# A Platform for Finding Attacks in Unmodified Implementations of Intrusion Tolerant Systems

Hyojeong Lee (student), Charles Killian, Cristina Nita-Rotaru
*Department of Computer Science, Purdue University*
{*hyojlee, ckillian, crisn*}*@purdue.edu*

Jeff Seibert
*MIT Lincoln Laboratory*
*jeffrey.seibert@ll.mit.edu*

## 1 Introduction

Intrusion tolerant systems (ITS) introduce a new paradigm in designing secure systems in that they provide correct operations even if a fraction of components are compromised and controlled by adversary. Many ITS are designed to meet both of safety and liveness when the network is stable. However, previous work [1] has shown attacks that can degrade the performance of ITS implementations extensively so that they are not usable anymore.

Despite the importance in usability of ITS, finding such attacks in unmodified implementations is a tedious task given the difficulty of debugging distributed systems in general and the complexity and subtleties of ITS.

We propose Turret, a platform for automatically finding performance attacks in unmodified implementations of ITS running in realistic environments. We focus on implementations running in realistic environments because they often include optimizations not considered at the design phase and the complexity leads to unexpected scenarios observed only in real environments. We focus on performance attacks since today no distributed system can be expected to be practical without maintaining some level of performance in stable networks. Finally, we focus on ITS because they are specifically designed to tolerate compromised participants.

## 2 Turret

Turret leverages virtualization to allow the unmodified user binary to run natively in the user-specified operating system and uses a well-known network emulator to tunnel the network traffic. Turret finds attacks conducted through message content and delivery manipulation by using a malicious protocol proxy implemented in the network emulator.

The malicious proxy intercepts messages sent by a corresponding node and conducts malicious actions based on the message type and the message format. Malicious delivery actions we consider include delaying, dropping, diverting and duplicating messages. In message contents manipulation, instead of random bit flipping, we modify value of message fields according to the type of the message and the fields to enable more meaningful lying.

In directing malicious proxies, Turret uses a controller that takes two different approaches to find attacks, brute force and a greedy approach. In the brute force approach, the controller generates all possible single malicious actions and combinations of malicious actions. Then it tries each strategy to find attacks that degrade performance. A limitation of this approach is that the number of possible combinations is practically unlimited. We address this limitation by implementing a greedy approach based on our previous work [4] adapted in a virtualized environment. Specifically, we added functionality that allows us to take a snap shot of the entire system consisting of several virtual machines, including the network conditions and to be able to stop, resume and rollback the execution.

## 3 Results

We demonstrate Turret, by applying it to several ITS, PBFT [3], Steward [2], and Prime [1]. Turret found, on the reference implementations of PBFT and Steward, a total of 18 protocol-level performance attacks, 16 of which were not previously reported to the best of our knowledge. We also use Turret to help us to develop a robust implementation of Prime and found several implementation-level performance attacks.

## References

[1] AMIR, Y., COAN, B., KIRSCH, J., AND LANE, J. Byzantine replication under attack. In *In DSN* (2008).

[2] AMIR, Y., DANILOV, C., DOLEV, D., KIRSCH, J., LANE, J., NITA-ROTARU, C., OLSEN, J., AND ZAGE, D. Scaling byzantine fault-tolerant replication to wide area networks. In *In DSN* (2006).

[3] CASTRO, M., AND LISKOV, B. Practical byzantine fault tolerance. In *OSDI* (1999).

[4] LEE, H., SEIBERT, J., KILLIAN, C., AND NITA-ROTARU, C. Gatling: Automatic attack discovery in large-scale distributed systems. In *NDSS* (2012).