An information-gathering botnet for private cloud environments

Problem statement

Companies can not easily monitor private cloud environments without changing the underlying architecture of the cloud.

- Many enterprises are moving towards cloud computing
- Cloud computing environments cannot be monitored in the same ways as physical data centres
- Cloud computing solutions are often proprietary
- PROBLEM STATEMENT: Companies can not easily monitor private cloud environments without changing the underlying architecture of the cloud.

In this presentation

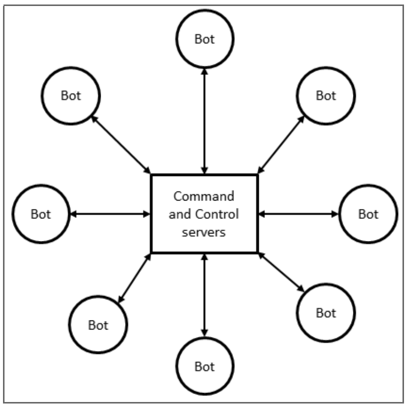
- Overview of botnets
- Overview of cloud computing
- Solution
  - Overview
  - Architecture
  - Program flow
- Evaluation of solution

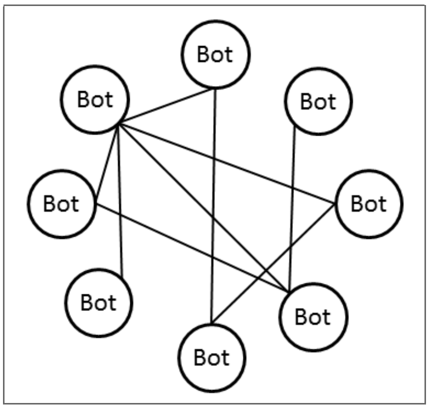
- Overview of botnets
- Overview of cloud computing
- Solution
  - Overview
  - Architecture
  - Program flow
- Evaluation of solution

Botnet

A network of compromised computers that is controlled by an attacker (the botmaster).
Definition of botnet: A network of compromised computers that is controlled by an attacker (the botmaster).
The bots are spread all over the world which makes it nearly impossible to destroy the botnet. Many modern botnets also have the ability to update themselves, allowing them to avoid anti-virus and add new functionality.

Botnet topologies





There are various topologies that is used by botnet authors. The two main ones is centralized, also known as Command-and-Control, and peer to peer. In a centralized botnet, all the bots communicate with a central server. The server is responsible for sending commands to the bots and receiving data from them. In a Peer-to-Peer structure the bots comunicate with each other. The botmaster can use any bot to give a new command to the network.

Botnet detection and defence

Same propagation mechanism as other malware
Network monitoring
Honeypots
Botnets make use of the same propagation mechanisms as other malware, which mean thay can be detected by antivirus programs. Network monitoring uses software to scan network traffic for known botnet patterns. Only efficient against botnets that do not blend in. Honeypots are machines that are infected on purpose. The behaviour and patterns of the botnet can then be studied by researchers.

Botnet detection and defence (II)

Remove servers
Hijack a sever
Index poisoning
Sybil attack
The easiest way to neutralise of a Command-and-Control botnet is to get rid of the server it depends on. If the bots cannot connect to the server, they will not be able to do anything malicious. A riskier strategy is to try to take over a server. This effectively gives you control over at least a part of the botnet. You can then try to get the botnet to destroy itself. Peer-to-Peer networks have no centralized server, so the previously mentioned defenses are not affective against them. This is where the next two attacks come in. Index poisoning is effecive against bot that uses the Peer-to-Peer architecture to pass on commands. The bots insert commands into the peer-to-peer table under predetermined indices to allow other bots to retrieve them. Adding other data under the same indices will prevent bots from retrieving the commands. A sybil attack inserts or replaces nodes in a P2P network in an attempt to monitor or change botnet traffic.

Cloud computing

> Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

Cloud essential characteristics

On-demand self service.
Broad access
Resource pooling
Rapid elasticity
Measured service
On demand self service: system will be given required resources without human intervention.
Broad access: cloud instance can be acessed through the internet on various devices.
Resource pooling: all available resources are added into a pool, from which any instance can draw.
Rapid elasticity: instances can acquire resources on-demand, allowing them t efficiently handle surges.
Measured service: clients only pay for resources they consume.

Cloud service models

Software as a Service
Platform as a Service
Infrastructure as a Service

Software as a Service
Platform as a Service
Infrastructure as a Service

Deployment models

Public
Private
Hybrid
Community

Public
Private
Hybrid
Community

Problem statement (reminder)

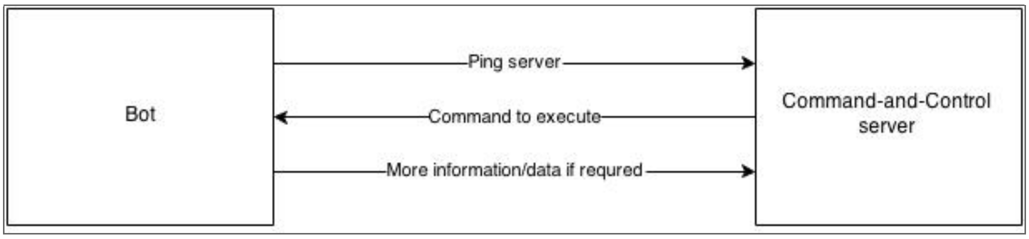
Companies can not easily monitor private cloud environments without changing the underlying architecture of the cloud.
Just as a reminder: the problem statement. The soluion must allow companies to monitor employee usage of cloud resources.

The solution

Use a botnet to collect data about employee usage from cloud instances.
The solution: use a botnet to send usage data to a central service.

Solution

Command and control topology
Bot initiates contact

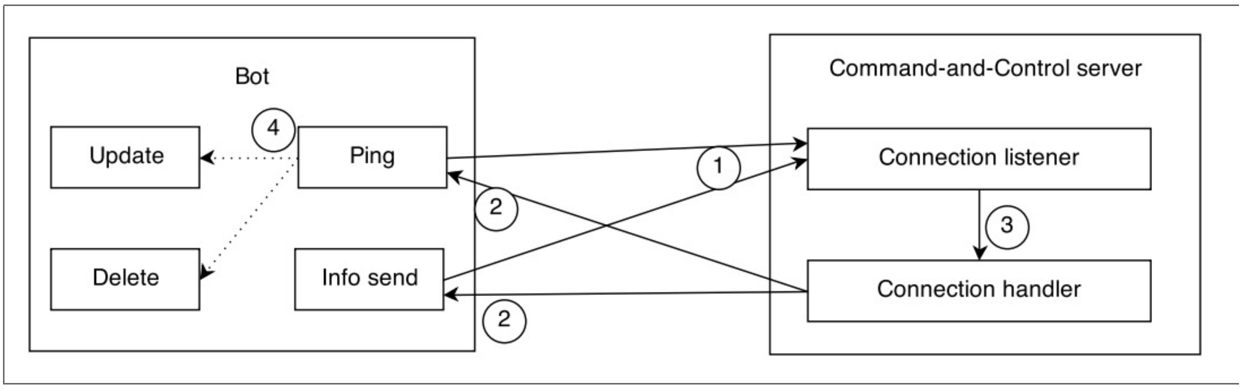


The solution uses a command-and-control topology, since data needs to be gathered in one place. The bot initiates contact, since it might be behind a NAT or firewall that prevents the server from initiating.

Solution features

Collect usage data
Update
Delete
Report
Extensibility
The botnet has various features, the most important one being the ability to gather usage data. For this proof of concept I implemented a keylogger. The bot can also update itself, to fix issues or add new modules or delete itself if the solution isn't needed anymore. The bots report the data they gathes to the central command and control server. The botnet was built with extensibility in mind, allowing users to easily add more functionality.

Solution components



This images shows the components and their interaction.
The bot initiates a connection by pinging the server (1)
Server hands off connection to separate thread (3)
Thread passes info back to the bot (2)
Bot passes the info to the correct module (4)

Communication protocol

TCP
All communication starts

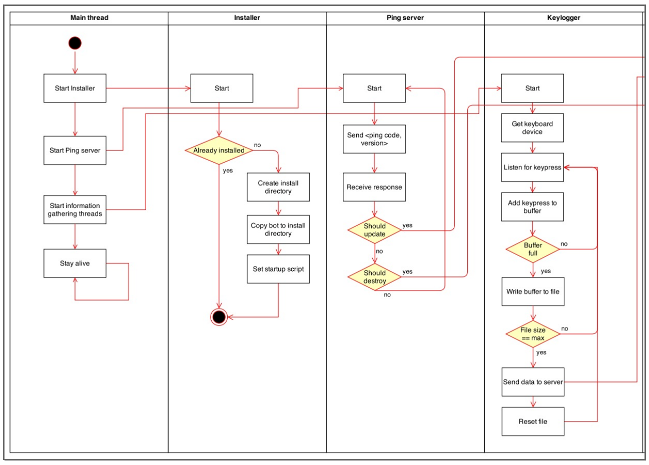
- Command code
- Bot version
  The bot initiates the TCP connection and pass a specific command code and the bot version to the server.

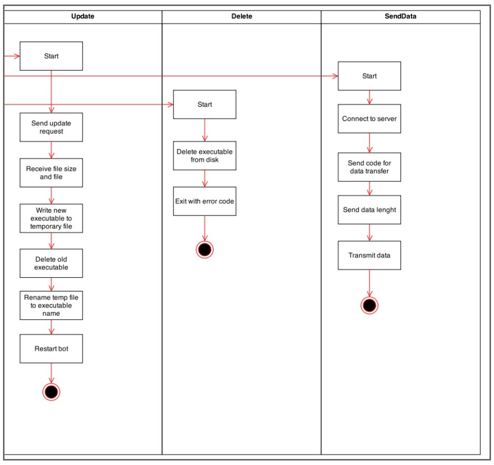
Command codes

P	Ping
S	Transmit data
U	Update
D	Destroy

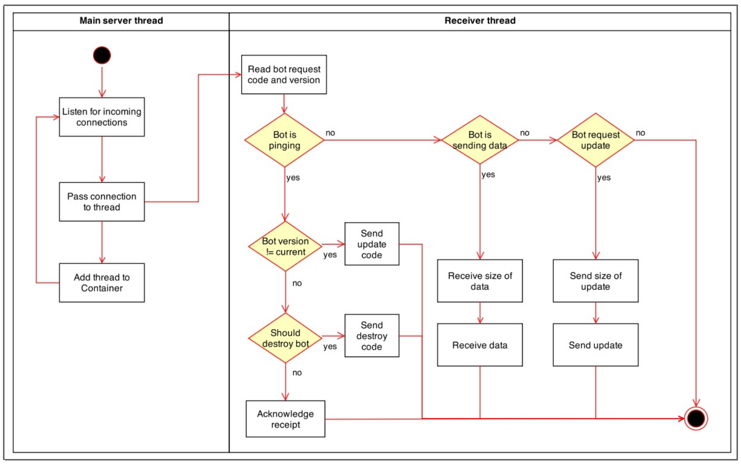
P	Ping
S	Transmit data
U	Update
D	Destroy

Bot program flow




Update
Delete
SendData
Start
Send update request
Receive file size and file
Write new executable to temporary file
Delete old executable
Rename temp file to executable name
Restart bot
Start
Delete executable from disk
Exit with error code
Start
Connect to server
Send code for data transfer
Send data lenght
Transmit data

Server program flow



Problem statement (again)

Companies can not easily monitor private cloud environments without changing the underlying architecture of the cloud.

Evaluation: Advantages

Self-propagating
Centralised data
Extensibility

Self-propagating
Centralised data
Extensibility

Evaluation: Disadvantages

No guarantee of 100% coverage
Too much data
Network traffic

No guarantee of 100% coverage
Too much data
Network traffic

Suggestions for future work

Propagation mechanisms

Data handling

Alternative uses

Propagation mechanisms

Data handling

Alternative uses

Questions?