Branch Prediction Analysis (BPA) attack is one of the most significant Side-Channel Attack (SCA). It causes serious issues on a machine hosting multiple services by exploiting shared resources. The current state of the art cloud technology provides a level of isolation by hosting processes on different VMs (Virtual Machines). Still, the scope of exploitation does not get eliminated even in the virtualization environment. The severity of the BPA attack and its normal-looking attack detecting mechanism makes its study very interesting and challenging. With the main research focus on security issues in the virtual environment, handling of Cross-VM BPA attack is the core of the present research work. The applicability of four BPA attack launching methods has been assessed on different types of VM configurations. Simulation of two important types of BPA (Branch Prediction Analysis) attacks; DTA (Direct Timing Attack) and TDA (Trace-Driven Attack) was also done on the most common VM configuration. With an in-depth study of attack launching methods and their behavior analysis, a four-eyed model Chaturdrashta is proposed. **Chaturdrashta** is comprised of two solutions: *Trilochan* to detect Cross-VM Direct Timing Attack (DTA) and *Trinetra* to detect Cross-VM Trace-Driven Attack (TDA). Solutions can successfully detect the attack by the time when just a few bits are predicted. The processing overhead of the proposed approach is hardly 1%. Additionally, *Trilochan* and *Trinetra* in their original form were also found capable of detecting the presence of the BPA attack launched with the Asynchronous and Synchronous BTB Eviction methods. A testbed comprising of various types of genuine processes was simulated to check the efficiency of solutions. With high accuracy in attack detection, the solutions do not have any false positives. The proposed solutions neither depend on any cryptographic algorithm nor manipulate any architectural components.

**Chaturdrashta** is a host-based solution, where one of the components is embedded to the Kernel. The other three components are implemented as Linux services. Such an implementation requires system reboot to bring their manipulations into effect. In turn, it reduces the scope of **Chaturdrashta** of getting exploited.

**List of Publication(s)**

1. **Published**

2. **Under Review**