Malware

Spring 2017

Botnets

Sergii Lysenko, PhD

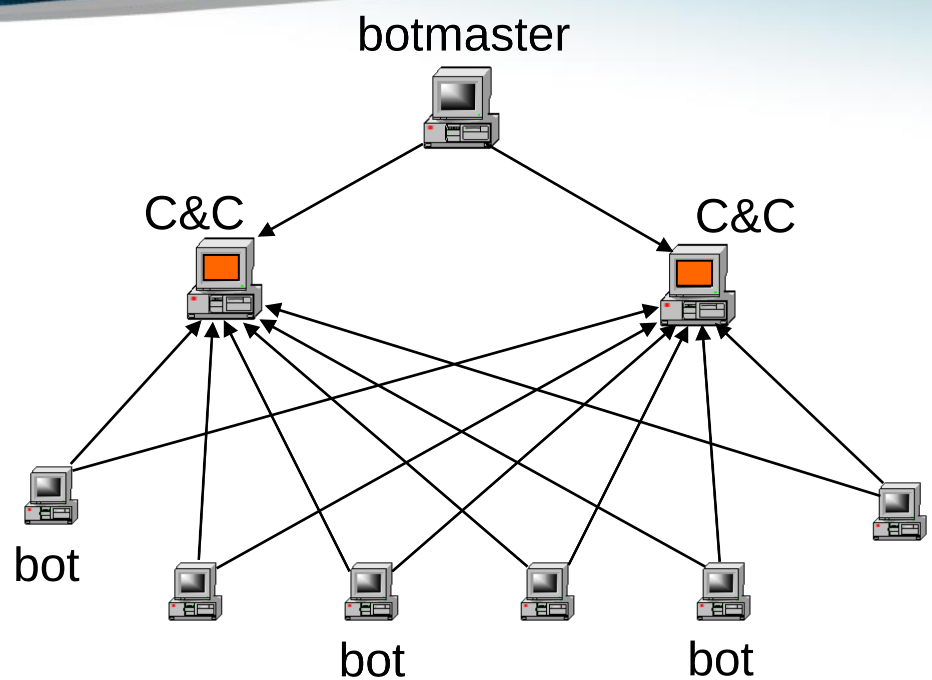
Co-funded by the Tempus Programme of the European Union

What Is a Botnet?

- Botnet: bot + network
  - Bot: compromised machine installed with remote controlled code
  - Networked bots under a single commander (botmaster, botherder)

- Botnet is the major threat nowadays
  - Large-scale worm attacks are old news
  - Profit: motivation for most attackers
    - Spam, phishing, ID theft, DoS blackmail
    - Botmaster with thousands of machines at command has attack power

Current Botnet Command & Control Architecture



- Bot periodically connects to one/some of C&C servers to obtain command
  - Hard-coded IPs or DNS names of C2 servers
- C&C: usually Internet Relay Chat (IRC) based

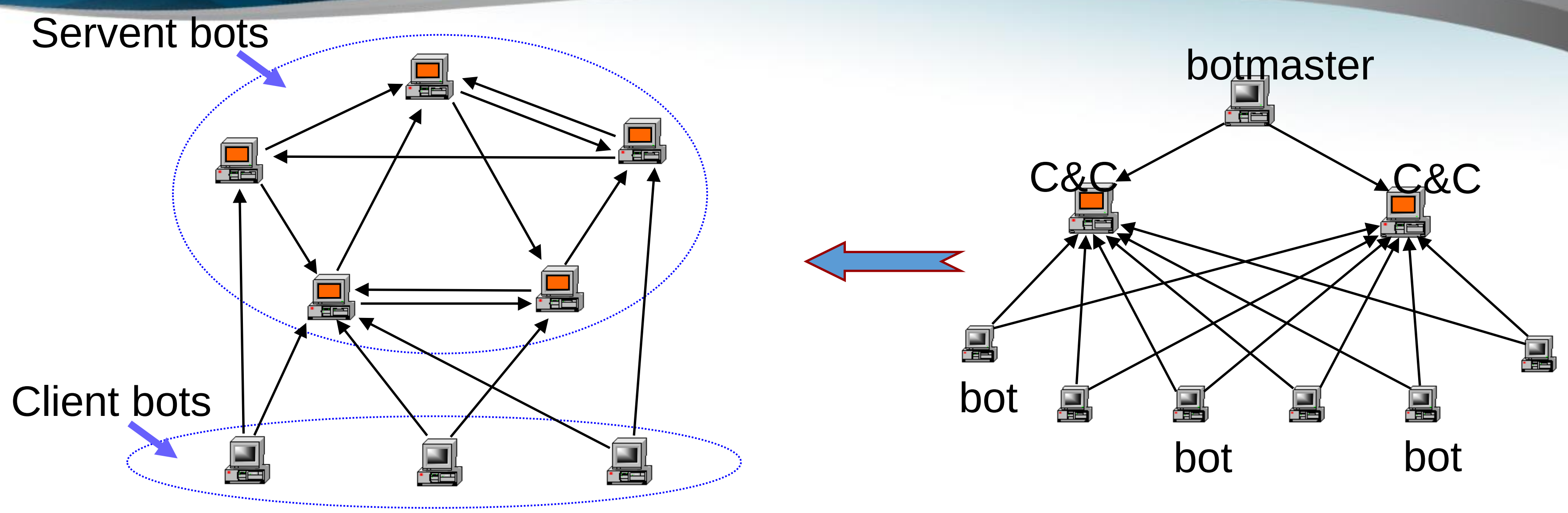
Three Possible types of Botnets

- Peer-to-peer structured botnets
  - More robust C2 architecture
  - We present a hybrid P2P botnet
- Honeypot-aware botnets
  - Honeypot is popular in malware defense
  - A general principle to remove inside honeypot spies
- Stealthy botnets
  - Keep bots as long as possible
  - We study “rootkit” techniques

Peer-to-Peer (P2P) based Control Architecture

- Weakness of C&C botnets
  - A captured bot (e.g., honeypot) could reveal all C2 servers
  - The few C2 servers can be shut down at the same time
  - A captured/hijacked C2 server could reveal all members of the botnet
- C&C centralized → P2P control is a natural evolution
  - P2P-based network is believed to be much harder to shut down

Hybrid P2P Botnet



- Bots: static IPs, able to receive incoming connections
  - Static IP ensures a stable, long lifetime control topology
- Each bot connects to its "peer list"
  - Only servent bot IPs are in peer lists

Botnet Monitor by Botmaster

- Botmasters know their weapons
  - Botnet size
  - bot IPs, types (e.g., DHCP ones used for spam)
  - Distribution, bandwidth, diurnal …
- Monitor via ***dynamical*** sensor
  - Sensor IP given in a monitor command
  - *One sensor, one shot, then destroy it*
    - Use a sensor's current service to blend incoming bot traffic

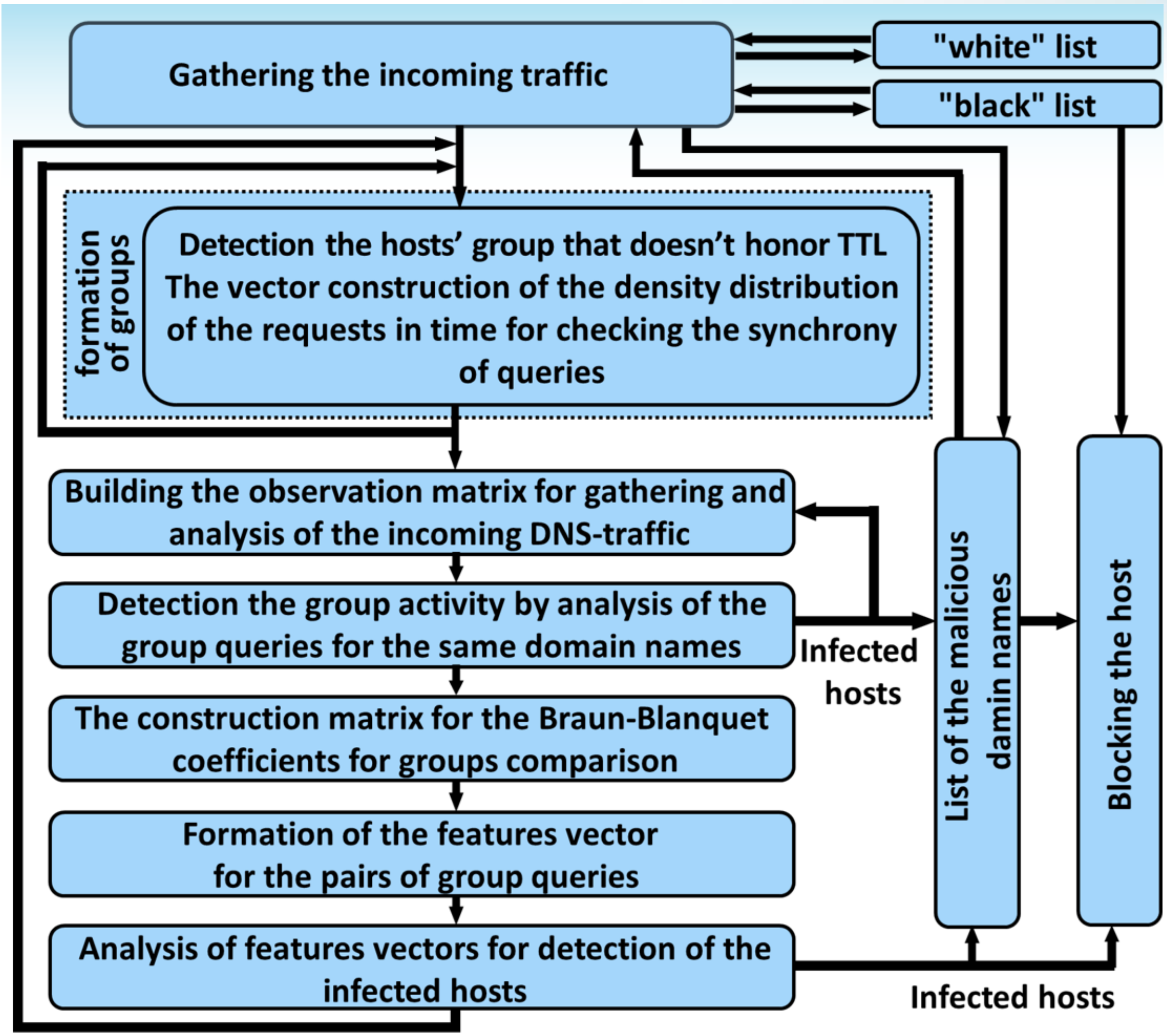
P2P Botnet Construction

- Botnet networked by peer list
- Basic procedures
  - New infection: pass on peer list
  - Reinfection: mix two peer lists
    - Ensure balanced connectivity
- Remove the normal P2P bootstrap
  - Or, increase entries in bootstrap as botnet grows

P2P Botnet Construction

- Peer-list updating procedure
  - Obtain current bots information
  - Request every bot connect to a sensor to obtain a new peer list
- Result: all bots have balanced connectivity to bots used in this procedure
  - Use once is enough for a robust botnet
  - Can be used to reconnect a broken botnet

A scheme of the botnet detection
Passive monitoring

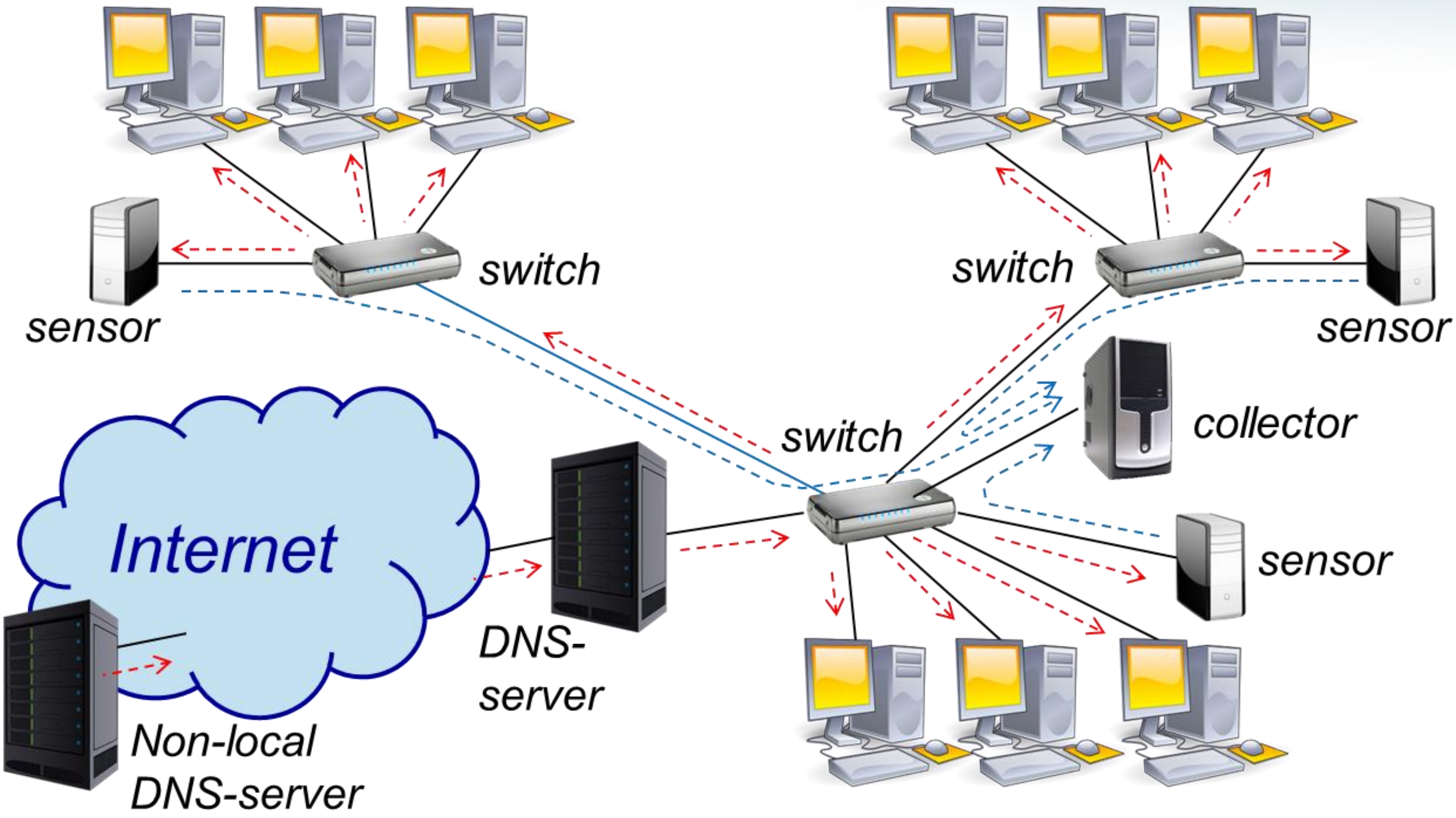


The method takes into account abnormal behaviors of the hosts' group, which are similar to botnets' behavior:

- hosts' group does not honor DNS TTL (flush local DNS-cache
- carry out repeated queries for domain names before TTL expiration
- implement the DNS-queries to non-local DNS-servers

Botnet Detection Process

Gathering the incoming traffic



Botnet Detection Process

Detection the hosts' group that does not honor TTL

If hosts' group is flushing the local hosts DNS-caches it means that the hosts' group does not honor TTL. In order to detect that fact the observation matrix is built.

Each row contains the hosts' MAC-addresses that requested the specific domain name during the TTL (so they possibly carry out a group activity)

Observation matrix V_{MAC}

MAC-addresses

	H_1	H_2	H_3	H_4	…	H_j
request to the specific domain name di	1	1	1	1	…	1
repeated request	1	0	1	1	…	0

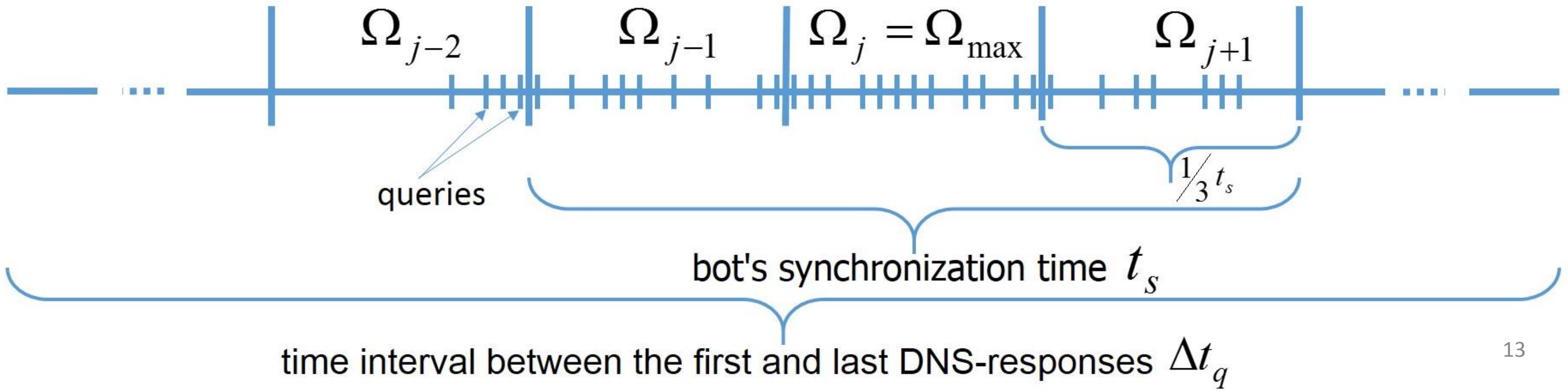
Botnet Detection Process

The vector construction of the density distribution of the requests in time for checking the synchrony of queries

We will consider the group of queries as synchronous if we observe the greatest number of queries for the domain name during the time when the bots of the botnet are performing queries - bot's synchronization time t_s. In order to check the synchrony of queries of the DNS-queries we divide the interval between the first and last DNS-responses Δt_q is into z intervals:

$$z = \left(t_{last} - t_{first}\right) / \tfrac{1}{3} t_s,$$

where tlast and tfirst - time of the last and first DNS-responses for domain name d_i within the TTL, during which the group activity is searching or the group flushing of local DNS-caches is fixed.

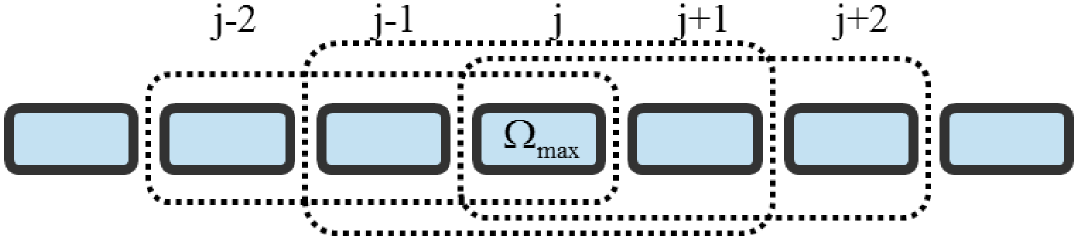


Botnet Detection Process

The vector construction of the density distribution of the requests in time for checking the synchrony of queries

For group query we build the vector of density distribution of z-elements for queries in time $\overline{W_{d_i}} = (\Omega)_{j=1}^{z}$, where Ω_j – number of queries within the z-th interval.

For the element of vector $\overline{W_{d_i}}$ with a maximum value Ω_{max} within j = max±2, we find two adjacent elements with the largest values so that all three elements could describe the query distribution of continuous interval, and then we calculate their sum (Sum_s). If $(1-\delta)\cdot Sum_s > Sum_r$, then the group query is the subject to further analysis, otherwise such group is discarded, where Sum_r - the sum of other vector elements $\overline{W_{d_i}}$



If queries are synchronous, the sets of MAC-addresses in the matrix V_{MAC} hosts groups are combined.

Botnet Detection Process

Building the observation matrix M_k
for analysis the incoming DNS-traffic

MAC-addresses of hosts' groups received from the matrix V_{MAC}

number of hosts in the group

query to local/non-local DNS-servers

sign of repeated request within TTL-period

NXDOMAIN error code in DNS-responses

hosts' group is "infected" or "suspicious"

number of iteration with the "suspicious " sign

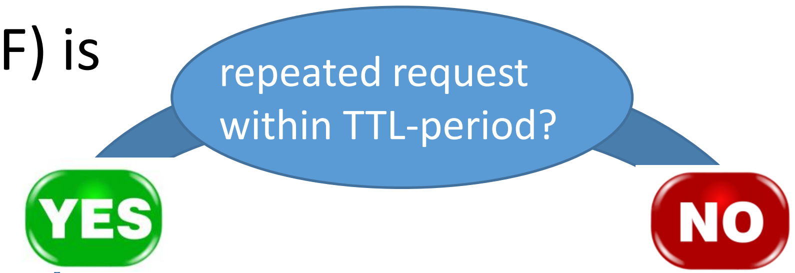
Domain names

	H_1	H_2	…	Hj	N_G	S	F	R	M	N
d_1	1	1	…	1	5	1	1	0	0	0
d_1	1	0	…	1	4	1	0	0	0	0
d_2	1	0	…	1	6	0.5	0	0	0	0
…	…	…	…	…	…	…	…	…	…	…

Botnet Detection Process

Building the observation matrix
for analysis the incoming DNS-traffic

If the repeated requests within TTL-period were observed, then the cell of the matrix observation $M_k(di,F)$ is marked as "1", otherwise - "0".



MAC-addresses of hosts' groups received from the matrix V_{MAC}

Domain names

	H_1	H_2	…	Hj	L	S	F	R	M	N
d_1	1	1	…	1	5	[illegible]	1	0	0	0
d_1	1	0	…	1	4	1	0	0	[illegible]	0
d_2	1	0	…	1	6	0.5	0	[illegible]	0	0
…	…	…	…	…	…	…	…	…	…	…

Botnet Detection Process

Building the observation matrix Mk
for analysis the incoming DNS-traffic

If the hosts' group have been requesting the domain name d_i to **a local** and **other DNS-servers**, then the cell of observation matrix $M_k(d_i,S)$ is marked as "0", if **only** to the **local DNS-server** - "0.5",
if only to a **non-local DNS-servers** – "1".

request to only local DNS-server

request to local and other DNS-servers

request to non-local DNS-servers

Domain names

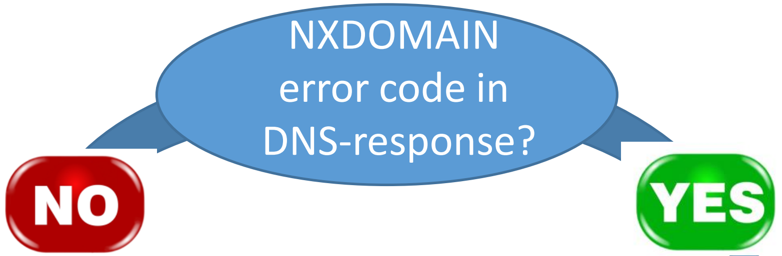
	H_1	H_2	…	Hj	N_G	S	F	R	M	N
d_1	1	1	…	1	5	1	[illegible]	0	0	0
d_1	1	0	…	1	4	0	0	0	0	0
d_2	1	0	…	1	6	0.5	0	[illegible]	0	0
…	…	…	…	…	…	…	…	…	…	…

Botnet Detection Process

Building the observation matrix Mk
for analysis the incoming DNS-traffic

If the DNS-responses for this group contain NXDOMAIN error code, then the cell of observation matrix M_k(di, R) is marked as "1", otherwise - "0".

The cell $M_k(d_i,F)$ will be filled at a next stage



Domain names

	H_1	H_2	…	Hj	N_G	S	F	R	M	N
d_1	1	1	…	1	5	1		0	0	0
d_1	1	0	…	1	4	1	0	0	0	
d_2	1	0	…	1	6	0.5	0	1		0
…	…	…	…	…	…	…	…	…	…	…

Botnet Detection Process

The similarity evaluation of hosts' groups

Comparison of the ***two groups*** of hosts G_1 and G_2, that sent the DNS-queries for two domain names d1 and d2 at time intervals Δt_1 and Δt_2 respectively, using the Braun-Blanquet coefficient:

$$K_B(G_1, G_2) = \frac{N_o}{\max\left[N_{G_1}, N_{G_2}\right]},$$

where N_o - the number of common elements in groups G_1 and G_2; N_{G_1} and N_{G_2} - the number of hosts in groups G_1 and G_2, respectively, $K_B(G_1, G_2) \in [0,1]$.

If the number of compared groups is ***more than two*** the Koch index of dispersity is used:

$$K_K(G_1, \ldots, G_q) = \frac{C - A}{(q-1)\cdot A}$$

where $G_1, \ldots, G_q$ - comparable groups of hosts; q - the number of comparable groups; $C = \sum_{i=1}^{q} N_{G_i}$ - total number of MAC-address in all groups; A - number of different MAC-addresses presented in groups; $K_K(G_1, \ldots, G_q) \in [0,1]$.

Botnet Detection Process

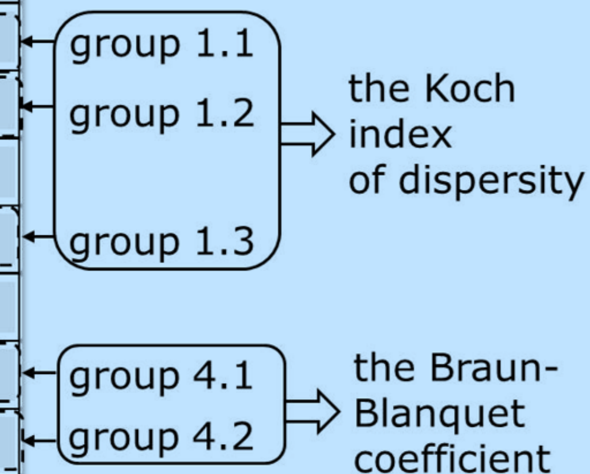
Detection the group activity by analysis of the group queries for the same domain names

Detection the groups' queries is made by comparison the groups by MAC-adresses.

Braun-Blanquet coefficient to compare ***two*** groups or ***dispersion index Koch*** for ***3 or more*** groups.

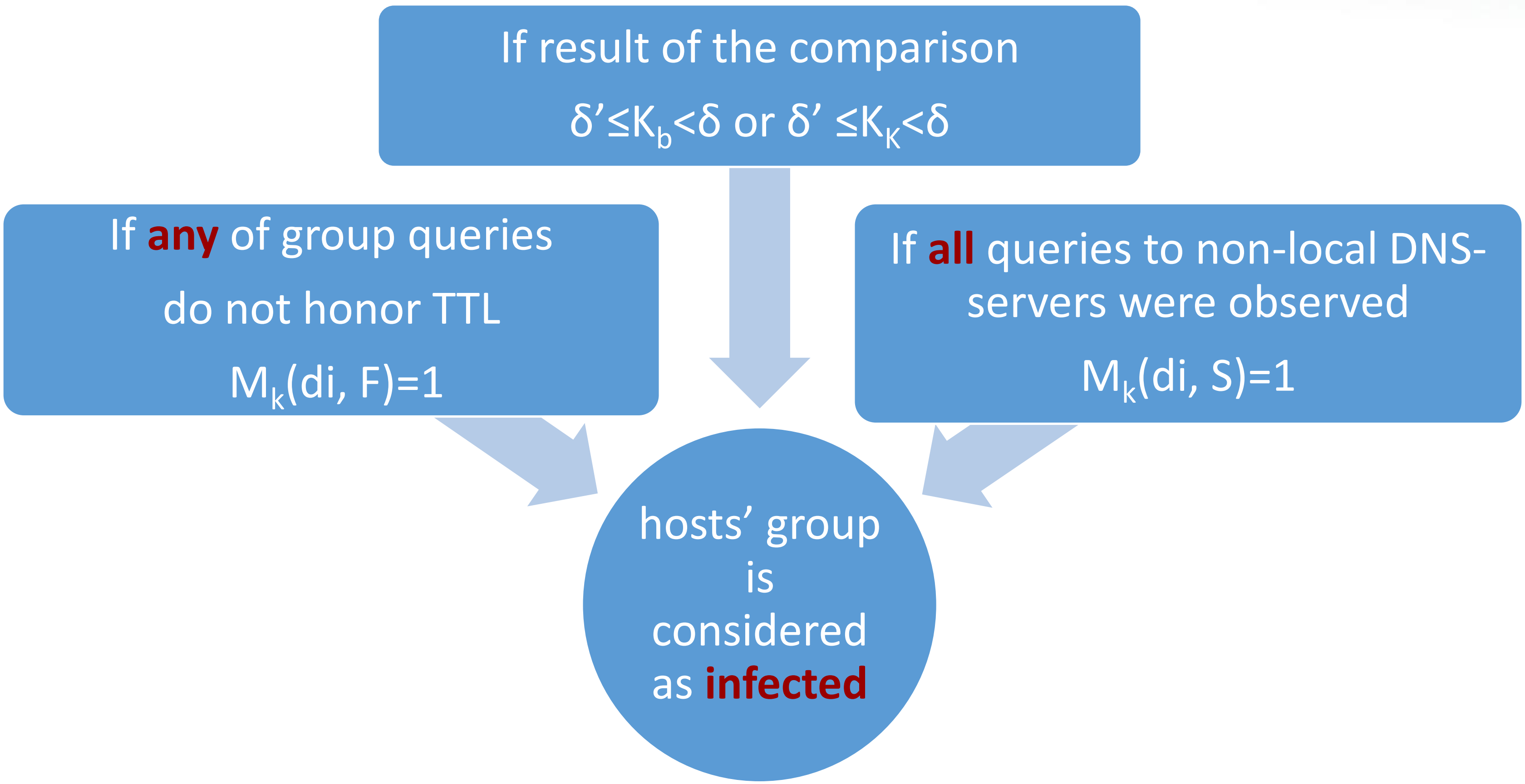
If the result of comparison exceeds the threshold $K_B \geq \delta$ or $K_K \geq \delta$, the hosts' group is considered as ***infected***, if $\delta` \leq K_B < \delta$ or $\delta` \leq K_K < \delta$ the hosts' group is considered as ***suspicious***

	H_1	H_2	…	Hj	N_G	S	F	R	M	N
d_1	1	1	…	1	5	1	1	0	0	0
d_1	1	0	…	1	4	1	0	0	0	0
d_2	1	0	…	1	6	0.5	0	0	0	0
d_1	1	1	…	1	5	1	1	0	0	0
d_3	1	1	…	1	4	0	0	1	0	0
d_4	1	1	…	1	5	0	0	0	0	0
d_4	1	1	…	1	7	0	0	0	0	0
d_5	1	1	…	1	4	0	0	0	0	0



Botnet Detection Process

An additional analysis of the observation matrix Mk when group queries do not honor TTL and use non-local DNS-servers



Botnet Detection Process

	H_1	H_2	…	Hj	N_G	S	F	R	M	N
d_1	1	1	…	1	5	1	1	0	0	0
d_1	1	0	…	1	4	1	0	0	0	0
d_2	1	0	…	1	6	0.5	0	0	0	0
d_1	1	1	…	1	5	1	1	0	0	0
d_3	1	1	…	1	4	0	0	1	0	0
d_4	1	1	…	1	5	0	0	0	0	0
d_4	1	1	…	1	7	0	0	0	0	0
d_5	1	1	…	1	4	0	0	0	0	0

group 1.1
group 1.2
group 1.3
⇒ the Koch index of dispersity

group 4.1
group 4.2
⇒ the Braun-Blanquet coefficient

	H_1	H_2	…	Hj	N_G	S	F	R	M	N
d_1	1	1	…	1	5	1	1	0	1	0
d_2	1	0	…	1	6	0.5	0	0	0	0
d_3	1	1	…	1	4	0	0	1	0	0
d_4	1	1	…	1	7	0	0	0	0.5	0
d_5	1	1	…	1	4	0	0	0	0	0

group 1
group 4

If groups are defined as infected or suspicious, we combine the set of MAC-addresses into one row for the domain name d in the matrix Mk

Botnet Detection Process

Further filling of the observation matrix Mk

If the hosts' group is identified as **infected** the cell of the observation matrix Mk(d;M) is marked as "1", and as "0.5" - if it was defined as **suspicious**.

hosts' group is identified as suspicious

hosts' group is Identified as infected

Domain names

	H_1	H_2	…	Hj	N_G	S	F	R	M	N
d_1	1	1	…	1	5	1	1	0	1	[illegible]
d_2	1	0	…	1	6	0.5	0	0	0	0
d_3	1	1	…	1	4	0	0	1	0	0
d_4	1	1	…	1	7	0	0	0	0.5	[illegible]
d_5	1	1	…	1	4	0	0	0	0	0

group 1

group 4

Botnet Detection Process

The construction of the lower triangular matrix for the Braun-Blanquet coeficients

We built ***the lower triangular matrix*** for the Braun-Blanquet coefficients B_k.
The rows of the matrix B_k are formed by increasing number of MAC-addresses in groups N_G.
The Braun-Blanquet coefficients are filled in the matrix, which were calculated for pairs of hosts' groups.
Calculation of the values for the column cells is terminated if $N_{G_i} / N_{G_{i+1}} < \delta'$.

the lower triangular matrix for the Braun-Blanquet coeficients

	d_3	d_5	d_1	d_2	d_4	…	N_G	S	F	R	M	N
d_3	1					…	4	0	0	1	0	0
d_5	1	1				…	4	0	0	0	0	0
d_1	0.8	0.8	1			…	5	1	1	0	1	0
d_2			0.5	1		…	6	0.5	0	0	0	0
d_4			0.71	0.43	1	…	7	0	0	0	0.5	0

B_k

Botnet Detection Process

Formation of the features vector for the pairs of group queries

For each pair of group queries when $K_B \geq \delta`$ from the matrix B_k we form the features vector $\overline{W_{G_1,G_2}}$, which can be defined

$$\overline{W_{G_1,G_2}} = \left(K_B(G_1,G_2), S_{G_1,G_2}, F_{G_1,G_2}, R_{G_1,G_2}, M_{G_1,G_2}\right),$$

where $S_{G_1,G_2}, F_{G_1,G_2}, R_{G_1,G_2}, M_{G_1,G_2}$ - behavioral features for two compared groups

Combined behavioral features for two compared groups, obtained from the matrix Bk

Botnet Detection Process

Definition of the combined behavioral features

Behavioral feature for:
the sign of query
to local/non-local
DNS-servers

$$S_{G_1,G_2}=\begin{cases} Unusual, if\ B_k(d_1,S)=B_k(d_2,S)=0, \\ Neutral, if\ B_k(d_1,S)=B_k(d_2,S)=0.5, \\ Dangerous, if\ B_k(d_1,S)=B_k(d_2,S)=1, \\ Suspicious\ \ otherwise. \end{cases}$$

Behavioral feature for:
the sign of the hosts' group "infected" or "suspicious", obtained at intermediate stages of analysis

$$M_{G_1,G_2}=\begin{cases} Neutral\ , if\ B_k(d_1,M)=B_k(d_2,M)=0, \\ Suspicious\ \ , if\ ((B_k(d_1,M)=0.5 \vee B_k(d_2,M)=0.5)\wedge \\ \wedge B_k(d_1,M)\neq 1 \wedge B_k(d_2,M)\neq 1)\wedge B_k(d_1,M)\neq B_k(d_2,M), \\ Dangerous\ \ , if\ B_k(d_1,M)=1 \vee B_k(d_2,M)=1 \vee \\ \vee (B_k(d_1,M)=B_k(d_2,M)=0.5 \wedge B_k(d_1,N)\neq B_k(d_2,N) \vee \\ \vee B_k(d_1,N)=B_k(d_2,N)=0), \end{cases}$$

Behavioral feature for:
the sign of repeated request
Within TTL-period

$$F_{G_1,G_2}=\begin{cases} Neutral, if\ B_k(d_1,F)=B_k(d_2,F)=0, \\ Suspicious, if\ B_k(d_1,F)\neq B_k(d_2,F), \\ Dangerous, if\ B_k(d_1,F)=B_k(d_2,F)=1. \end{cases}$$

Botnet Detection Process

Formation of the features vector for the pairs of group queries

	d_3	d_5	d_1	d_2	d_4	...	N_G	S	F	R	M	N
d_3	1					...	4	0	0	1	0	0
d_5	1	1				...	4	0	0	0	0	0
d_1	0.8	0.8	1			...	5	1	1	0	1	0
d_2			0.5	1		...	6	0.5	0	0	0	0
d_4			0.71	0.43	1	...	7	0	0	0	0.5	0

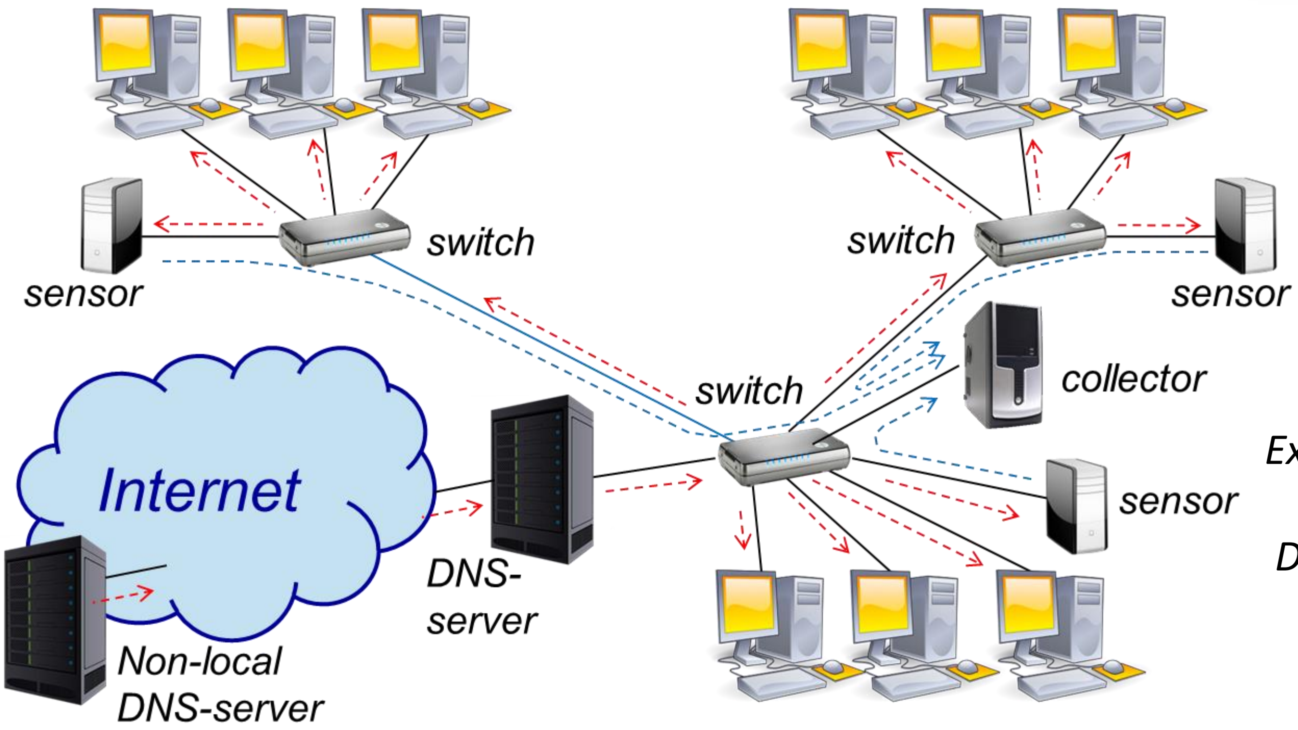
$K_b(G_1,G_4)$	$S_{G1,G4}$	$F_{G1,G4}$	$R_{G1,G4}$	$M_{G1,G4}$
0.71	Suspicious	Suspicious	Neutral	Dangerous

Analysis of the features vectors
is performed by the following rules:

$$f\left(\overline{W_{G_1,G_2}}\right)=\begin{cases} Not\_Infected, if\ K_B(G_1,G_2)<\delta \wedge S_{G_1,G_2}=Unusual \wedge \forall\overline{W_{G_1,G_2}}(j)\neq Suspicious\ \wedge \forall\overline{W_{G_1,G_2}}(j)\neq Dangerous, \\ Not\_Suspicious, if\ K_B(G_1,G_2)<\delta \wedge S_{G_1,G_2}\neq Unusual \wedge \forall\overline{W_{G_1,G_2}}(j)\neq Suspicious\ \wedge \forall\overline{W_{G_1,G_2}}(j)\neq Dangerous, \\ Infected, if\ \exists\overline{W_{G_1,G_2}}(j)=Dangerous\ \vee K_B(G_1,G_2)\geq\delta, \\ Suspicious\ \ else. \end{cases}$$

Experiments

Experimental conditions



bots: the SDBot family

Network: **100** hosts

Experiment time: **8** hours

DNS-traffic capture: tcpdump utility

Detection rate 92%, false positives 5-8%

Motivation

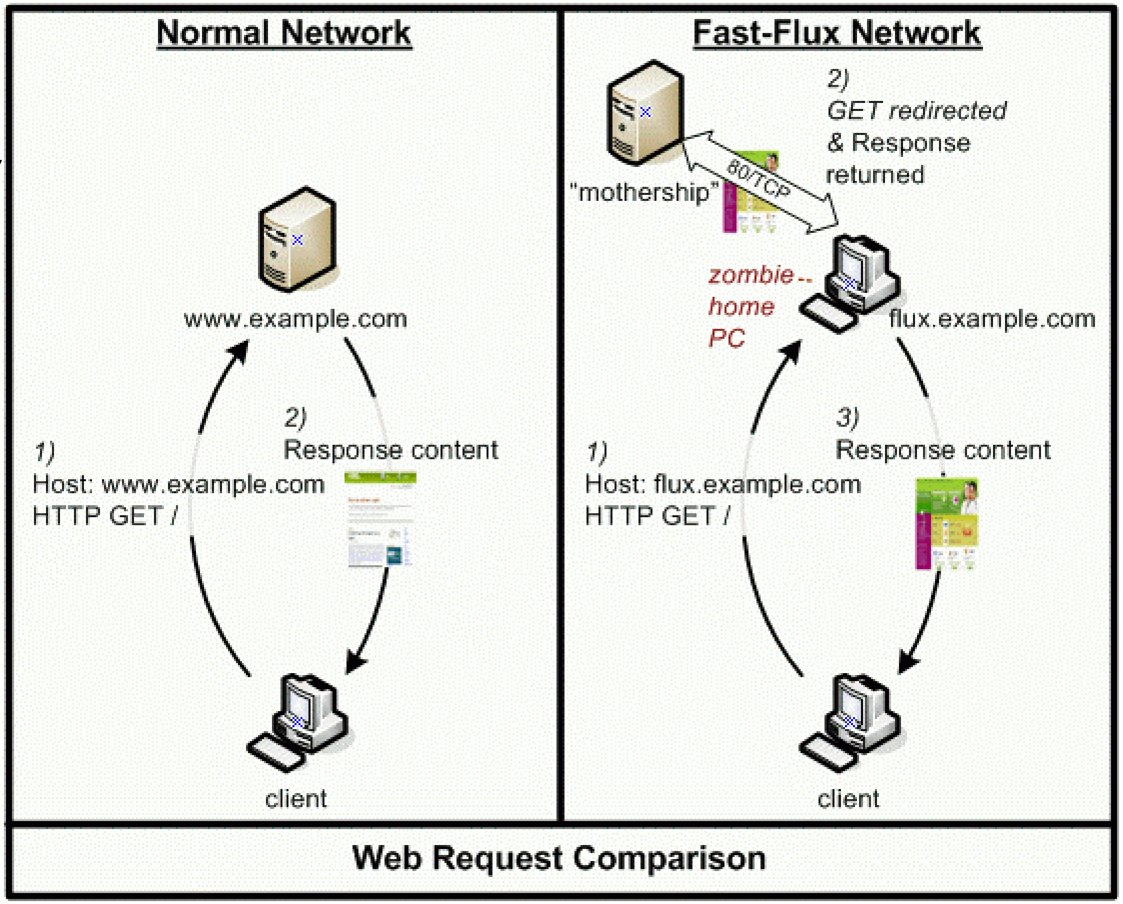
Botnets' evasion techniques

- DNS-tunneling
- fast-flux service network
- "domain flux" technology
- cycling of IP mappings

Botnets' evasion techniques

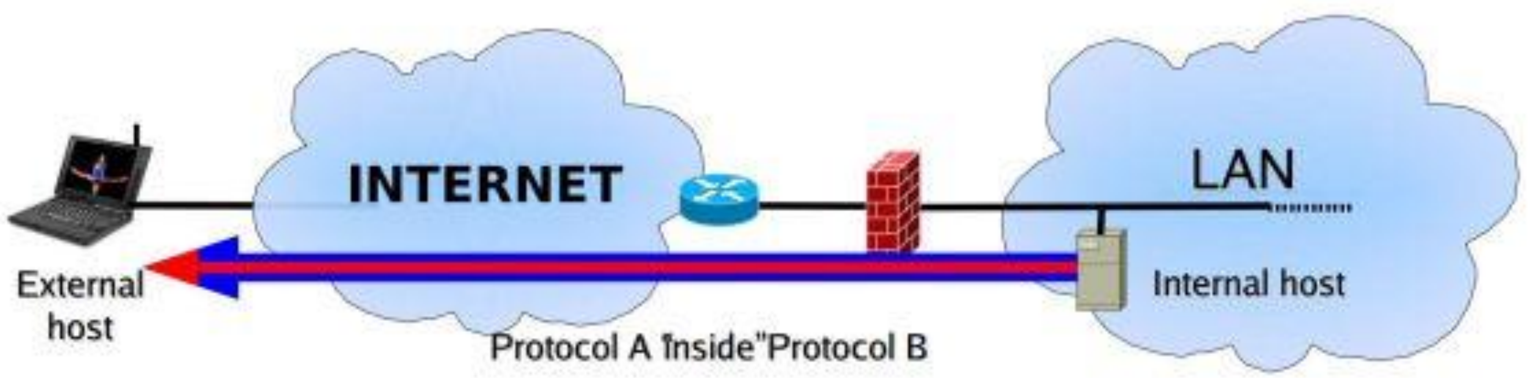
Fast flux service network uses a short TTL-periods and cyclic method of round-robin DNS.

Technique has an ability to evade the "black lists" of DNS

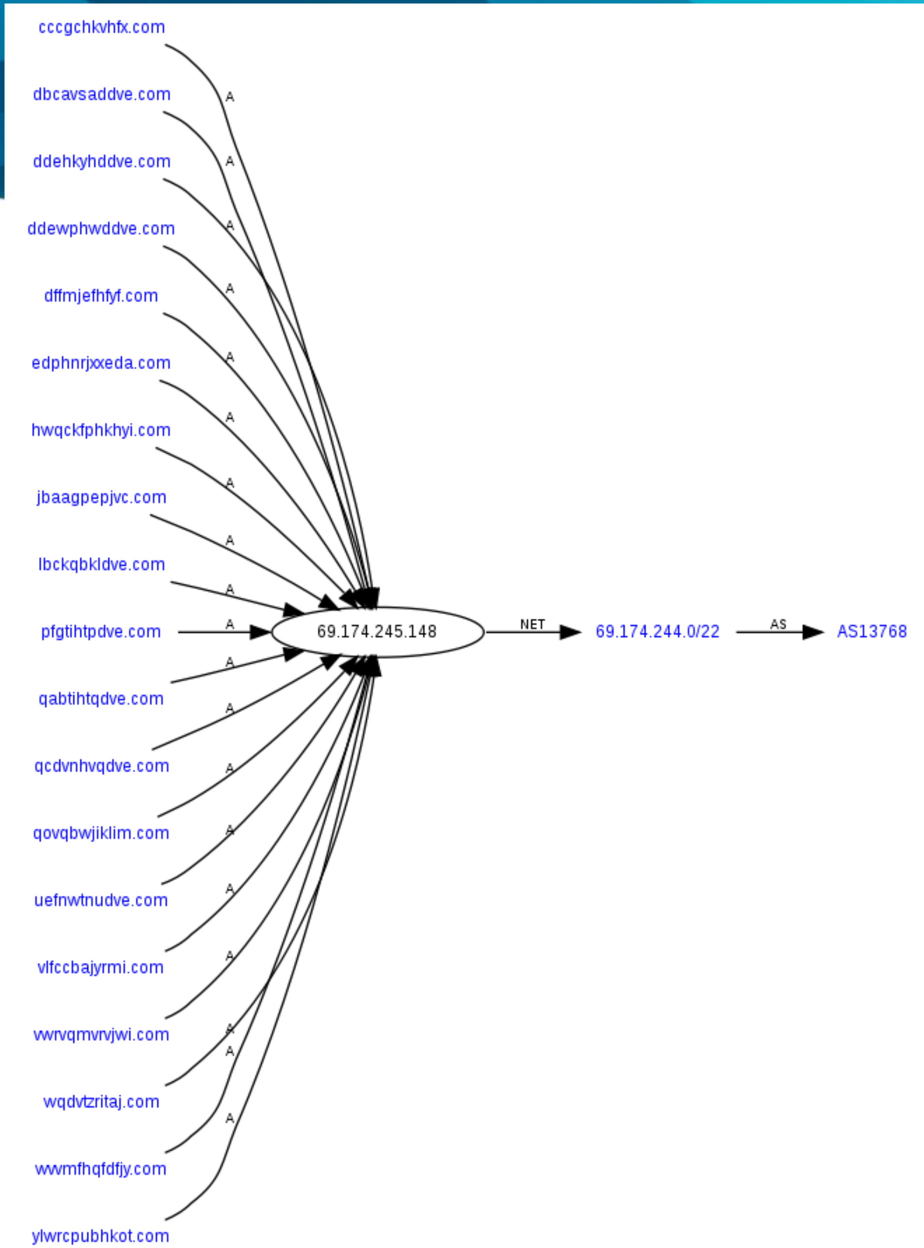


Botnets’ evasion techniques

DNS-tunneling allows an attacker to transmit an arbitrary traffic within the DNS-protocol specification by using the fields of a DNS-message in order to perform the botnet’s command and control

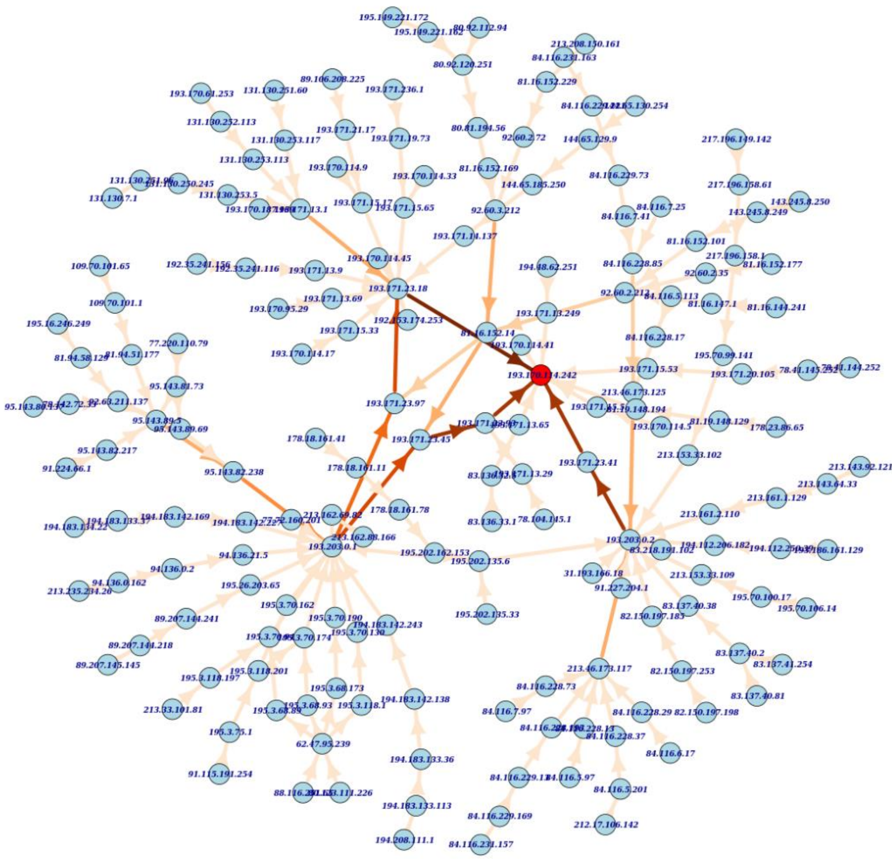


Botnets' evasion techniques



Domain flux - the technology that combines short TTL-periods and frequent changes of C&C-server's domain name

Botnets’ evasion techniques

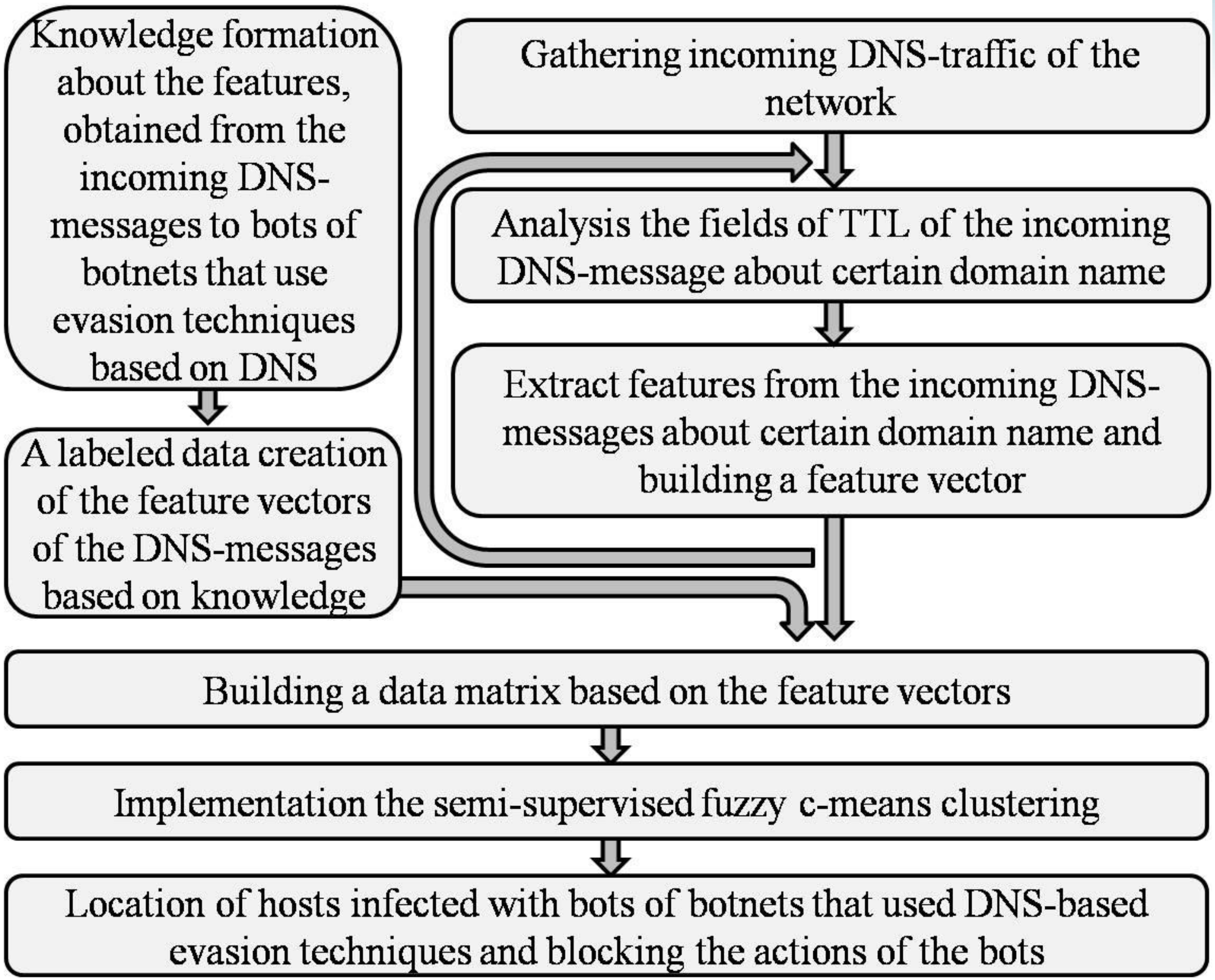


Cycling of IP mappings for the domain name of C&C-server

Anti-evasion technique for botnets detection

- Technique is based on a cluster analysis of the features obtained from the payload of DNS-messages
- The method uses a semi-supervised fuzzy c-means clustering
- Usage of the developed method makes it possible to detect botnets that use the DNS-based evasion techniques with high efficiency

THE SCHEME OF THE DNS-BASED ANTI-EVASION TECHNIQUE FOR BOTNETS DETECTION (PASSIVE MONITORING)



The feature vector of the incoming DNS-message about domain name

$$\overline{W_d} = \begin{pmatrix} L_N, N_U, E_N, T_{\text{mod}}, T_{med}, T_{aver}, N_A, N_{IP}, S_{IP}, \\ S_A, N_{UA}, S_{UA}, N_D, F_{UR}, E_R, L_P, F_S \end{pmatrix},$$

where L_N - the length of the domain name;

N_U - the number of unique characters in the domain name;

E_N - entropy of the domain name;

T_{mod} - TTL-period, mode;

T_{med} - TTL-period, median;

T_{aver} - TTL-period, average value;

N_A – the number of A-records corresponding to domain name in the DNS-message;

N_{IP} - the number of IP-addresses concerned with the domain name;

S_{IP} - the average distance between the IP-addresses concerned with domain name;

S_A - the average distance between the IP-addresses in the set of A-records for domain name in the incoming DNS-message;

The feature vector of the incoming DNS-message about domain name

N_{UA} - number of unique IP-addresses in sets of A-records corresponding to the domain name in the DNS-messages (feature is used if value N_A>1);

S_{UA} - the average distance between unique IP-addresses in sets A-record corresponding to the domain name in the DNS-messages (feature is used if value N_A>1);

N_D – number of domain names that share IP-address corresponding to domain name;

F_{UR} – the sign of the usage of uncommon types of the DNS-records, or DNS-records that are not commonly used by a typical client;

E_R – entropy of the DNS-records, which are contained in the DNS-messages;

L_P – maximum size of the DNS-messages about domain name;

F_S - the sign of success of DNS-query ($F_S = 0$ if DNS-query failed, and $F_S = 1$ if DNS-query was successful)

f_{E_B} - the dependence function of the DNS-message field entropy of its length

Evasion techniques' concern

Knowledge about evasion technique based on the features inherent to the DNS-message to bots presented as the rules:

$$if\ T_{\mathrm{mod}} \in [0{,}900]\ and\ \ T_{med} \in [0{,}900]\ and$$
$$T_{aver} \in [0{,}900]\ and\ F_S = 0\ and\ N_D \in [8;\infty] \Rightarrow$$
$$\Rightarrow domain\_\ flux$$

$$if\ \ (T_{\mathrm{mod}} \in [0{,}900]\ and\ \ T_{med} \in [0{,}900]\ and$$
$$\text{and}\ T_{aver} \in [0{,}900])\ and$$
$$and\,((N_A \in (5,\infty)\ and\ \ S_A \in (65535,\infty))or$$
$$or(N_{UA} \in (8,\infty)\ and\ \ S_{UA} \in (65535,\infty)))\ \Rightarrow$$
$$\Rightarrow fast\_flux$$

Evasion techniques' concern

Knowledge about evasion technique based on the features inherent to the DNS-message to bots presented as the rules:

$$if\ (L_N \in [75{,}255]\ and\ \ N_U \in (27{,}37])\ or\ E_N \geq f_{E_{B32}}\ or$$
$$or\ \ (E_R \geq f_{E_{B64}}\ or\ \ E_R \geq f_{E_{B256}})\ or\ \ F_{UR} = 1\ or$$
$$or\ L_P > 300 \Rightarrow DNS\_tunneling$$

$$if\ T_{\mathrm{mod}} \in [0{,}900]\ and\ \ T_{med} \in [0{,}900]\ and$$
$$and\ T_{aver} \in [0{,}900]\ and\ N_{IP} \in (5, \infty)\ and$$
$$S_{IP} \in (65535, \infty) \Rightarrow cycling\ \ of\ IP\ mappings$$

Performing of the semi-supervised learning

A labeled data creation of the feature vectors of the DNS-messages based on knowledge

Let denote $X = \{x_i\}_{i=1}^{N_x}$ the labeled data,

$Y = \{y_i\}_{i=1+N_x}^{N_z}$ as unlabeled data, where N_x - the number of objects in the labeled data, N_z - the total number of different domain names

Let denote $H = \{h_i\}_{i=1}^{N_h}$ as a set of predefined clusters of objects,

N_h - the number of clusters:

- **h_1 - cycling of IP mapping,**
- **h_2 – domain flux,**
- **h_3 – fast flux,**
- **h_4 – DNS-tunneling,**
- **h_5 – cluster that contains normal queries**

Each feature vector of labeled data belongs to one of the predefined clusters

Botnet Detection Process

Building a data matrix based on the feature vectors

Using the feature vectors $\overline{W_d}$ we form the incoming DNS-messages, the data matrix V.

Each line of matrix V is the feature vector of incoming DNS-messages $\overline{W_d}$ about certain domain name, $V = \left(v_{ij}\right)_{i=1,\, j=1}^{N_z,\, N_q}$, $V(i,) = \overline{W_d}$,

where N_q - the total number of features of the incoming DNS-messages that indicate the use the evasion techniques by bots of botnets

Botnet Detection Process

Implementation the semi-supervised fuzzy c-means clustering for identifying the queries in the network that may indicate the functioning of the bots of botnets

The objects of the clustering are the feature vectors, obtained from payload of the incoming DNS-messages about certain domain name

The result of clustering is a degree of membership of the feature vectors to one of four clusters, where the membership of feature vector $\overline{W_d}$ to cluster h_i, $i=\overline{1,4}$ indicates the queries executing using the evasion techniques.

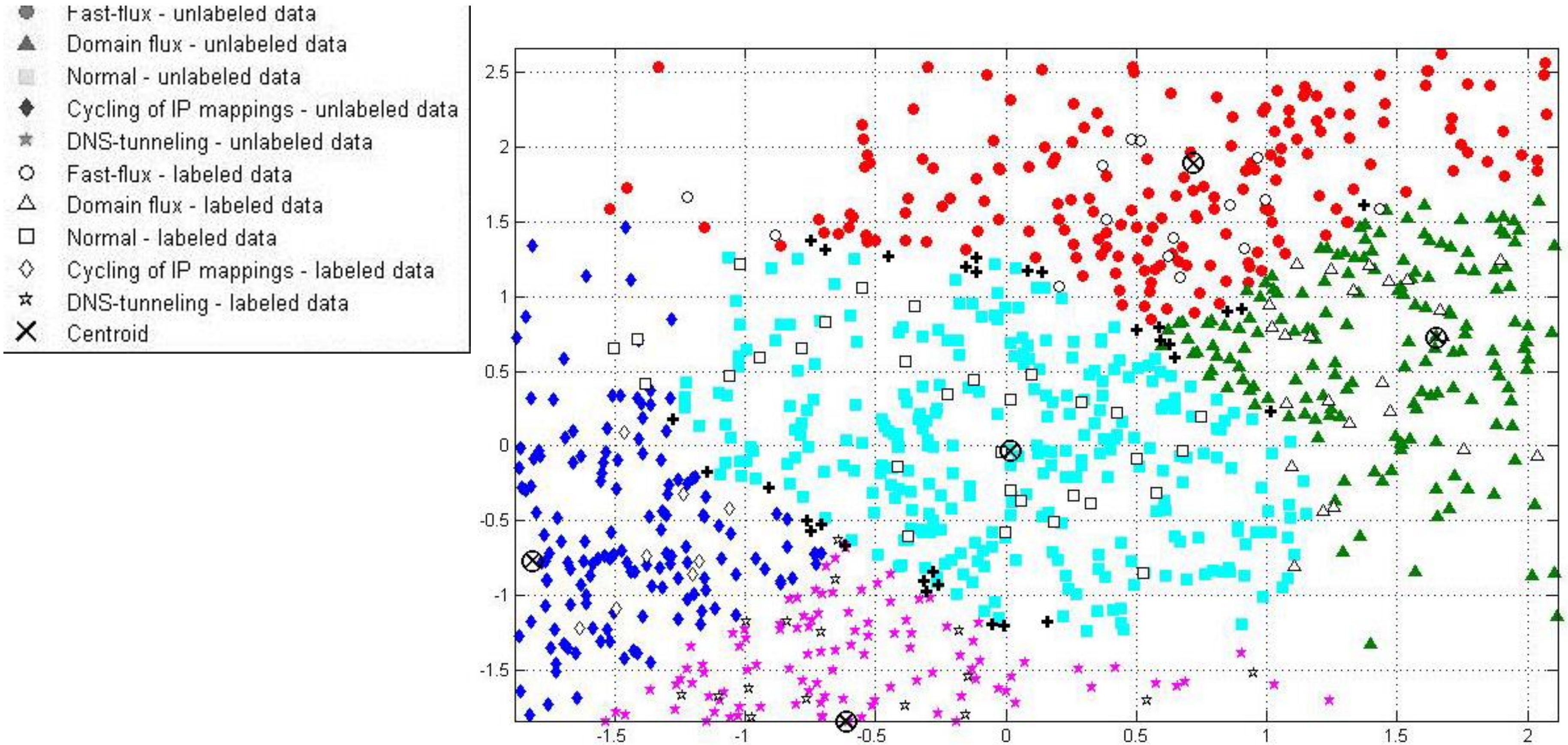
Membership of the feature vector to the fifth cluster indicates that the queries were performed to legitimate resources

As the distance between the clustering object and center of cluster the Mahalanobis distance was used

Based on the logs we can localize the bots of botnet in the network and block their actions

Experiments without Active Probing

The results of the clustering
Plane projection of the set of feature vectors of DNS-messages, which are distributed on clusters



Active DNS probing

Features obtained by means of active monitoring

S_{NS} - the average distance between the IP-addresses for the set of NS-records for the domain name;

N_{NS} - number of the NS-records in the DNS-response ;

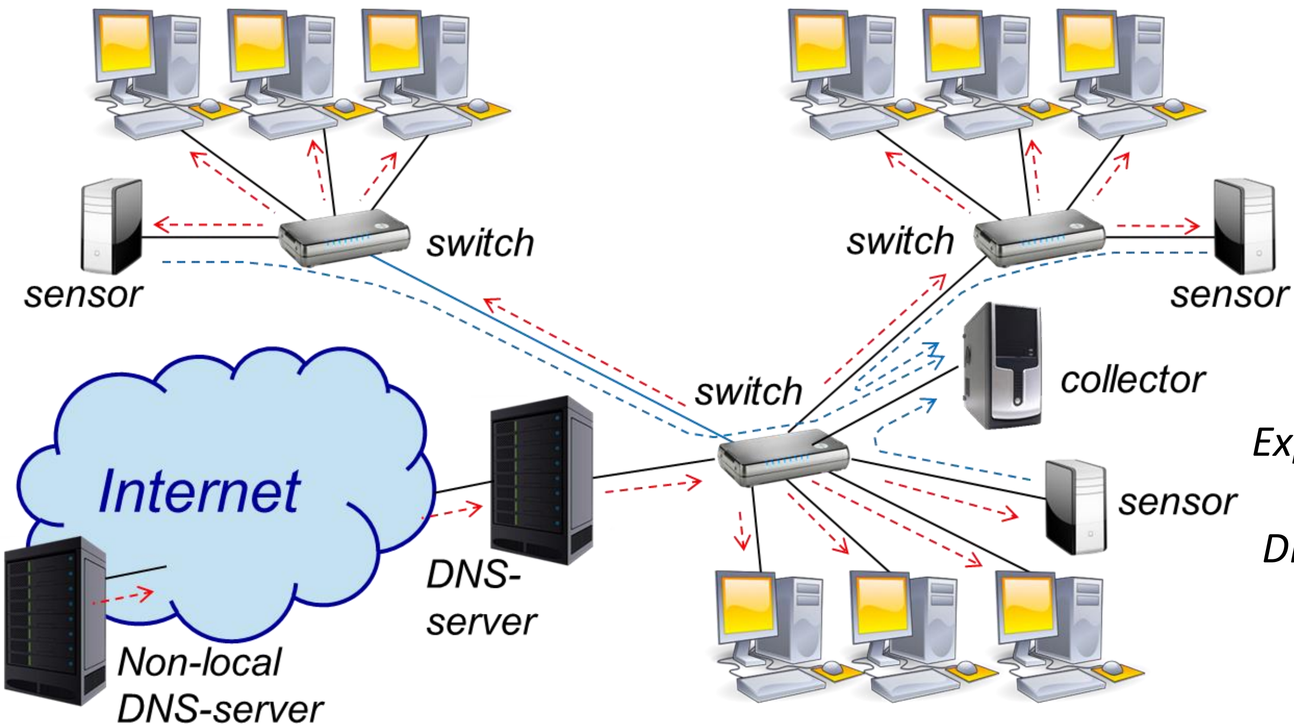
V_{retry} - value of the fields ***retry***, received from the DNS-response by a SOA-request;

N_{ASN} - amount of different numbers of autonomous systems, which include IP-addresses associated with server names;

N_{ASA} - the number of different numbers of autonomous systems (ASN), which include IP-addresses associated with the domain name

Experiments

Experimental conditions



bots: the SDBot family

Network: **100** hosts

Experiment time: **24** hours

DNS-traffic capture: tcpdump utility

Experiments

Experimental results number of queries carried out by bots, detected queries carried out by bots and false positives

Name of evasion technique	Number of queries carried by bots	Botnets detection technique based on passive monitoring, described in [9]	Improved botnets detection technique based on passive monitoring	Botnets detection technique BotGRABBER based on passive and active monitoring
		Detected queries carried out by bots/ False positives, %		
h1, cycling of IP mapping	308	299/2	301/2	301/1
h2, "domain flux"	1432	1326/3	1406/3	1406/1
h3, "fast flux"	485	389/3	425/3	425/2
h4, DNS-tunneling	144	142/0	142/0	142/0
Total	2369	2156(91%)/8	2274(96%)/8	2274(96%)/4

Summary

- We have to be well prepared for future botnets
  - Only studying current botnets is not enough
- It is an ongoing war between botnet attacks and defenses

Questions