Senior Thesis

Hannah Atmer

I. Introduction

Siteswap is a notation system for juggling patterns. Analogous to musical notation. This notation system is closely integrated with algorithms that determine the set of possible patterns that can be juggled by a human being. This paper investigates the application of similar principles to peer-to-peer system design. This paper presents a peer-to-peer system that leverages these principles in order to obscure the network traffic fingerprint of the system by using dynamic routing patterns.

II. Siteswap Algorithm

siteswap generation algorithm and explanation here

proof of mathematical soundness of notation and how it can be translated into a dynamic routing algorithm for computer systems. Because the algorithm produces verifiable systemic checksums while remaining adaptive to churn. The key is that the algorithm specifies that a pattern is only valid if the number of network participants is acceptable for the specific pattern being used for routing at that time by the system. corresponding to number of items (network participants)

At every point in time, nodes in different stages of communication with other nodes Takes certain amount of time from initiation to completion of communication. This is equivalent to a throw/catch event

nodes generate possible patterns, perform summation and division, to verify correctness of current routing pattern

III. Routing Algorithm

Applying the siteswap algorithm to system design enables a state machine model of routing tables

available transitions within the state machine are computable, can be proposed to group using traditional 2PC methods built into grouping protocols

The number of nodes in the group known by each node because a specific siteswap routing pattern can only exist on that number of nodes. The number of nodes in the group informs seed values of the siteswap generation algorithm. Any node in the system can propose a change the routing pattern if the number of nodes change (evident as a hole in the pattern) or if security settings prompt an activation of the systems inherent polymorphic capabilities.

Whether or not multiplexing siteswaps can be included in the possible routing pattern generation algorithm is specified in system configuration. This enables an increased variety of available patterns, but only for systems that can use the IP protocol's broadcast functionality. Use of the broadcast functionality presents added security risks since broadcasting systems are relatively unusual on the internet today.

The method for generating the routing table: number of nodes in group = average, expand randomly, pairwise modifications, shuffle

IV. Design Considerations

The result is a polymorphic communication protocol for P2P systems with organized grouping

a) Failure Recognition: A node failure is equivalent to a drop, the pattern gets a hole, all later nodes see the hole.

b) Failure Recovery: Every node knows the current pattern, this pattern repeats until a new pattern proposed and accepted by the group. The first nodes to see hole propose a new pattern with 1 fewer nodes in it and initiate a transition.

c) Other: A new node added to the system becomes a full group member in as much time as it takes for the system to propose and accept a new pattern.

State transitions of the routing table allow for structural changes of groups.

latencies/timing misc. and grouping optimizations discussion

Easily added distributed checksum (failure indicator). This is useful for any application in which a single-bit or single-byte distributed checksummable, failure aware, and dynamically modifiable protocol can be used.

V. System Implementation

Congregate[7] is a peer-to-peer system that implements the protocol described in this system. The current state of development is a true-to-specs implementation of the Scatter[6] distributed system. It uses Lamport's BPCon[1][2] system design for replication. It will be modified to use the siteswap routing protocol specified in this paper, and the efficiency of the protocol will be compared to the efficiency of the original routing protocols.

graphs and other metrics comparing the two routing protocols (AB testing)

VI. Discussion

This paper presents the theory, algorithms and example implementation of a mechanism of enabling polymorphic network traffic fingerprints for P2P systems.

A major problem for distributed systems that are intended to evade detection is the action of calling home to the command and control of the system. The methods used to date have not resolved the problem of how to request updates from a centralized authority without creating obvious traffic patterns.

The IP addresses of the command and control must be hardcoded somewhere, and historically DNS has been used to fetch the IP address. This leads to ”DNS sinkholing” in which the DNS service is targeted by authorities in order to disable the system.[4] More recently, the IP address of the command and control server has been posted on public blockchains such as the Bitcoin network, thus making it resistant to sinkholing and easy to modify. But using a blockchain for command and control creates a clear traffic pattern of connecting to the distributed system that supports the blockchain, so a better solution is the use of intelligent grouping to evade detection while still receiving command-and-control updates from the system.

VII. Analysis

Intelligent distributed grouping algorithms allow groups of nodes to organize into local groups while remaining resilient to churn. The routing algorithms in this paper enable adaptable grouping and are based on criteria to make traffic appear normal. The implicit signaling intrinsic to the siteswap-inspired algorithm allows for distributed group consensus about the routing table state transitions as a byproduct of normal system network traffic.

The traffic pattern described in this paper is more lightweight than consensus-based grouping algorithms like Paxos and its derivative BPCon. Using state machine based grouping and routing protocols means that the number of nodes in group is dynamic, and that a variety of protocols from the transport level of the OSI model can be used to masquerade the system as an group of machines that interact as part of normal business operations. For example, a group of 10 machines might be composed by two that communicating over ssh, another two transferring data over RDP, one acting as an HTTP server, and the remaining 5 masquerading as HTTP clients to that server. The dynamic routing state machine enables dynamic membership and failure awareness for this group. The nature of the siteswap protocol dictates that the group will be naturally partitioned into subgroups that can adaptively select appropriate cover protocols intra-group communication.

Countermeasures can be taken to prevent this type of covert communication between nodes, but these computational resources necessary for system discovery as well as for a takedown are prohibitive. Pattern finding in network traffic, from a LAN scale to internet backbone scale, requires significant computational resources. There are many different possible siteswaps patterns for any given number of nodes, so the network traffic fingerprinting generated by the system is not static. HTTPS encryption conceals the network traffic in the massive amount of HTTPS traffic on the internet, and protocol masquerading makes it even more difficult to detect that the system maintains a covert coordination channel via a dynamic routing table. DNS sinkholing is also difficult because the system will naturally create temporarily independent local groups that will detect and adapt to failure in any single node. Only an adversary with the ability to massively the disrupt the system as a whole would be able to use techniques resembling a DNS takedown.

VIII. Conclusion

The system proposed in this paper implements efficient polymorphic grouping and routing for P2P systems.

References

[1] L. Lamport, *Byzantizing Paxos by Refinement*, 2011

[2] L. Lamport, *Practical byzantine fault tolerance and proactive recovery*, 2002

[3] L. Glendenning et al, *Scalable Consistency in Scatter*, 2011

[4] Y. Nadji et al, *Beheading Hydras: Performing Effective Botnet Takedowns*, 2013

[5] J. Buhler and R. Graham *Juggling Patterns, Passing, and Posets*, 2005.

[6] D. Perkins *Multi-person Siteswaps*, 2001.

[7] https://github.com/hatmer/python-congregate