Poster: Towards a Collaborative and Retrospective Understanding of Security Events with Retro-Future

Calvin Ardi, John Heidemann
USC/Information Sciences Institute
{calvin,johnh}@isi.edu

Gina Fisk, Mike Fisk
Los Alamos National Laboratory
{gina,mfisk}@lanl.gov

Christos Papadopoulos
Colorado State University
christos@cs.colostate.edu

1 INTRODUCTION

All future interesting security events have already happened. The fundamental problem with security is *we only know an event was interesting after the fact*; as we understand its novelty through an unexpected impact or the need for a new response. Some of the most interesting exploits are zero-day attacks and insider threats. These can only be addressed through recovery and mitigation, not prevention. Unfortunately, today, the data needed for forensics analysis, recovery, and mitigation is unavailable to us because the event has always already happened and today's network is ephemeral.

To enable post-event understanding, we need to be able to rewind time to obtain necessary data and identify recoverable state. We need an efficient and effective Internet Digital Video Recorder (DVR) to record the state of the Internet in multiple places, capturing multiple data streams with enough fidelity that we can replay and recreate novel events *after* they happen. Previous systems have explored storage and search of specific data (BGP, DNS, or packet data) at single sites [2, 6, 8], and sometimes require expensive hardware [3]. Our approach will provide an integrated view of the Internet across several protocols and use a combination of commodity hardware, parallelism, and abstraction to reduce total cost-of-ownership. By providing much better forensics, response, and recovery tools than are possible today, we hope to lessen the impact and close the window of "interesting" events.

Post-event understanding also requires collaboratively sharing relevant data *across* organizations as events touch and cross many networks. Each organization has its own unique and incomplete view of the Internet and local network: data sharing during and after an event helps expedite the response process by collectively increasing the global knowledge and effort against an attack. Prior work [1, 4, 5] has looked at the creation of logical frameworks for enabling and implementing data sharing. Our goal is to leverage this existing body of work and provide organizations with the tools and controls to manage the risk-benefit trade-off in data sharing while pursuing advances in network security research and understanding.

We see three important challenges in providing a DVR that is able to share data between organizations: the system must accommodate varying policy and privacy constraints, and be efficient and cost-effective. First, network data capture and sharing is only possible when it meets policy and privacy constraints of all the organizations involved. We must accommodate permission and privacy constraints by decoupling capture from use and supporting semi-trusted collaboration. Second, we need new techniques to maximize the effective history through multi-resolution abstraction and custom pruning of data streams. Finally, we must exploit commodity hardware for parallel search and distributed storage both to leverage their low cost and store data near its collection point to minimize operational costs.

This poster will describe Retro-Future, an Internet DVR system that provides post-event understanding with time travel and cross-site data sharing (§2). Retro-Future collects and archives several relevant data streams (packets, BGP, DNS, and others), provides new approaches to analyze the data at large scale, uses abstraction techniques to manage timescales and representations, builds situational understanding across these data streams, and provides privacy and risk management to support sensitive data and semi-trusted sharing across organizations. We present an ongoing case study (§3) in how time travel and data sharing in Retro-Future is used to detect bots and botnet activity across networks.

2 POST-EVENT UNDERSTANDING WITH THE RETRO-FUTURE SYSTEM

Our Retro-Future system is an Internet DVR designed to enable post-event understanding by recording, replaying, and sharing details of security events at one or across many sites. We next give a brief overview on how we design Retro-Future to enable information sharing by accommodating policy and privacy constraints and be efficient and cost-effective.

Enabling Information Sharing by Respecting Policy and Privacy: The goal of controlled information sharing across organizations is to facilitate efficient and collaborative post-event understanding while balancing the privacy and exposure risks associated with such sharing.

We enable sharing in Retro-Future with principled risk and privacy management using a query/response system: data owners share specific information about their data (within their policy and privacy constraints) in response to queries

ACM acknowledges that this contribution was authored or co-authored by an employee, contractor, or affiliate of the United States government. As such, the United States government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for government purposes only.

Posters-Demos '18, August 20–25, 2018, Budapest, Hungary

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5915-3/18/08...\$15.00

<https://doi.org/10.1145/3234200.3234249>

made by remote users. Data owners first set and modify accesses based on their trust relationships and sharing policies with other organizations. Remote queries made to this data are then *moderated* such that more specific queries are more likely to be answered than broad queries. Finally, queries are remotely processed and its response further limited to *control disclosure*, ensuring that we protect users' privacy in the data and maintain the owners' control of their data.

Enabling Time Travel with Limited Storage and Money:

To meet the goals of time travel in post-event understanding with the constraints of limited storage and financial resources, we maximize the utility of what is stored and avoid the use of expensive hardware and high bandwidth use.

Retro-Future maximizes the utility of what is being stored with two approaches: storing network traffic at multiple resolutions (full packets, packet headers, flows) and using knowledge about applications, protocols, or sites to cleverly store the essential data. Applying knowledge about what and how it is collected allows us to reduce redundancy further: route changes and common DNS queries can be compressed and communication with known-good sites can be pruned.

Retro-Future is cost-effective by leveraging commodity hardware and distributed indexing, search, and processing. In addition to using existing frameworks and storage (Apache Hadoop and Hadoop Distributed File System), Retro-Future integrates indexing and search by continuously indexing multiple network and system data types through *timefind*, allowing users to run search queries on down-selected data in a given time range. *timefind*, available at [7], operates in tandem with existing packet and log capture systems, supporting indexing and search for 18 data types, including .pcap, Windows/Linux system log, and firewall data. By providing a simple and unified interface across heterogeneous data types, users can use *timefind* to quickly pull together all relevant data, even on different backends and systems.

3 USING RETRO-FUTURE IN NETWORK SECURITY APPLICATIONS

With a broad set of challenges, we next describe a challenge scenario to focus our work and quantify the benefits of data sharing and time travel using Retro-Future: detecting bots and botnet activity.

Detecting DGA-based Bots and Botnet Activity: Today, organizations can detect DGA-based bots and botnet activity using software like BotDigger [9] by analyzing DNS traffic. While BotDigger works well in detecting bots and botnet activity for large organizations (like Colorado State University, where it originated), BotDigger becomes much less sensitive for organizations that are not as large and diverse as CSU: smaller or less diverse organizations might not be able to detect as much, or any, botnet activity.

Our insight is that cross-site sharing and time travel with Retro-Future allows organizations to improve their detection

Figure 1: Detecting suspect C&C domains and IPs at each site independently.

Figure 2: Improving the sensitivity of BotDigger's detection with controlled data sharing between sites over time.

of botnet activity. Cross-site data sharing allows larger, more diverse organizations to share sensitive data securely with smaller or less diverse organizations to help the latter detect malicious activity on their networks. Time travel helps organizations process DNS data effectively with Retro-Future's pipeline for collecting, searching, and processing DNS data.

We first ask if sites can detect malicious activity on their own with BotDigger, and then if their detection performance is improved when they share. We run BotDigger over roughly 2 years (2016-Jan-01 to 2017-Dec-31) at CSU (20k hosts, 5.2 B queries/month), USC/ISI (158, 46.2 M), and LANL (26k, 3.3 B). BotDigger looks at local hosts' DNS queries, clustering together and labeling queried domains and resolved IPs as suspect based on their linguistic features.

Figure 1 shows us that sites can sometimes detect malicious activity on their own. CSU, a large and diverse campus, consistently sees activity (# of detected domain/IP sets) over the two-year period. Less diverse sites like LANL and USC/ISI sometimes sees zero activity for long periods of time.

Figure 2 shows us that with Retro-Future's data sharing, sites improved their detection performance. Each site showed an increase in amount of activity seen, adding an additional 1–15 suspect IP/domain sets previously undetected at each corresponding site.

4 CONCLUSIONS

This poster has described our initial steps towards a collaborative and retrospective understanding of security events with Retro-Future. Retro-Future is our Internet DVR system that enables post-event understanding with time travel and cross-site data sharing, allowing organizations to collect, archive, and securely share relevant data for large-scale analysis. We've presented how sharing can help in an ongoing, long-term case study in detecting malicious activity. Our tools and source code are available on our website at <https://ant.isi.edu/retrofuture> and we continue to improve upon their performance and capabilities.

REFERENCES

- [1] Mark Allman and Vern Paxson. 2007. Issues and Etiquette Concerning Use of Shared Measurement Data. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC '07)*. ACM, New York, NY, USA, 135–140. <https://doi.org/10.1145/1298306.1298327>
- [2] C. J. Antonelli, M. Undy, and P. Honeyman. 1999. The Packet Vault: Secure Storage of Network Data. In *Proceedings of the 1st Conference on Workshop on Intrusion Detection and Network Monitoring - Volume 1 (ID'99)*. USENIX Association, Berkeley, CA, USA, 11–11. <http://dl.acm.org/citation.cfm?id=1267880.1267891>
- [3] Arbor Networks. 2013. PeakFlow. <http://www.arbornetworks.com/>. (2013). <https://web.archive.org/web/20130728070205/http://www.arbornetworks.com:80/products/peakflow/sp>
- [4] K. Claffy and E. Kenneally. 2010. Dialing Privacy and Utility: A Proposed Data-Sharing Framework to Advance Internet Research. *IEEE Security Privacy* 8, 4 (July 2010), 31–39. <https://doi.org/10.1109/MSP.2010.57>
- [5] Gina Fisk, Calvin Ardi, Neale Pickett, John Heidemann, Mike Fisk, and Christos Papadopoulos. 2015. Privacy Principles for Sharing Cyber Security Data. In *Proceedings of the IEEE International Workshop on Privacy Engineering*. San Jose, California, USA. <https://doi.org/10.1109/SPW.2015.23>
- [6] Stefan Kornexl, Vern Paxson, Holger Dreger, Anja Feldmann, and Robin Sommer. 2005. Building a Time Machine for Efficient Recording and Retrieval of High-volume Network Traffic. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement (IMC '05)*. USENIX Association, Berkeley, CA, USA, 23–23.
- [7] LANL and USC/ISI. 2016. timefind and indexer. (2016). <https://ant.isi.edu/software/timefind/>
- [8] Steven McCanne and Van Jacobson. 1993. The BSD Packet Filter: A New Architecture for User-level Packet Capture. In *Proceedings of the USENIX Winter 1993 Conference Proceedings on USENIX Winter 1993 Conference Proceedings (USENIX'93)*. USENIX Association, Berkeley, CA, USA, 2–2.
- [9] Han Zhang, Manaf Gharaibeh, Spiros Thanasoulas, and Christos Papadopoulos. 2016. BotDigger: Detecting DGA Bots in a Single Network. In *Proceedings of the International Workshop on Traffic Monitoring and Analysis (TMA '16)*. IFIP.