**Traditional Honeypots and their history:**

A honeypot is a decoy computer system that often emulates exploited web vulnerabilities and waits for attackers to exploit that weakness. They are an additional level or system functioning alongside traditional Internet security systems. Honeypots purposely lure attackers to retain information on their attacking techniques and patterns used during malicious activities. They also uncover unconventional or new hacking methods. Traditionally, interaction was limited to the web browser.

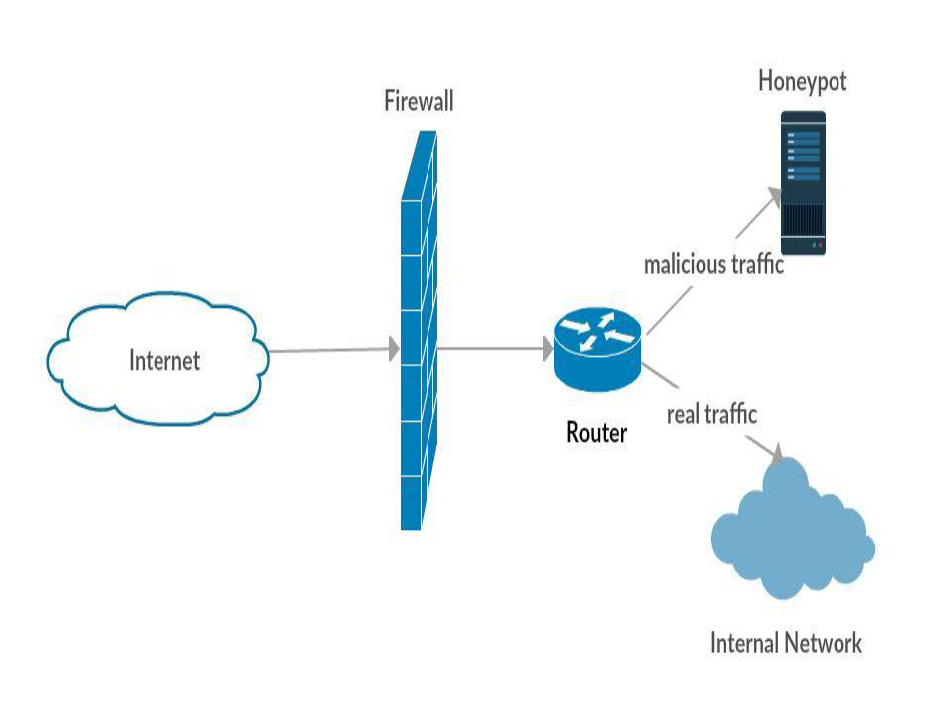


Fig 1. Basic honeypot design

**Honeynets:**

A network of honeypots. A key component is the honeywall that has capabilities such as data capture, control, analysis and alerting.

**Honey**-**farms**:

A set of honeynets.

**Honeypots are characterised by:**

Interaction level, deployment modes and deployment categories.

**The recent problem:**

Preventing distributed denial of service(DDoS) attacks where the intruder attacks from multiple sources. The attacker compromises and takes over the local system(masters), and manipulates close-by systems(DDoS agents or slaves) to launch multiple attacks to the target server(Hrishikesh Arun Deshpande, 2015).

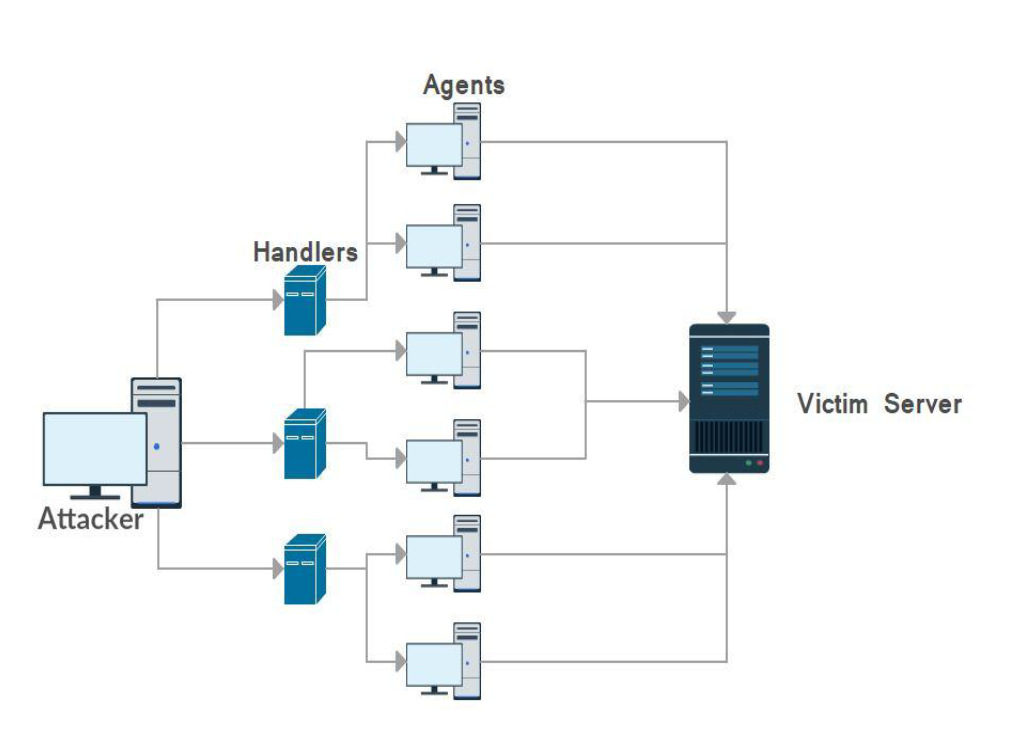


Fig 2. Botmaster and DDoS agents attacking server

**Solution**:

Preventing DDoS attacks by creating a mesh of honey VM's and honey daemons(honey-d's).

Countering flooding attacks(overflow of thousands of requests), the virtual honeypot continuously monitors potential attacks and redirects traffic (modifies router's routing information) to the honey VM network upon detection. Finally, the firewall will block the malicious traffic.

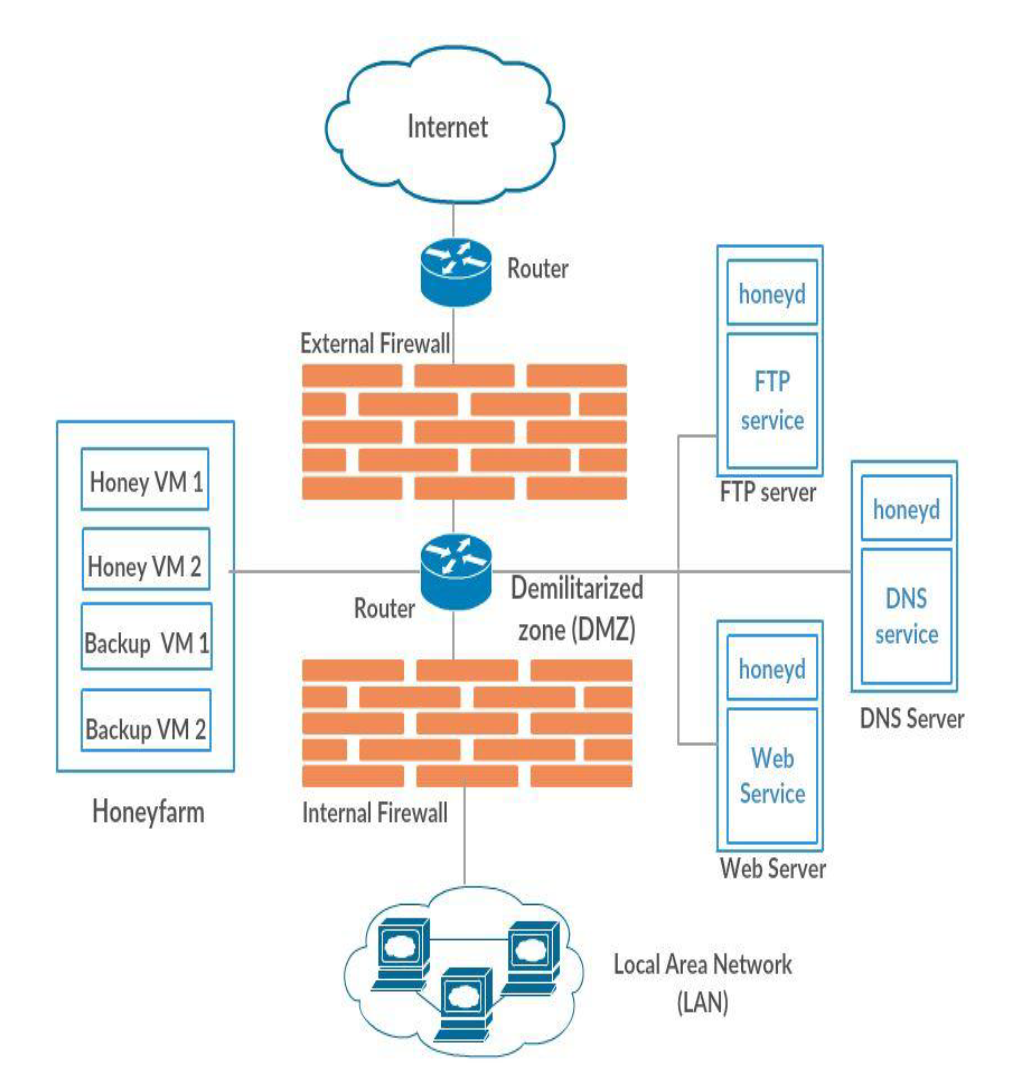
Honey VMs are customised to mimic the real servers as a countermeasure to crashing attacks(data corruption, etc). Each honeypot is also backed up and activated when the original VM is compromised. Machine learning algorithms form a baseline behaviour from the attacker's requests and adopts a challenge-response model if the VM decides to investigate 

Fig 1. The HoneyMesh Architecture

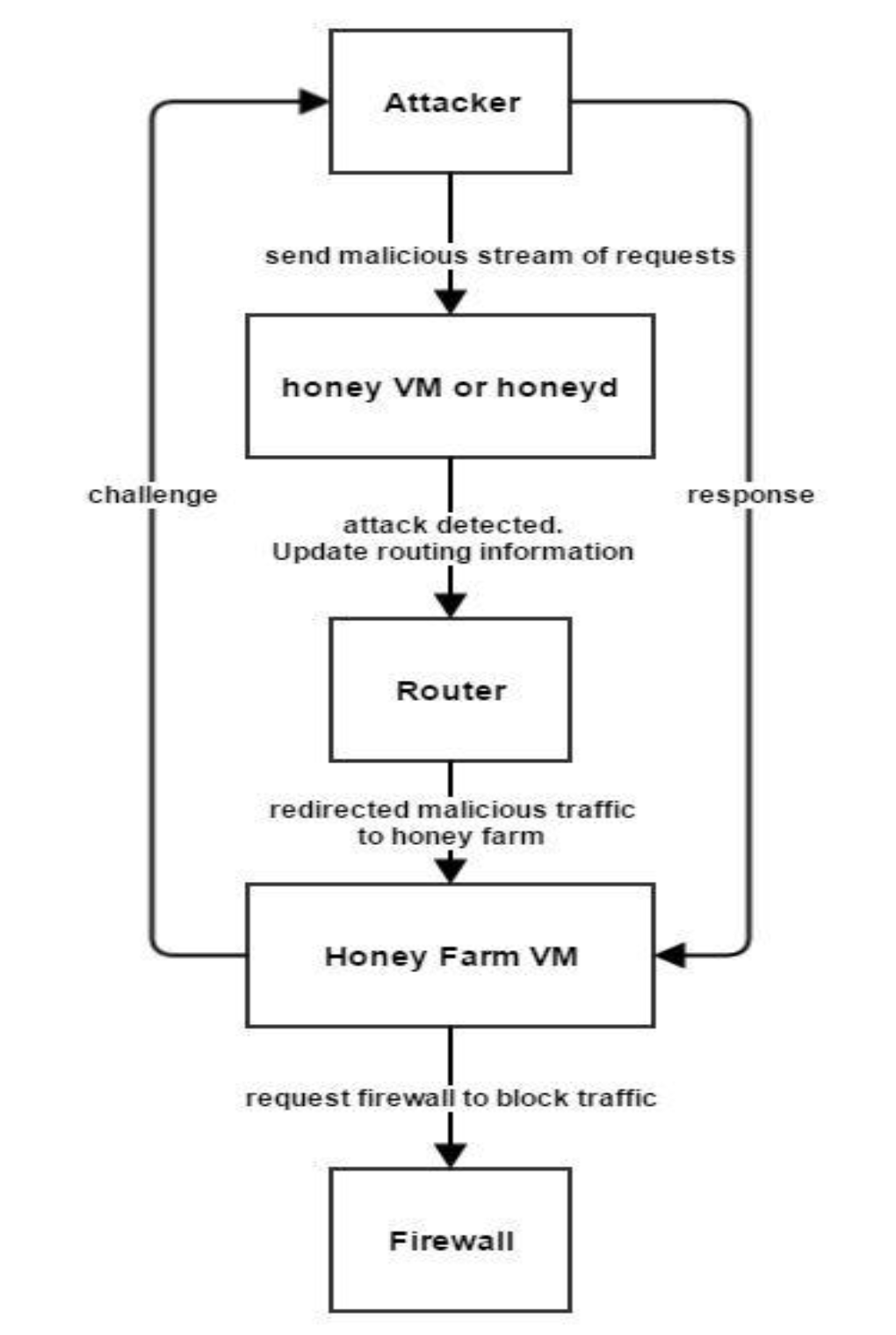


Fig 2. Challenge-Response Model

**Advantages**:

- Virtual honeypots are scalable, economical and requires low maintenance.

- A back-up copy of the honey VMs ensures the intrusion detection and deflection continually functions despite the original honeypot being compromised by a DDoS.

- Restoring a compromised honeypot is cheap and requires minimum downtime.

- Even if the DDoS attack is missed by the honeypot farm, it may be caught by the honey-d that runs on individual production servers as an extra authentication and intrusion detection system (Hrishikesh Arun Deshpande, 2015).

- Redirects malicious traffic to protect the legitimate network.

**Disadvantages**:

- Although the organisation's local area network is fully protected, there needs a mechanism to protect the organisation's routers from flooding attacks (Hrishikesh Arun Deshpande, 2015).

- Hosted on a single server to reduce costs.

c- Because the VMs share the same physical resources, they may interfere with each other in complex and unpredictable ways.

- The risk that the honey VM is taken over and controlled by the botmaster.

- Only tracks and captures activity where an attacker has probed. Otherwise, it is idle.

**Related** **works**:

Fu proposed an approach to camouflage the VM honeypots against advanced bots that scan for VMs. By emulating the honeypot's link latency to mimic the real network 's latency, bots cannot differentiate them. By changing the Linux Kernel parameter HZ(microseconds) it improves the OS timing accuracy (Fu, 2006). Future works will let the emulation run on nanosecond resolution.

Tang's double honeypot strategy uses intelligent honeypots that divert malicious traffic to another honeypot and can be extended to n-tier honeypots with diverse propagation algorithms(Wang, Wu, Cunningham, & Zou, 2010).

According to Rechavi specific attack patterns are location-dependent (Rechavi & Berenblum, 2015).

A new roaming honeypot has some predictive ability to DDoS attacks. It's key concept is blending the malicious flows with the physical migration of legal flows (Zhang, 2010). It finds the attacks early and roams ahead so the location of the host is unpredictable to the attacker. Another new technique uses ant-based detection to deny DDoS attacks using roaming virtual honeypots(Selvaraj, Marwala, & Madhav Kuthadi, 2016).

**Analysis**:

In the event the single server is compromised, the honey-mesh will fail. There is no simulation done to verify the effectiveness of this model in terms of detecting a hijacked honey VM and surviving the single server failure. Also, there is a lack of mechanisms to strengthen the VM's ability to hide itself as a VM.

In most cases of a DDoS attack, the IP addresses are forged to hide the real identity of the attacker. Improving the binary code analysis will enhance the honeypot's ability to find the botmaster's real IP(Joshi & Sardana, 2011).

**Comparisons**:

If this honey-mesh model was run on a multi-server cloud environment like the one suggested by Zhao, the survivability of the application is much more robust (Zhao, 2014).

The honeypots follows a very reactive process where they have to wait to be attacked, unlike the model that proposes to detect botnets before they attack with a 40% accuracy(Moon, Kim, Hur, & Kim, 2012).

**Conclusion:**

The shortcoming of the single server, multi-level virtual honey-mesh can be fixed by running it on multiple servers in the cloud. Honeypots are futile if they are recognised as VMs. Hence, camouflaging its VM identity by altering its link latency is essential for deception to work. Some honey VMs can be configured to leverage off the double-honeypot design for additional deception. Also, some honey VMs can follow the roaming defence strategy(Zhang, 2010).

Overall, there needs a strategy for honeypots to deal with different attacks so this model can counter against other DDoS attacks. The analysis of attacks can be improved by considering the attacker's topological location and additionally, the strategy to deal with it can be improved by classification algorithms that tailor the response to the attack type(Costarella & Chung, 2014).

References

Costarella, C. & Chung, S. (2014). Hardening Honeynets against Honeypot-Aware Botnet Attacks. Retrieved from https://www.tacoma.uw.edu/sites/default/files/sections/InstituteTechnology/C\_Costarella.pdf

Fu, X. (2006). On Recognizing Virtual Honeypots and Countermeasures.

Hrishikesh Arun Deshpande,. (2015). HoneyMesh: Preventing Distributed Denial of Service Attacks using Virtualized Honeypots. *IJERT*, *V4*(08). http://dx.doi.org/10.17577/ijertv4is080325

Joshi, R. & Sardana, A. (2011). *Honeypot*. Enfield, N.H.: Science Publishers.

Moon, Y., Kim, E., Hur, S., & Kim, H. (2012). Detection of botnets before activation: an enhanced honeypot system for intentional infection and behavioral observation of malware. *Security And Communication Networks*, *5*(10), 1094-1101. http://dx.doi.org/10.1002/sec.431

Rechavi, A. & Berenblum, T. (2015). Hackers Topology Matter Geography. *IEEE􀀬􀀨􀀨􀀨􀀒􀀤􀀦􀀰􀀃􀀬􀁑􀁗􀁈􀁕􀁑􀁄􀁗􀁌􀁒􀁑􀁄􀁏􀀃􀀦􀁒􀁑􀁉􀁈􀁕􀁈􀁑􀁆􀁈􀀃􀁒􀁑􀀃􀀤􀁇􀁙􀁄􀁑􀁆􀁈􀁖􀀃􀁌􀁑􀀃􀀶􀁒􀁆􀁌􀁄􀁏􀀃􀀱􀁈􀁗􀁚􀁒􀁕􀁎􀁖􀀃􀀤􀁑􀁄􀁏􀁜􀁖􀁌􀁖􀀃􀁄􀁑􀁇􀀃􀀰􀁌􀁑􀁌􀁑􀁊􀀬􀀨􀀨􀀨􀀒􀀤􀀦􀀰􀀃􀀬􀁑􀁗􀁈􀁕􀁑􀁄􀁗􀁌􀁒􀁑􀁄􀁏􀀃􀀦􀁒􀁑􀁉􀁈􀁕􀁈􀁑􀁆􀁈􀀃􀁒􀁑􀀃􀀤􀁇􀁙􀁄􀁑􀁆􀁈􀁖􀀃􀁌􀁑􀀃􀀶􀁒􀁆􀁌􀁄􀁏􀀃􀀱􀁈􀁗􀁚􀁒􀁕􀁎􀁖􀀃􀀤􀁑􀁄􀁏􀁜􀁖􀁌􀁖􀀃􀁄􀁑􀁇􀀃􀀰􀁌􀁑􀁌􀁑􀁊*, *􀀕􀀓􀀔􀀘􀀃􀀬􀀨􀀨􀀨􀀒􀀤􀀦􀀰􀀃􀀬􀁑􀁗􀁈􀁕􀁑􀁄􀁗􀁌􀁒􀁑􀁄􀁏􀀃􀀦􀁒􀁑􀁉􀁈􀁕􀁈􀁑􀁆􀁈􀀃􀁒􀁑􀀃􀀤􀁇􀁙􀁄􀁑􀁆􀁈􀁖􀀃􀁌􀁑􀀃􀀶􀁒􀁆􀁌􀁄􀁏􀀃􀀱􀁈􀁗􀁚􀁒􀁕􀁎􀁖􀀃􀀤􀁑􀁄􀁏􀁜􀁖􀁌􀁖􀀃􀁄􀁑􀁇􀀃􀀰􀁌􀁑􀁌􀁑􀁊􀀕􀀓􀀔􀀘􀀃􀀬􀀨􀀨􀀨􀀒􀀤􀀦􀀰􀀃􀀬􀁑􀁗􀁈􀁕􀁑􀁄􀁗􀁌􀁒􀁑􀁄􀁏􀀃􀀦􀁒􀁑􀁉􀁈􀁕􀁈􀁑􀁆􀁈􀀃􀁒􀁑􀀃􀀤􀁇􀁙􀁄􀁑􀁆􀁈􀁖􀀃􀁌􀁑􀀃􀀶􀁒􀁆􀁌􀁄􀁏􀀃􀀱􀁈􀁗􀁚􀁒􀁕􀁎􀁖􀀃􀀤􀁑􀁄􀁏􀁜􀁖􀁌􀁖􀀃􀁄􀁑􀁇􀀃􀀰􀁌􀁑􀁌􀁑􀁊􀀕􀀓􀀔􀀘􀀃􀀬􀀨􀀨􀀨􀀒􀀤􀀦􀀰􀀃􀀬􀁑􀁗􀁈􀁕􀁑􀁄􀁗􀁌􀁒􀁑􀁄􀁏􀀃􀀦􀁒􀁑􀁉􀁈􀁕􀁈􀁑􀁆􀁈􀀃􀁒􀁑􀀃􀀤􀁇􀁙􀁄􀁑􀁆􀁈􀁖􀀃􀁌􀁑􀀃􀀶􀁒􀁆􀁌􀁄􀁏􀀃􀀱􀁈􀁗􀁚􀁒􀁕􀁎􀁖􀀃􀀤􀁑􀁄􀁏􀁜􀁖􀁌􀁖􀀃􀁄􀁑􀁇􀀃􀀰􀁌􀁑􀁌􀁑􀁊􀀕􀀓􀀔􀀘􀀃􀀬􀀨􀀨􀀨􀀒􀀤􀀦􀀰􀀃􀀬􀁑􀁗􀁈􀁕􀁑􀁄􀁗􀁌􀁒􀁑􀁄􀁏􀀃􀀦􀁒􀁑􀁉􀁈􀁕􀁈􀁑􀁆􀁈􀀃􀁒􀁑􀀃􀀤􀁇􀁙􀁄􀁑􀁆􀁈􀁖􀀃􀁌􀁑􀀃􀀶􀁒􀁆􀁌􀁄􀁏􀀃􀀱􀁈􀁗􀁚􀁒􀁕􀁎􀁖􀀃􀀤􀁑􀁄􀁏􀁜􀁖􀁌􀁖􀀃􀁄􀁑􀁇􀀃􀀰􀁌􀁑􀁌􀁑􀁊􀀕􀀓􀀔􀀘􀀃􀀬􀀨􀀨􀀨􀀒􀀤􀀦􀀰􀀃􀀬􀁑􀁗􀁈􀁕􀁑􀁄􀁗􀁌􀁒􀁑􀁄􀁏􀀃􀀦􀁒􀁑􀁉􀁈􀁕􀁈􀁑􀁆􀁈􀀃􀁒􀁑􀀃􀀤􀁇􀁙􀁄􀁑􀁆􀁈􀁖􀀃􀁌􀁑􀀃􀀶􀁒􀁆􀁌􀁄􀁏􀀃􀀱􀁈􀁗􀁚􀁒􀁕􀁎􀁖􀀃􀀤􀁑􀁄􀁏􀁜􀁖􀁌􀁖􀀃􀁄􀁑􀁇􀀃􀀰􀁌􀁑􀁌􀁑􀁊􀀬􀀨􀀨􀀨􀀒􀀤􀀦􀀰􀀃􀀬􀁑􀁗􀁈􀁕􀁑􀁄􀁗􀁌􀁒􀁑􀁄􀁏􀀃􀀦􀁒􀁑􀁉􀁈􀁕􀁈􀁑􀁆􀁈􀀃􀁒􀁑􀀃􀀤􀁇􀁙􀁄􀁑􀁆􀁈􀁖􀀃􀁌􀁑􀀃􀀶􀁒􀁆􀁌􀁄􀁏􀀃􀀱􀁈􀁗􀁚􀁒􀁕􀁎􀁖􀀃􀀤􀁑􀁄􀁏􀁜􀁖􀁌􀁖􀀃􀁄􀁑􀁇􀀃􀀰􀁌􀁑􀁌􀁑􀁊*.

Selvaraj, R., Marwala, T., & Madhav Kuthadi, V. (2016). Ant-based distributed denial of service detection technique using roaming virtual honeypots. *IET Communications*. http://dx.doi.org/10.1049/iet-com.2015.0497

Wang, P., Wu, L., Cunningham, R., & Zou, C. (2010). Honeypot detection in advanced botnet attacks. *IJICS*, *4*(1), 30. http://dx.doi.org/10.1504/ijics.2010.031858

Zhang, X. (2010). Design and implementation of a new roaming honeypots defense strategy.

Zhao, M. (2014). Multi-level VM Replication based Survivability for Mission-critical Cloud Computing.