Frequency Minimal Moving Target Defense Using Software-Defined Networking

Allen Stage and Vladimir Georgiev

Mentors: Prof. Prasad Calyam, Dr. Saptarshi Debroy, Minh Nguyen
Cloud systems can be vulnerable to a variety of threats:

- **Information Leakage**
  - Cause: Eavesdropping, Traffic Interception
  - Effect: Loss of confidentiality
- **Integration Violation**
  - Cause: Intercept/Alter, Repudiation
  - Effect: Loss of integrity
- **Denial of Service**
  - Cause: Trojan Horse, Resource Exhaustion
  - Effect: Loss of Availability
- **Illegitimate Use**
  - Cause: Spoofing, theft
  - Effect: Improper Authentication
## Top Cloud Computing Threats in 2013

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Data Breaches</td>
<td>2. Data Loss</td>
<td>3. Account Hijacking</td>
<td>4. Insecure APIs</td>
</tr>
</tbody>
</table>
Denial of Service and Loss of Availability

- **LOA**
  - Loss of Availability

- **DOS**
  - Denial of service
  - Simple to execute
  - Attacker bombards a server with requests, and render it completely useless for other users
Traditional Security Strategies

- Cryptographic strategies against LOA
  - Proof-of-Retrievability (POR) [1]
  - Proof of Data Possession (PDP) [2][3][4][5]

- Strategies against DDOS
  - Router filtering [6][7]
  - Instrument prevention system (IPS) [8][9][10]
Moving target defense (MTD) is the concept of controlling change across multiple system dimensions by moving around VMs hosting services.

MTD focuses on enabling safe operation in a compromised environment, rather than trying to create a perfectly secure environment.
Why MTD for cloud security?

- Improves resilience through randomization, helps achieve cyber defense goals
  - Increased cost to attacker
  - Decreased knowledge of whether or not attack was successful
    - Increased chance of attacker detection
- Contains proactive (preventive) and reactive (cure) defense to prevent attacks
- Intelligent proactive and reactive strategies can help tackle LOA attacks!
# Related Work on MTD for Cloud Security

<table>
<thead>
<tr>
<th>Related work</th>
<th>Strengths</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>[11]</td>
<td>Shuffling static IP addresses of attacked VMs</td>
<td>Only reactive strategy</td>
</tr>
<tr>
<td>[12]</td>
<td>Moving proxies to application servers to thwart attack</td>
<td>Attacker can realize defense strategy in place</td>
</tr>
<tr>
<td>[13]</td>
<td>Proactive VM migration using attack traffic signature</td>
<td>Too reliant on accuracy of signature detection</td>
</tr>
<tr>
<td>[14]</td>
<td>Multiple VMs host same service, users are only redirected</td>
<td>Not really MTD, limited cost-effectiveness</td>
</tr>
<tr>
<td>[15]</td>
<td>Attackers are marginalized within a small pool of decoy VMs</td>
<td>Does not guarantee 100% regular user redirection</td>
</tr>
</tbody>
</table>
Our Research Goals

- Both proactive and reactive movement strategies
- Optimal cost effective migration strategy
- Trade-off between cost of movement and difficulty for attacker to guess
- Attacker should not know about the movement and keep targeting the old VM
Our FM-MTD Novelty

- Our SDN-enabled migration scheme performs dynamic VM migration
  - Whereas, existing works resorts to IP address shuffling
- Our scheme is both proactive and reactive
  - Whereas, existing works are purely reactive
- Our scheme is adaptive to attack probability and attack budget
  - Whereas, existing use migration frequency that is static
- Our scheme considers heterogeneous VM pool
  - Whereas, existing works assume a homogeneous VM pool
MTD System Model

- Malicious and regular users accessing the services hosted by a target VM
- Authentication server to authorize users
- Open flow controller to detect attack, run MTD logic, and perform migration
  - Only regular users redirected to new VM

![Diagram showing the system model with authentication server, open flow controller, and target VMs.]

**LEGENDS**
- Control path
- Attack path
- Regular path
- Data migration path
Three Big Questions

- Where to move?
  - Finding the optimal candidate VM to migrate
  - Identifying the most pertinent VM selection factors
  - Periodic/on-demand information collection
  - Finding the factors’ relative importance to create migration logic
- When to move?
  - Finding the optimal frequency of movement
  - Not too frequent as migration incurs cost, and not too seldom as increases probability of getting attacked
- How to move?
  - Mostly pertains to implementation issues
  - Proactive/reactive migration execution
  - Runtime migration or file copy
  - Redirection of regular users
Optimal Migration Frequency

- Ideal frequency should be such that it is not too frequent, while not being too infrequent
  - Too frequent
    - can waste valuable network resources
  - Too infrequent
    - makes VM more vulnerable

Movement costs resources, just like moving houses costs time and money
The optimization can be formulated as

\[ \text{maximize}(T_m) \]

\[ T_m \leq \text{cyberattack inter-arrival time} \]

Assume the random variable representing the attack inter-arrival time be \( z \) which is the sum of two independent and random variables for Attacked and Idle periods \( x \) and \( y \), respectively.

The distribution of attack interval \( z \) is obtained by:

\[
f_Z(z) = f_X(x) * f_Y(y) = \int_{-\infty}^{+\infty} f_X(z-y) f_Y(y) dy
\]

\[
= \left\{ \begin{array}{ll}
\frac{\lambda_a \mu_i [e^{-\lambda_a z} - e^{-\mu_i z}]}{\lambda_a - \mu_i} & \forall \lambda_a \neq \mu_i \\
\lambda_a^2 z e^{-\lambda_a z} & \text{otherwise}
\end{array} \right.
\]
To quantify optimal $T_m$, calculate probability of VM getting attacked before migration

$$\text{Prob}\{\text{VM getting attacked before migration}\}$$

$$= \text{Prob}\{z \leq T_m\} \quad \text{(VM attack being memoryless)}$$

$$= \int_{-\infty}^{T_m} f_Z(z) \, dz$$

$$= \left\{ \begin{array}{ll}
\int_0^{T_m} \frac{\lambda_a \mu_i [e^{-\lambda_a z} - e^{-\mu_i z}]}{\lambda_a - \mu_i} \, dz & \forall \lambda_a \neq \mu_i \\
\int_0^{T_m} \lambda_a^2 z e^{-\lambda_a z} \, dz & \text{otherwise}
\end{array} \right.$$  

$$= \left\{ \begin{array}{ll}
\mu_i (e^{-\lambda_a T_m} - 1) + \lambda_a (1 - e^{-\mu_i T_m}) \over \lambda_a - \mu_i & \forall \lambda_a \neq \mu_i \\
1 - e^{-\lambda_a T_m} (\lambda_a T_m + 1) & \text{otherwise}
\end{array} \right.$$  

$\text{Lambda}_a$ is representative of attack period

$\text{Mu}_i$ is representative of idle period
Migration interval ($T_m$) optimization for different attack budgets

*A visual representation of the equation slides, with many movement frequencies in a graph*
VM selection factors:
- Capacity: New VM should have enough resources (compute/storage)
- Bandwidth: New VM should not be too far to cause extended service interruptions
- Reputation: New VM should not be prone to attack or have prior history of getting attacked

Selection criteria

$$\maximize (S_p^v)$$

where

$$S_p^v = w_c \times C_p + w_b \times B_p^v + w_r \times R_p^j$$
We argue that the previous history of a VM in terms of instances of cyber attacks is a critical factor in deciding the suitability for selection:

- Instances of successful attacks (alpha)
- Instances of unsuccessful attacks (beta)
- Instances of attack-free status (gamma)

Cumulative fair reputation model

\[ R_p^j = 1 - \frac{\alpha_p^j + \frac{\beta_p^j}{\beta_p^j + \gamma_p^j}}{\alpha_p^j + \beta_p^j + \gamma_p^j} \quad \forall \ p \in V \]
Performance Evaluation

- Target Application – Just-in-time news feeds
- Using a software-defined networking controller we developed
  - Contains python and shell scripts that we have written to execute the movement modules
- Scripts will move our application to a new VM
Setup on testbed consists of the following components

- One target VM at Illinois rack hosting the target application
- Four non-malicious clients at four different locations
- Two attackers simulating regular client behavior
- Up to 30 candidate VM’s at different locations simulating varied scenarios
- Controller with software components of control module
Performance Evaluation

Different color groups represent different aggregates (locations)
Cyber attack Impact

Impact of cyber attack on requests from client4

Notice the trend? (Hint: the axis matter)

1 attacker

2 attackers
Impact of Location Selection

- Process of selecting ideal frequency minimal candidate VM over static homogenous
- Response time for client 4 with a less than ideal VM can lead to service quality improvement, compared to attack, but quite less when compared to ideal, in this case up to a factor of ~4
Installed Kentucky PKS2 with similar features as our ideal candidate, the exception being the achievable throughput

- Varying the size of the application
- Increased transfer times affects the service interruption time in the case of an attack
Illinois is targeted, while hosting
UCLA is targeted, but not hosting
Rutgers is not targeted
This time proactive is performed, varying the probability of the attack by varying attack budget.

Optimal migration frequency performs better, up to 50% at lower ends.

Success rate sharply decreases with growing number of VM’s, as guessing out of 30 versus 5 becomes more difficult.

---

**Graphs:**

1/10 budget ratio

1/100 budget ratio
Conclusion

- Proactive movement using our ‘when to move’ module is successful in preventing a greater number of attacks

- Reactive movement using our ‘where to move’ module results in a better response time
Further thoughts and future considerations

- Larger amounts of VM’s created larger run times in the modules, as would be expected.
- A thought on this would be that with a larger number of VM’s the attack probability becomes extremely low anyway, as determined by the frequency optimization.
- Another thought on this is controller type, as discussed in the next slide.
We started on DeterLab then switched to GENI
  - Overall, this turned out to be a good thing! But did come at a cost for only having 10 weeks

Time-management
  - An example is “wasted” time on irrelevant problems (such as with DeterLab node login)
    - These things improved drastically with experience!

Experiment with controllers other than POX

It is a learning process!
What we learned and takeaway

- LaTex, and other ins and outs of research paper fundamentals
- Presentation giving on a weekly basis, as well as listening skills involved in them
- Many different areas from just our own project!
  - Software-Defined Networking fundamentals
  - Moving Target Defense Fundamentals
  - An in-depth look at different topologies and test beds for networking
    - GENI, DeterLab
- How to read and appreciate the contents of research papers (3 pass method, etc.)
- Teamwork!
- How to make a poster, and in depth use of Powerpoint

Most important of all, a great appreciation for research and all the hard work that goes into producing it
A special thanks to all the mentors and research directors of every project for their continued help and guidance for us undergraduates.