and a user interface together allow an analyst to locate evidence relating to an incident efficiently and transparently.

Our future work focuses on the development of synopses for various types of data on the Internet. We are currently testing a novel synopsis that keeps track of temporal changes in packet size distribution and inter-arrival times on a per-flow basis. A synopsis to capture various network mappings such as protocol mappings and topologies is also under investigation. On the deployment side, we are extending the reach of ForNet into service provider networks and into other universities.

Quis Custodiet Ipsos Custodes?

References

[1] S. M. Bellovin, M. Leech, and T. Taylor. ICMP traceback messages. In *Internet Draft*. IETF, Oct 2001.

[2] H. Burch and B. Cheswick. Tracing anonymous packets to their approximate source. In *Proc. USENIX LISA*, Dec 2000.

[3] D. Dean, M. Franklin, and A. Stubblefield. An algebraic approach to IP traceback. In *Proceedings of NDSS*, Feb 2001.

[4] Sanstorm Enterprises. Netintercept. http://www.sandstorm.com/.

[5] I. Hamadeh and G. Kesidis. Packet marking for traceback of illegal content distribution. In *Proceedings of International Conference on Cross-Media Service Delivery (CMSD)*, May 2003.

[6] I. Hamadeh and G. Kesidis. Performance of IP Address Fragmentation Strategies for DDoS Traceback. In *Proceedings of IEEE IPCOM*, Oct 2003.

[7] A. Mankin, D. Massey, C. L. Wu, S. F. Wu, and L. Zhang. On design and evaluation of "intention-driven" ICMP traceback. In *Proc. IEEE International Conference on Computer Communications and Networks*, Oct 2001.

[8] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for IP traceback. In *Proceedings of the 2000 ACM SIGCOMM Conference*, Aug 2000.

[9] K. Shanmugasundaram, H. Bronnimann, and N. Memon. Payload attribution via hierarchical bloom filters. In *11th ACM Conference on Computer and Communications Security (CCS'04)*, Oct 2004.

[10] K. Shanmugasundaram, M. Kharrazi, and N. Memon. Nabs: A system for detecting resource abuses via characterization of flow content type. In *In the proceedings of the 2004 Annual Computer Security Applications Conference*. IEEE, December 2004.

[11] K. Shanmugasundaram, N. Memon, A. Savant, and H. Bronnimann. Fornet: A distributed forensics system. The Second International Workshop on Mathematical Methods, Models and Architectures for Computer Networks Security, May 2003.

[12] 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 *ACM SIGCOMM*, Aug 2001.

[13] D. Song and A. Perrig. Advanced and authenticated marking schemes for IP traceback. In *IEEE Infocomm*, 2001.

[14] S. Staniford-Chen and L.T. Heberlein. Holding intruders accountable on the internet. Proceedings of the 1995 IEEE Symposium on Security and Privacy, 1995.

[15] Y. Zhang and V. Paxson. Detecting stepping stones. In *Proceedings of the 9th USENIX Security Symposium*, Aug 2000.