Russ McRee
Bryan Casper

INCIDENT RESPONSE IN VIRTUAL ENVIRONMENTS: CHALLENGES IN THE CLOUD

About us

- We’re part of the security incident response team for Microsoft Online Services Security & Compliance
- We ask more questions than provide answers
- This presentation is meant to evoke discussion and likely provide more takeaways for you
- Incident response in the cloud is relatively new
  - Trial and error, experience gained on the fly

Agenda

- Definition
- Services
- Motives
- Incidents
- Enhancements
- Assumptions
- IR
- Recommendations

Cloud definition

- How do we define “the cloud”
  - highly redundant
  - resources on demand
  - scalable
  - operations managed by someone else on your behalf
  - rapid deployment of code and VMs
- How do cloud services vary?

Services considered

- Amazon EC2
- Google GAE
- Windows Azure

- Clearly there others, only so much time to play

Cloud services – Amazon EC2

- Full OS control (Windows and Linux)
- Can use S3 to backup snapshots
- Network ACL's
  - Whitelist rules only
  - TCP\UDP\ICMP [SRC and DST Ports]
  - Source IP range

Cloud Services – Google (GAE)

- Python or Java apps
- Can manage access via Google Aps Domain
- Dashboard with lots of metrics
- Security-centric features include
  - Permissions
  - Blacklist
    - DoS Protection Service for Python or Java
  - Additional security-specific logging must be developed for the app via the appropriate SDK

Cloud services – Windows Azure

- Supports .NET, PHP, Ruby, Python, or Java
- Application Logging via Trace Listeners
  - ETW, trace, debug
  - To access logs, must write log data to blob storage / table storage
- Monitoring Agent
  - event logs, perf counters, crash dumps, custom logs
- Use Diagnostics API to Configure and Collect
  - Event Logs
  - Performance Counters
  - Trace/Debug information (logging)
  - IIS Logs, Failed Request Logs
  - Crash Dumps or Arbitrary files

Attacker Motives

- Abuse resources
- Fraud
- Attack other resources from the cloud
- Competition attacks
  - Force resource expenditure causing net loss
  - Billing models based on storage, bandwidth, CPU time/count, node count
  - Repudiation

Real incidents

- MSN 3rd party Korea: Gumblar
  - Content Delivery Network
- Twitter component of Bing Maps
  - Social networking component hosted in Azure

MSN 3rd party Korea: Gumblar

- Gumblar steals FTP credentials, modifies JavaScript files
- Korean staff running AV noticed that a Korean web page contained the Gumblar malware.
- Security team notified and engaged
- Having a listing of URL's and identify those that belong to caching services
  - In this case the URL from cache was different than the normal sites URL
- Critical to understand how files are uploaded into the cloud
- Critical to understand how to effectively remove files from the cloud

MSN 3rd party Korea: Gumblar

- Investigation revealed that a 3rd party developer system was compromised by Gumblar
- Infected JavaScript was uploaded to the cloud a month earlier
  - Enhanced Detection critical
  - Failure of site owner to appropriately purge the cloud due to inadequate knowledge on how to perform this activity.

MSN 3rd party Korea: Gumblar

- Lessons learned:
  - Good Logging is critical; understand how to request logs.
  - The file moves through the cache and after a period of time the file is deleted from the cache.
  - Logs of when the file was originally uploaded along with MD5 hash allowed for the team to know when it was uploaded and by what IP address / username.
  - Understanding of which time zone the logs may be in (Most likely GMT format)

Twitter component of Bing Maps

- App deployed to Azure
  - No input request size check for x and y map variables
  - Large values loaded, causing the application to crash

Twitter component of Bing Maps

- Lessons learned:
  - No app logs, actual failure discovered by accident
  - No immediate access to web logs
  - Build logging into app
  - Standard web app sec best practices still apply
  - Beware Agile development without proper SDL, gateway check, etc.

Enhance the apps you deploy

- What APIs are being utilized?
  - To write to storage
  - To allocate more resources
- Is the app code itself secure?
  - SDL?
- Ensure proper app logging
  - How are your logs stored at rest?
  - Are you fully cognitive of where logs are stored and do you have immediate access to them per incident?

Enhance the apps you deploy

- You cannot respond to what you cannot see
  - Apps should provide visibility with end to end monitoring if they are deemed “critical”
  - Baselines
    - What is normal and expected?
    - Can you threat model against deployment and architecture assumptions in order to validate?
    - “Application ACLs will protect my cloud instances and apps from being abused.”
      - Are you sure?

Infrastructure assumptions

- Are your apps/instances appropriately…
  - Routing
  - DNS
    - TMI via lookups?
    - What if cache is poisoned or records are manipulated, how would investigate it if you're not managing DNS?
    - You've given up further control, classic attacks still work i.e. registrar hacks. Are they harder to analyze as a result?
  - Firewall
    - ACL

Data in the cloud

- Should you store sensitive information in the cloud at this time?
  - Are cloud services proven enough yet?
- Recommended that no medium or high business impact data be stored in the cloud
- This gets a bit cloudy when you measure SaaS vs. pure cloud services

Incident response capability

- IR node?
  - Log collection
  - tools
- Need snapshot capability: can it be remounted for investigation as read only, state preserved?
- See *Forensics considerations in next generation cloud environments* - Robert Rounsavall

Incident response changing

- Incident response teams for entities using cloud services must intimately understand architecture and data flow
  - What are the attack vectors?
    - From cloud to your enterprise
    - From your enterprise to the cloud
  - Can you effectively do an “operational threat model”?
- IR team must understand content “upload” and the native application attributes

Incident response changing

- Touch points between legacy infrastructure and cloud
  - Do vulns or exploits in non-cloud resources that have access to cloud become realized
    - Think Gumblar incident
- Vulns in cloud deployed apps vs. classic deployment are still simply vulns
  - A Ruby on Rails 0-day doesn’t care where it lives

Incident response changing

- Do you trust your cloud neighbors?
  - Remember abusing WCF to perform remote port scans?
- Memory analysis?
- Blob storage analysis?
- Will legacy tools run on your cloud nodes?
  - Have you tested, confirmed, and drilled the process?

Cloud services as mitigations

- Caching cloud can help offset DDoS
- Assuming contracts/SLAs are met and cloud service is well managed, service may be better than ISP/colo services
- Outage prevention via failover capabilities may be more nimble
  - Your core datacenter L2/L3 router pair fails, need hot standby to stay online
  - Cloud services assume redundancy that could prevent concerns as above

Recommendations - Technical

- IP filters
  - DoS protection
- flow monitoring
- cloud toolkit
  - scripts and tools relevant to the cause
  - IR node
- cloud developer kits for better deployment understanding

Recommendations - Risk

- Application portability is part of all provider's charter, but also moves the risk around
  - From data center to the cloud
    - From the cloud to a private cloud (back in your data center)
- Data classification defined by business
  - If PII, high impact data is to go in the cloud can you wrap in a hard candy shell around it just like you already do?
  - Are cloud services ready to handle sensitive data?

Recommendations - SLA

- SLA
  - Contract language
    - Is it clearly defined?
  - What can incident responders expect from provider?
    - support
    - response time
    - account reset
  - Evidence and log retention and acquisition
  - Legal considerations if your cloud instances are compromised and utilized maliciously (subject to subpoena)

Recommendations - Development

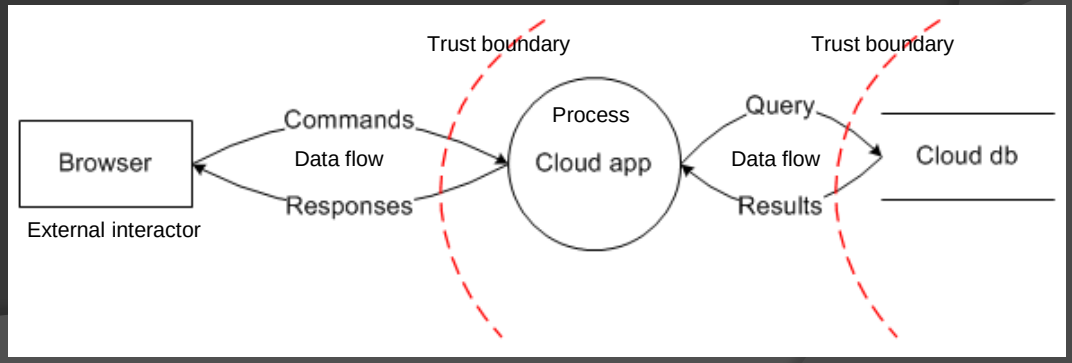
- Agile development and operational best practices don’t always converge
- Developers don’t typically account for operational considerations
- Security Response Plan (SRP)
  - All apps deployed should have an SRP
  - IR needs to be part of the development process
    - Define requirements for logging, tooling, access management, fix deployments, escalation
- Code level threat modeling applies

Recommendations - Operational

- Vulnerability assessment
  - Scanning your cloud presence
- Vulnerability management
  - Patching
  - Updates
  - Fix deployment
  - Standard images
- Who deploys what?
  - Separation of duties
- Operational threat modeling

Recommendations – Threat Model

- The same threat modeling practices developers should utilize for code development can be utilized in an operational capacity
- Infrastructure threat modeling
  - Vision (scope)
  - Model (diagram)
  - Identify Threats
  - Mitigate
  - Validate



In closing

- IR teams should be very clear about operational considerations for resources beyond their control
  - KNOW YOUR CLOUD
- Log, log, log
- Balance risk against business gain
  - If risk exceeds your well-informed comfort, assign risk via threat modeling or assessment

Q & A

- rmcree@microsoft.com
- bcasper@microsoft.com