

Detecting Targeted Attacks Using Shadow Honey Pots

Presented By Archana Viswanath



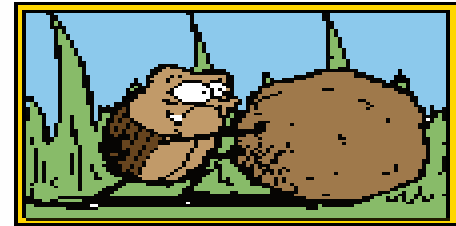
Honeypots - Where did the name come from???

- Honeypot is often understood to refer to the English children's character Winnie-the-Pooh, a stuffed bear who was lured into various predicaments by his desire for pots of honey.



Honeypots – Network Parlance

- An Internet-attached server.
- Aim - acts as a decoy to lure potential hackers
- Actions - Studies their activities and monitors how they are able to break into a system.
- The intruder will have no idea that she/he is being tricked and monitored.



Honeypots – Purpose

- Learn the weaknesses in the system.
- To catch and stop the hacker.
- Create more secure systems that are potentially invulnerable to future hackers.





Honey Pots...

- They are a resource that has no authorized activity and no production value.
- This means that any interaction with a honeypot is most likely malicious or unauthorized.
- Any connections sent to the honeypot are most likely a probe, scan or attack.

Honeypots Effective Against....

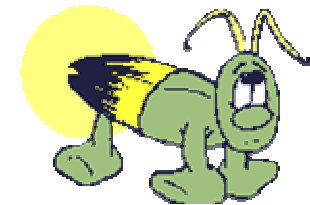
- Effective against Scanning worms.
- Ineffective against – Topological , hit-list worms.....(Why???)
- Typically used only for server-type applications.

Anomaly Detection Systems

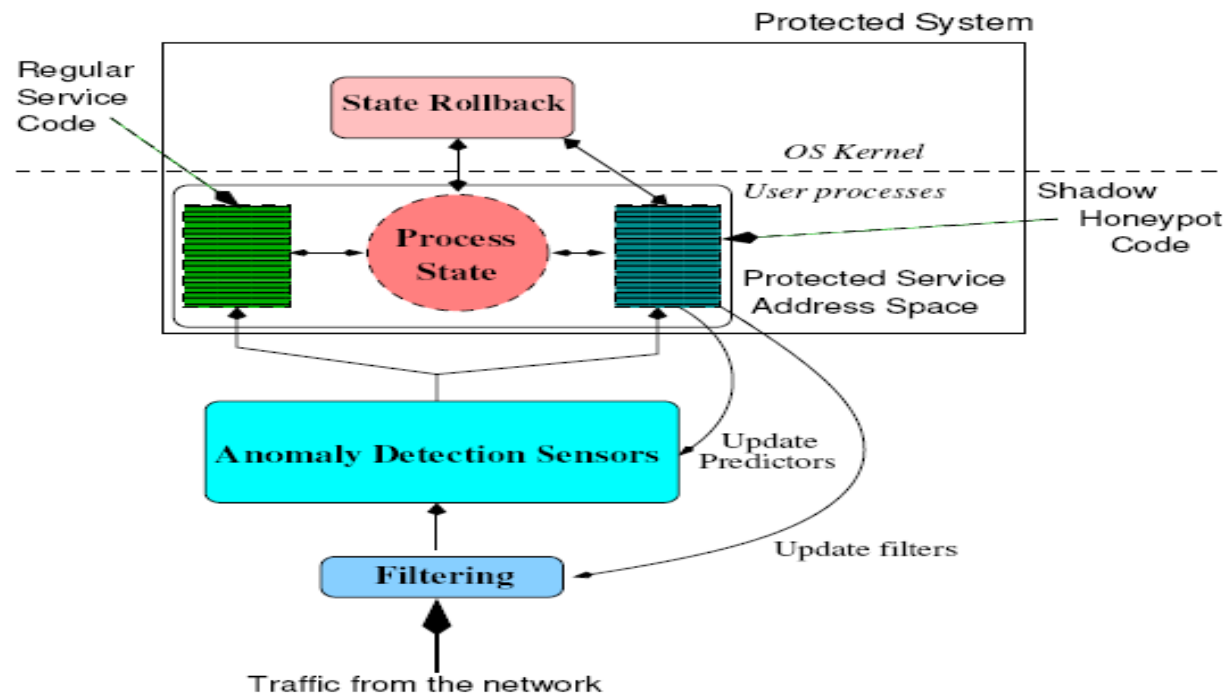
- Detects anything anomalous from normal behavior of the system.
- **Advantage** – Possibility of detecting and responding to previously unknown attacks.
- **Disadvantage** –
 - (i). Tune the system to detect more potential attacks (Low FN, High FP)
 - (ii). Tune the system to be more insensitive to attacks (High FN, Low FP)

Shadow Honey pots

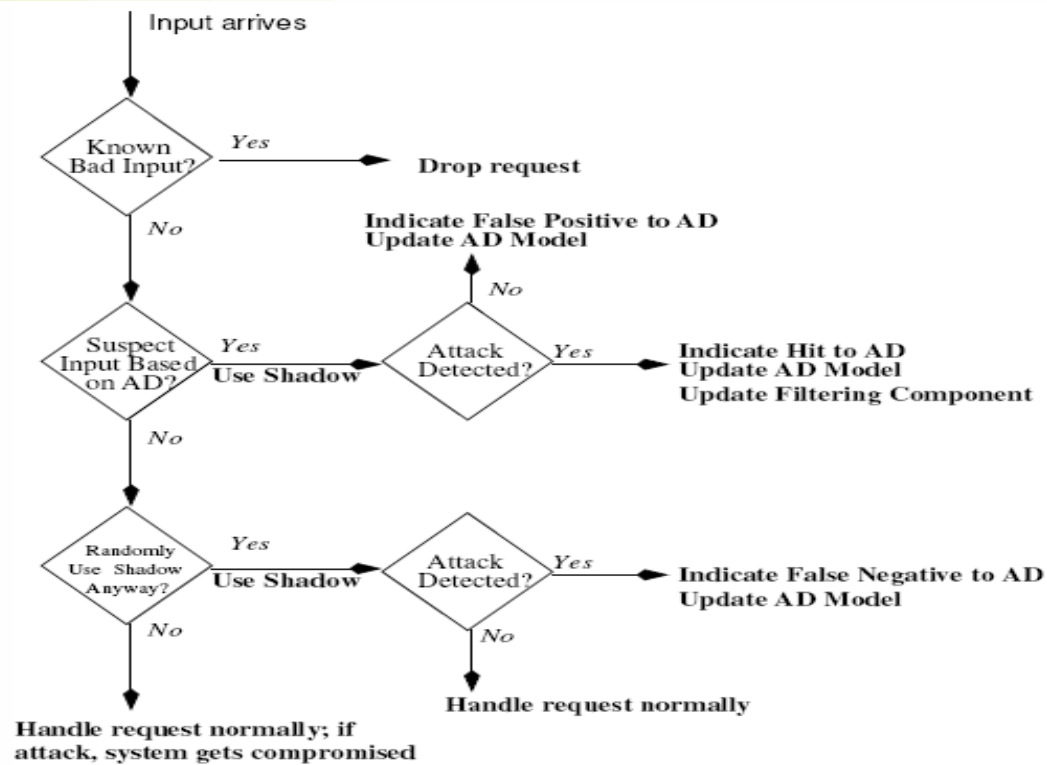
- Combines the features of Honey pots and Anomaly Detection Systems.
- ADS monitors the incoming traffic.
- Anomalous traffic is further processed by the shadow honeypot.
- Shadow is an instance of the protected software.



Shadow Honeypot Architecture



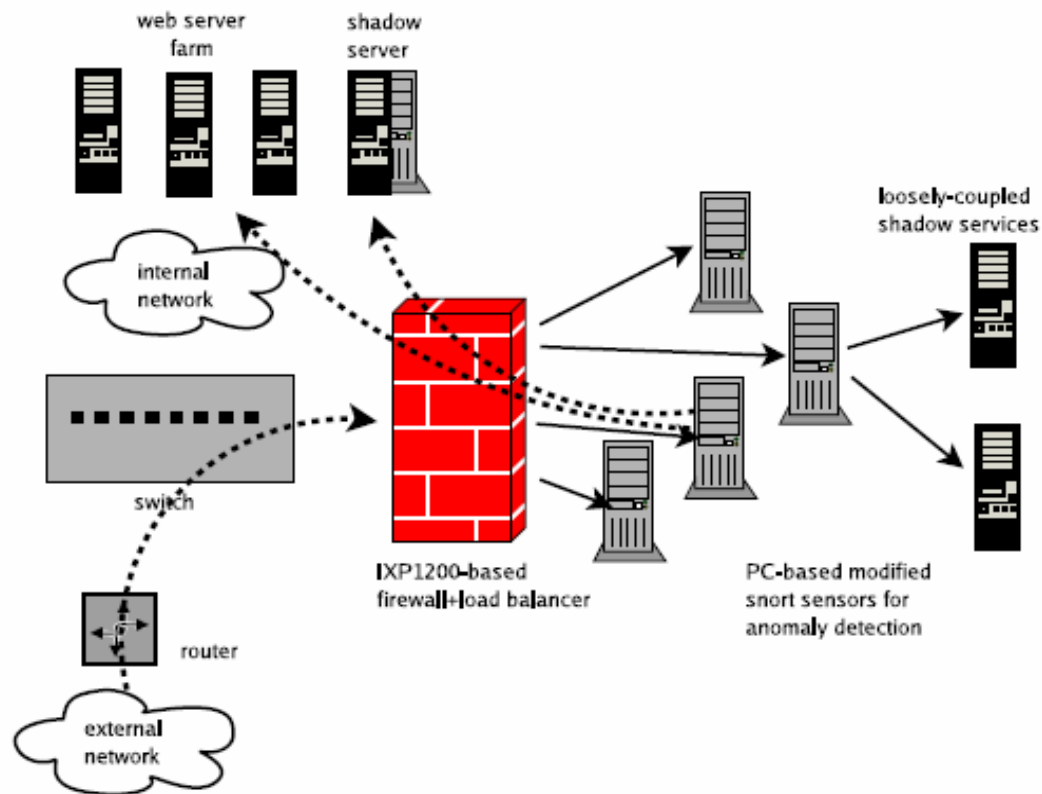
System Workflow



Shadow HoneyPot Prototype

- Network Processor – custom load balancer & filter
- Snort Sensors – connected to the network processor
- Shadow HoneyPot – connected to the sensors.

Shadow Honeypot Prototype





Implementation

Filtering and Anomaly Detection

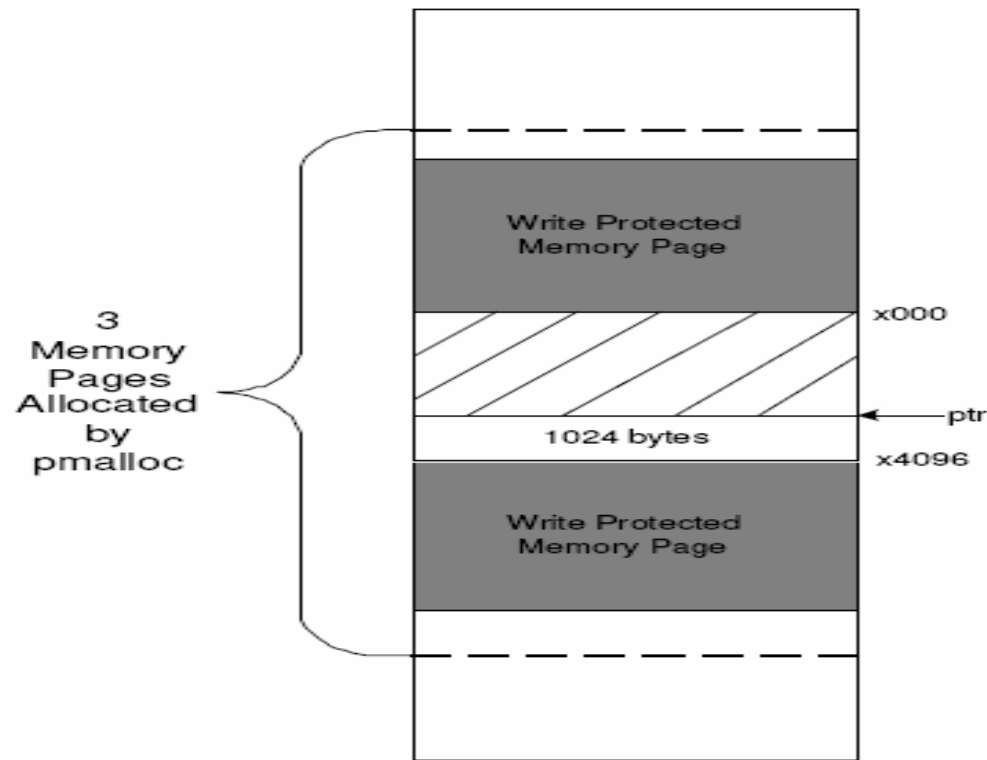
- Two Anomaly Detection Systems – Payload Sifting & Abstract Payload Execution
- Payload Sifting – fingerprinting- High False Positives
- Abstract Payload Execution – Buffer Overflow detection – searches for sufficiently long sequences of valid instructions.

Implementation...

Shadow Honeypot Creation

- Use `pmalloc()` instead of `malloc()` for heap allocation.
- `pmalloc()` allocates two additional zero filled, write-protected pages that bracket the requested buffer.
- Buffer overflow – will cause the process to receive a Segmentation Violation signal.
- This is caught by the signal handler which notifies the OS to abort all changes.
- If no violation then changes are persisted.

Implementation... Memory Allocation



Limitations



- Effectiveness of the rollback mechanisms for the transactions depends on their proper placement for committing state changes and latency of the detector.
- Loosely coupled shadows – weak against attacks that depend on the system state.
- Not explored – feedback from shadow honeypots to tune the anomaly detection components.

Take Away

- Its obvious that fine tuning will get you a step closer to efficient detection.

