Automated Extraction of Threat Signatures from Network Flows

Piotr Kijewski
CERT Polska/NASK

FIRST 2006 Conference, Baltimore, USA
25-30th June 2006

Agenda

- Identifying the problem
- Definition of a network threat signature
- Characteristics of a good signature
- Architecture of a signature extraction system
- Comparing by hashing – extracting signatures ”on-line”
- Extracting signatures ”off-line”
- Reduction of false alarms
- Classifying the extracted signatures
- Implementation
- Test results
- The future

Identifying the problem

- Time window between vulnerability publication and the appearance of a threat utilizing the vulnerability constantly growing shorter
- The generation of threat signatures mostly a manual process
- The process is slow and prone to errors
- Can it be automated?

Definition of a network threat signature

- A representation of a set of features of a threat
- Examples:
  - information from network packet headers
  - packet payload
  - frequency of appearance of certain ASCII characters
  - temporal characteristics of flows
- Relationship between a threat signature and an attack signature

Example of a signature

```
alert udp any any -> any 1434 (msg: „SQL Slammer"; content:
"|04 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 DC C9
B0|B|EB 0E 01 01 01 01 01 01 01|p|AE|B |01|p|AE|B|90 90 90
90 90 90 90 90|h |DC C9 B0|B|B8 01 01 01 01|1|C9 B1
18|P|E2 FD|5 |01 01 01 05|P|89 E5|
Qh.dllhel32hkernQhounthickChGetTf|B9|llQh32.dhws2_f
|B9|etQhsockf|B9|toQhsend|BE 18 10 AE|B|8D|E|D4|P|FF
16|P|8D|E|E0|P|8D|E|F0|P|FF 16|P|BE 10 10 AE|B|8B 1E 8B
03|=U |8B EC|Qt|05 BE 1C 10 AE|B|FF 16 FF D0|1|C9|QQP|81
F1 03 01 04 9B 81 F1 01 01 01 01|Q|8D|E|CC|P|8B|E|C0|P|FF
16|j|11| j|02|j|02 FF D0|P|8D|E|C4|P|8B|E|C0|P|FF 16 89 C6
09 DB 81 F3|<a|D9 FF 8B|E|B4 8D 0C|@|8D 14 88 C1 E2 04 01
C2 C1 E2 08| )|C2 8D 04 90 01 D8 89|E|B4|j|10
8D|E|B0|P1|C9|Qf|81 F1|x|01|Q|8D|E|03|P|8B|E|AC|P|FF D6
EB|"; )
```

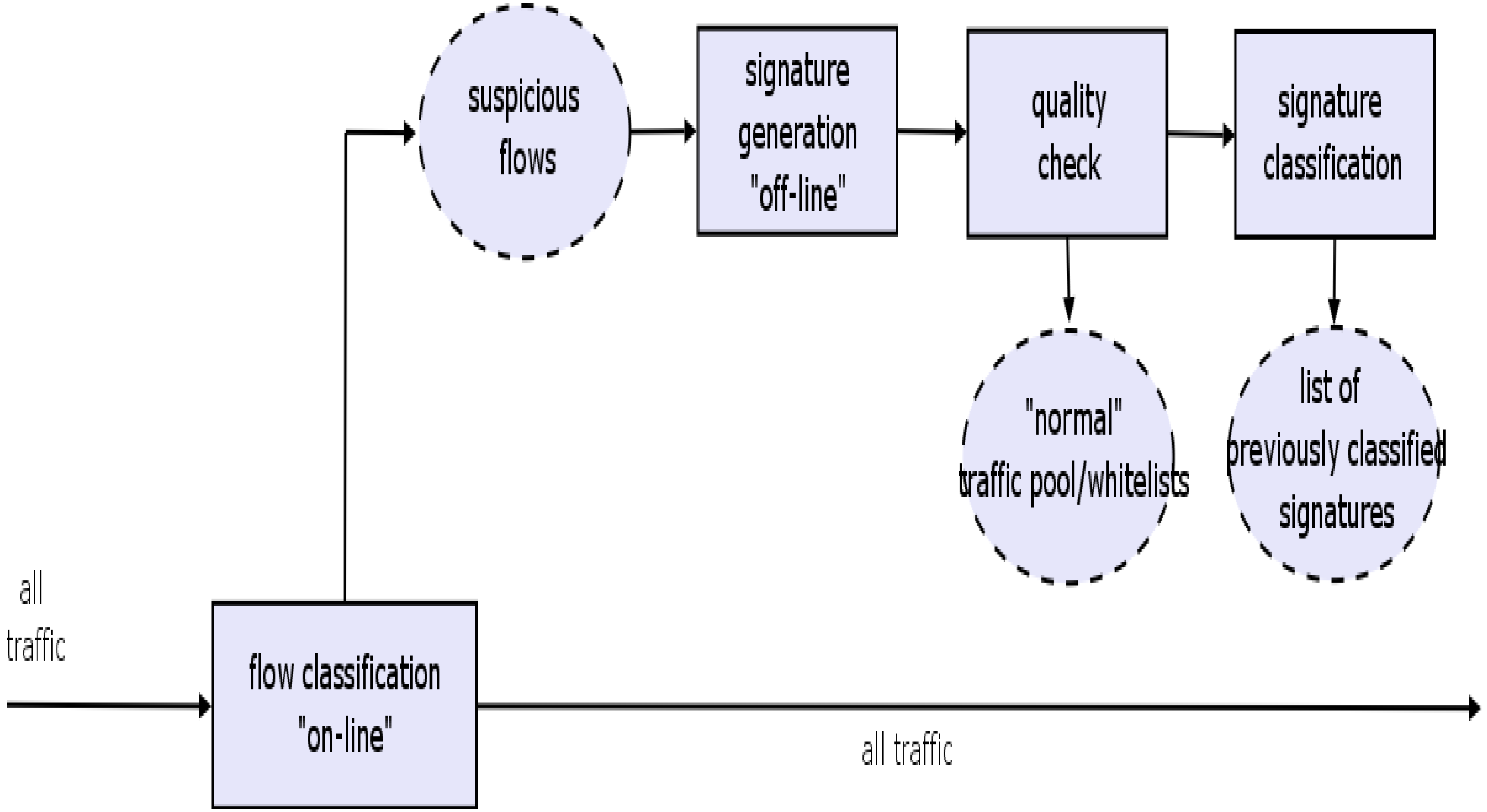
Characteristics of a good signature (1/2)

- Detects the attack
- Low false alarm rate
- Can be generated quickly
- Independent of application level protocols
- Can be used in existing IDS/IPS systems

Characteristics of a good signature (2/2)

- Exploit vs vulnerability
- Usage of the ”de facto” standard: signatures representing a sequence of bytes that characterize a threat
- Operating at a network level allows for the quick deployment of the signature until hosts patched (important from an early warning point of view)

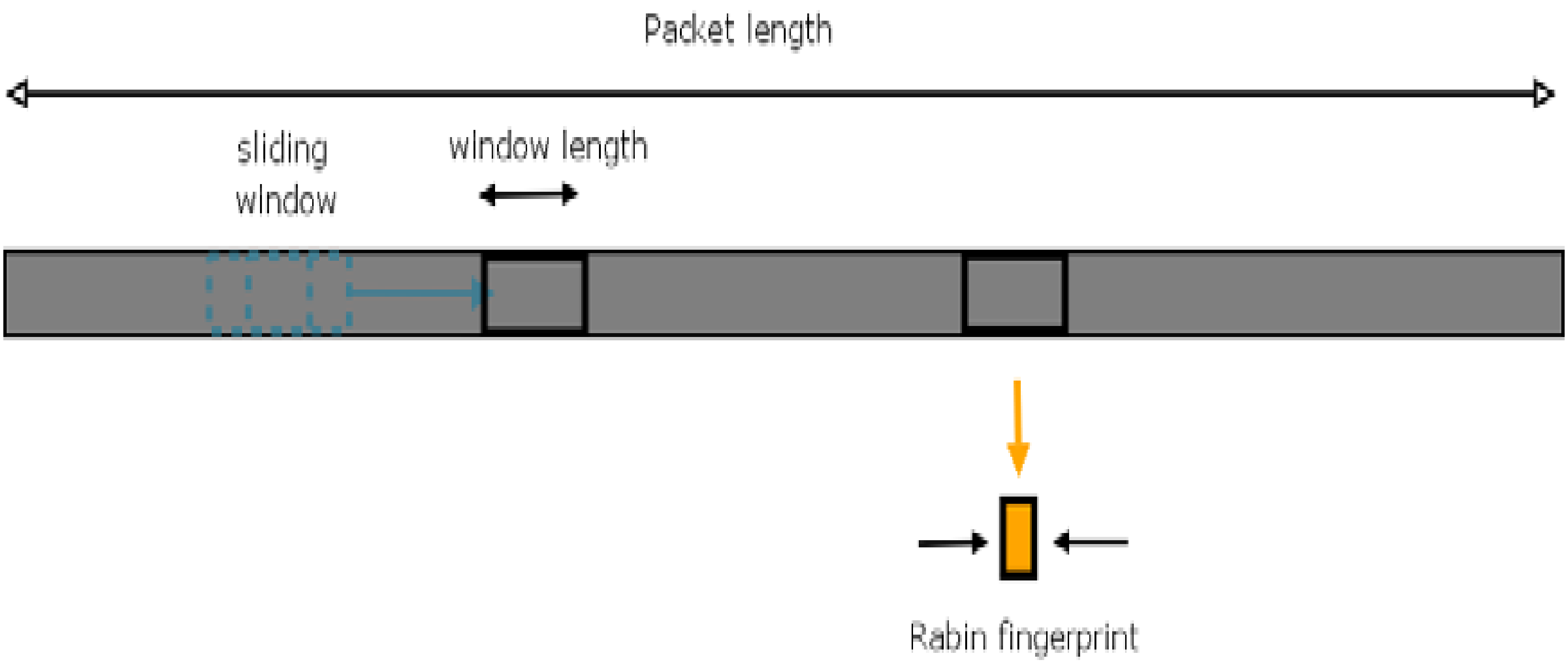
Architecture of a signature extraction system



Comparing by hashing (1/6)

- Simplest way to identify attacks – comparing and cataloging packets by cryptographic hashes
- MD5 hash = attack signature
- In practice works only in a honeynet environment (example: Internet Motion Sensor project)
- Any modification to packet -> new hash
- Cannot identify the sequence of bytes that make up the essence of the attack

Comparing by hashing – sliding window across a packet (2/6)



Comparing by hashing (3/6)

- Sliding window mechanism: better identification of the constant in the packet
- … but many hashes formed (if s is the packet size in bytes, β is the window length, the amount of hashes equals $s - \beta + 1$)

Comparing by hashing (4/6)

- Rabin fingerprints as a hash function (basis of the Rabin-Karp string searching algorithm)
- Calculate the hash of a window shifted by one character based on the calculation of the previous window
- Rabin hash = attack signature
- Method may be applied both to production networks and honeynets

Comparing by hashing (5/6)

- To improve efficiency:
  - Sample based on a bitmask (for example sample only hashes that have four least significant bits set to zero)
  - Compute flows only in one direction (for example only from a client to a server)

Comparing by hashing (6/6)

- Sampling introduces the risk of missing an attack or not identifying the most interesting sequence
- Problems with window length: the smaller the window size the higher the probability of detecting the attack but also the higher the chance of a false alarm
- Polymorphism: polymorphic attacks may be missed as they may not contain long enough sequences to fill a window
- Efficiency

Generating signatures ”off-line” (1/3)

- More complex algorithms may be utilized in the ”off-line” mode
- Example: Longest Common Substring algorithm (LCS)
- Our proposal: use Rabin windows to initially classify flows (detected anomalies), the actual generation of signatures transferred to other algorithms (like LCS)

Generating signatures ”off-line” (2/3)

- Define grouping rules:
  - Completed flows are periodically grouped based on their Rabin similarity (for example, group all expired flows to the same destination port that contain 30% of the same fingerprints)
  - Heuristics: for every group, check the amount of unique sources in a given period. If a threshold is reached, the group is sent for further analysis ”off-line”
  - An external process computes LCS on every submitted group

Generating signatures ”off-line” (3/3)

- Potential to detect polymorphic attacks (if in a honeynet environment)
- The grouping rule checks the groups that are composed of only one flow and are sent for off-line analysis
- Algorithms other than LCS (example, Smith-Waterman) can analyse all the submitted groups together – there should exist small disjoint common sequences that have to remain constant for the exploit to function

Reduction of false alarms

- The longest common substring may not be the best substring
- The created signature should be compared to a list of benign signatures (whitelists)
- A pool of normal flows may be kept for comparison
- Vetting by an operator

Classification of signatures (1/2)

- It is important to review a new event on the network
- A generated signature may be compared to previously classified ones
- There may be very many signatures, it is useful to compare with a certain signature class
- Need to define a similarity function

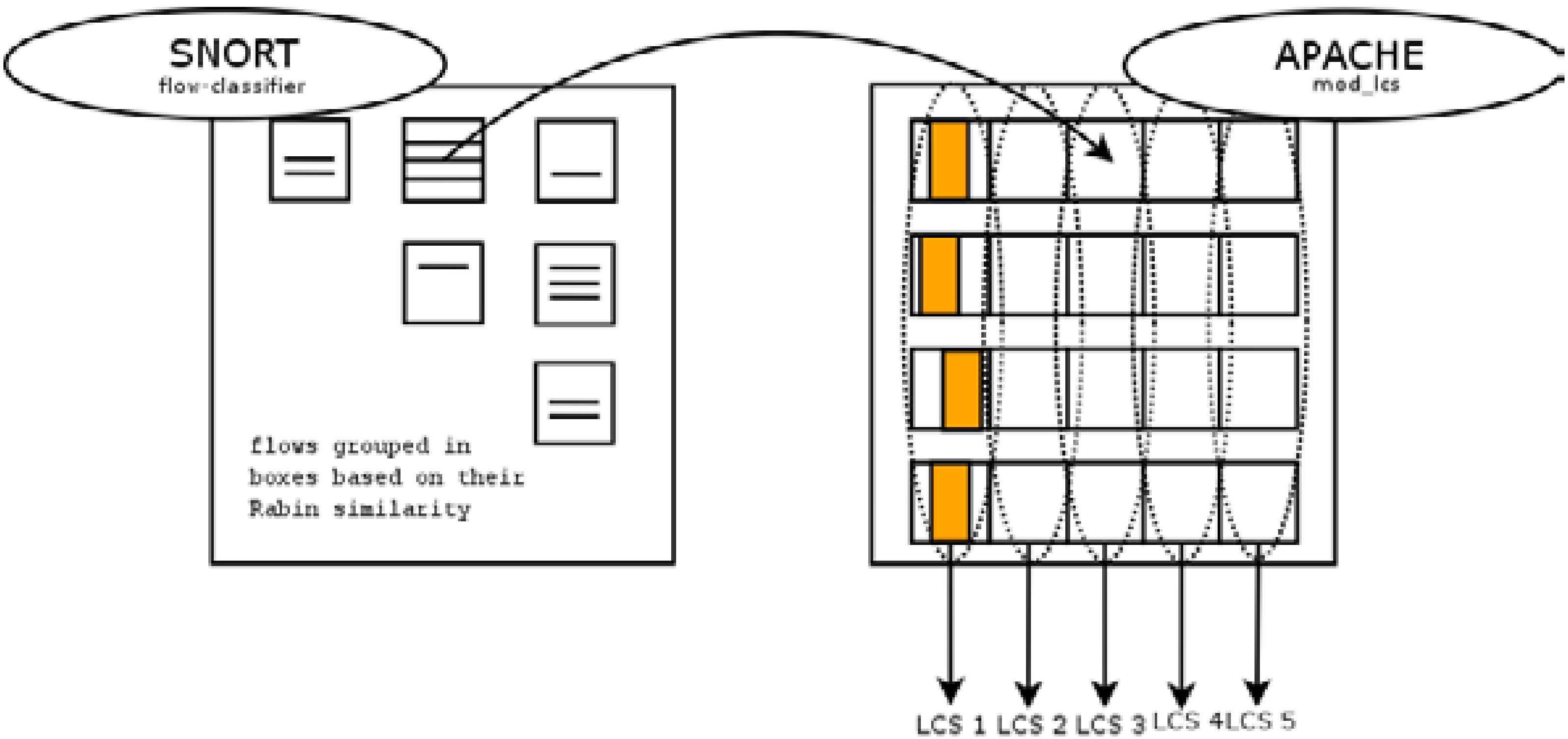
Classification of signatures (2/2)

- Levenshtein distance between strings as a distance metric
- Use clustering algorithms (simplified *dbscan*)
- Signatures are periodically clustered and manually classified (with support from Bleeding Snort rules)
- For efficiency reasons, long repetitions of characters (such as NOOPs) are packed to a certain maximum length
- Dynamic radius of a cluster based on the length of the core member in order to allow for better clustering of both short and long signatures

Implementation (1/2)

- Base software: *snort* and *Apache2*
- Rabin fingerprints implemented as *snort* plugin called *flow-rabin* on top of the standard *flow* and *stream4* plugins
- The *flow-rabin* plugin is the basis for the *flow-classifier* plugin, which implements various preliminary grouping rules
- When a threat cluster is detected, the cluster is transferred to the *mod_lcs Apache* module for LCS signature extraction
- Communication between *snort* and *mod_lcs* TCP based
- External clustering process (implemented in PHP5)

Implementation (2/2)



Test results (1/2)

- 24 hours monitoring of 5 /26 subnets (honeyd/nepenthes)
- Total 775 716 packets collected
- Grouping rules: 3 distinct sources with flows that are 30% similar in a space of 5 minutes
- 408 LCS signatures generated (LCS generated per packet)
- 63 clusters formed
- 63 signatures computed (one per cluster)
- 7 signatures found to generate false positives (based on a trace of ”normal” traffic)
- 21 further signatures dropped (vetting process)

Test results (2/2)

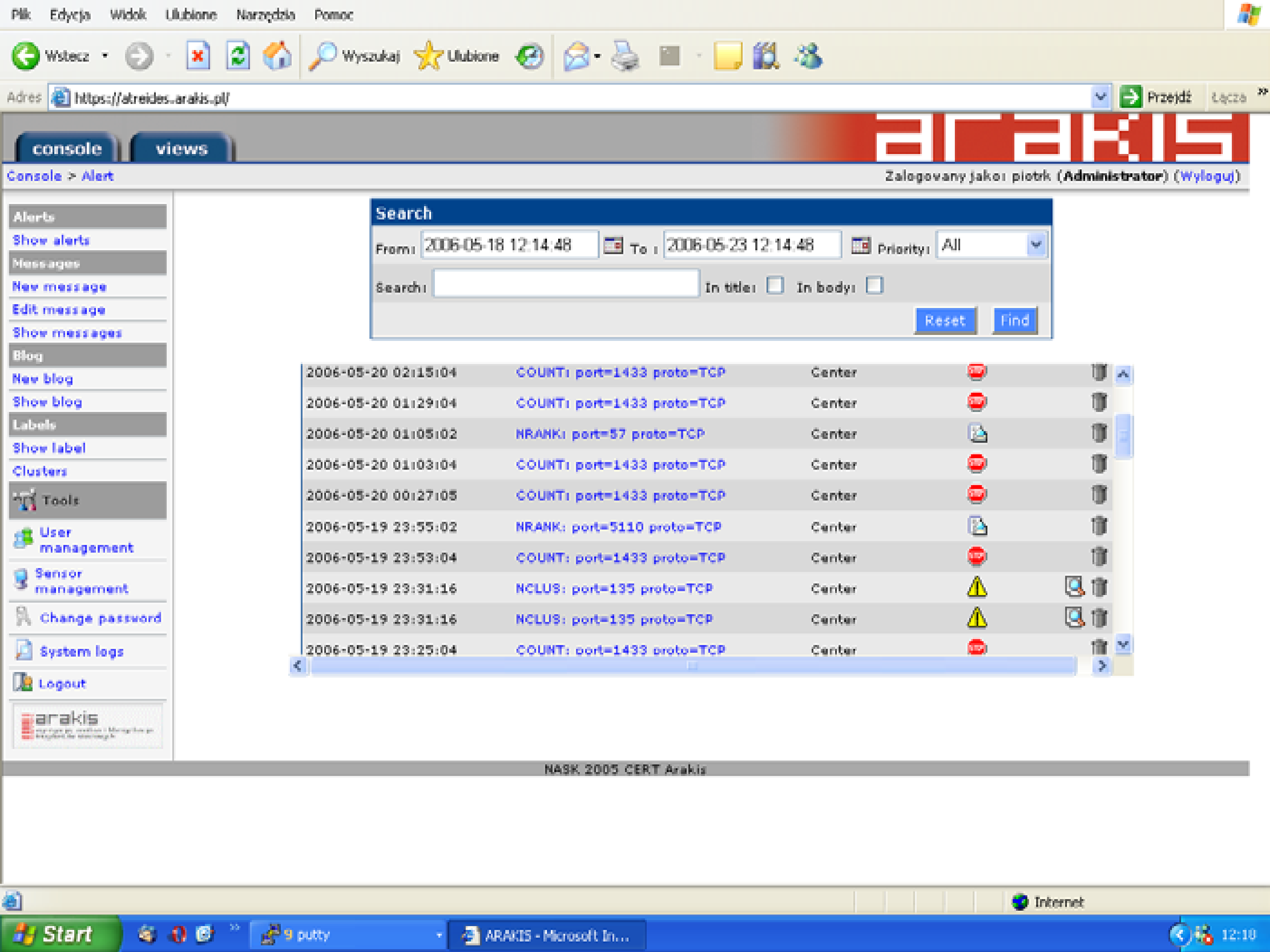
- The 35 remaining clusters:
  - LSA exploit (port 445/TCP) – 10 clusters
  - ASN1 exploit (port 445/TCP, port 139/TCP) – 8 clusters
  - Winpopup spam (ports 1026-1029 UDP) – 5 clusters
  - RPC DCOM (port 135/TCP, 1025/TCP) – 4 clusters
  - Shellcode x86 NOOP (port 445/TCP) – 2 clusters
  - Port 1026/UDP unknown [1] – 2 clusters
  - SQL Slammer (port 1434/UDP) – 1 cluster
  - Port 1433/TCP unknown [2] – 1 cluster
  - NetBIOS query (port 139/TCP) – 1 cluster
  - HTTP OPTIONS query (port 80/TCP) – 1 cluster

[1] Probably related to Winpopup spam

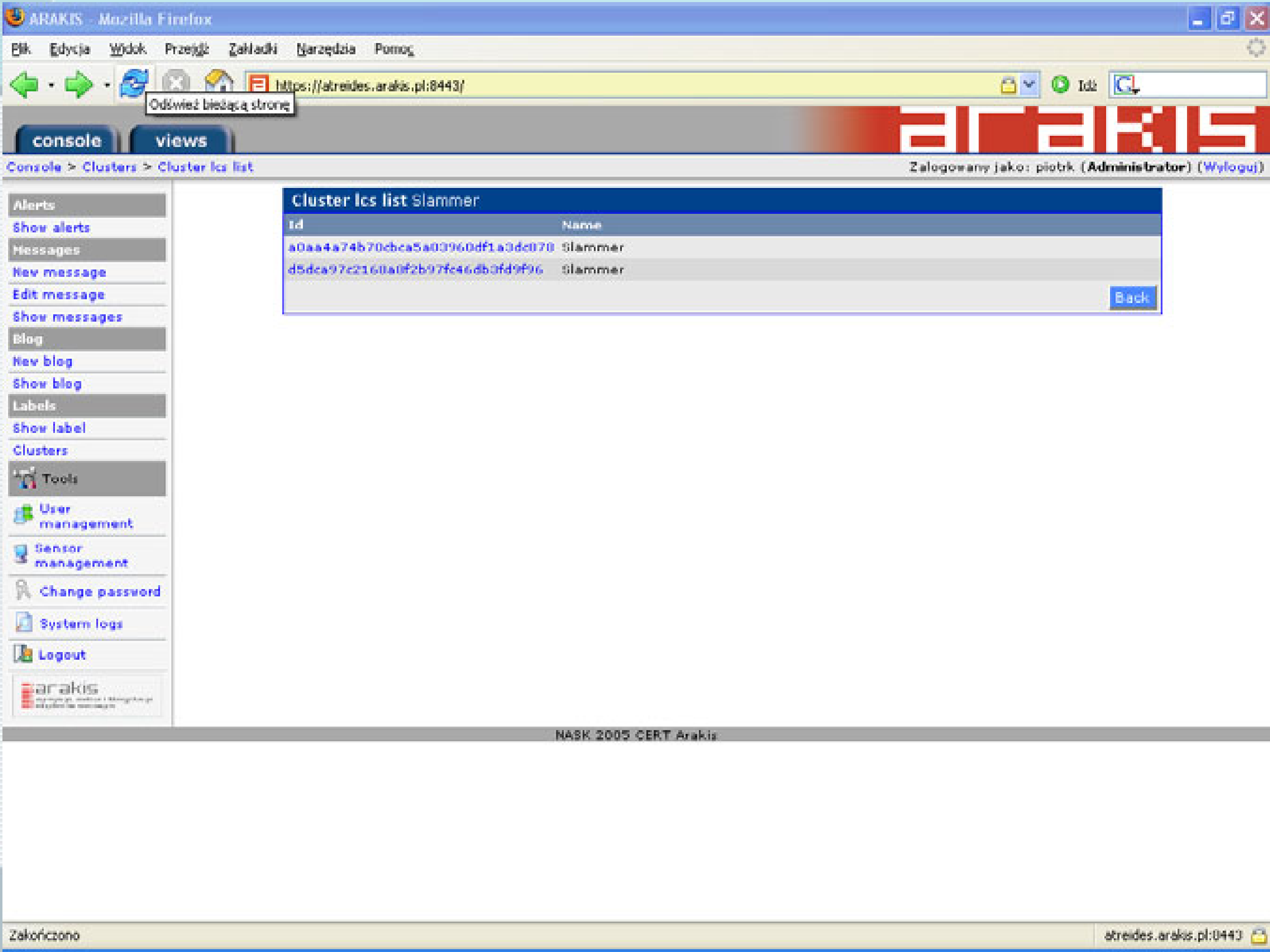
[2] A large amount of short packets to the standard MS SQL Server port - possibly a brute force attempt. It was not identified by any Snort rules.

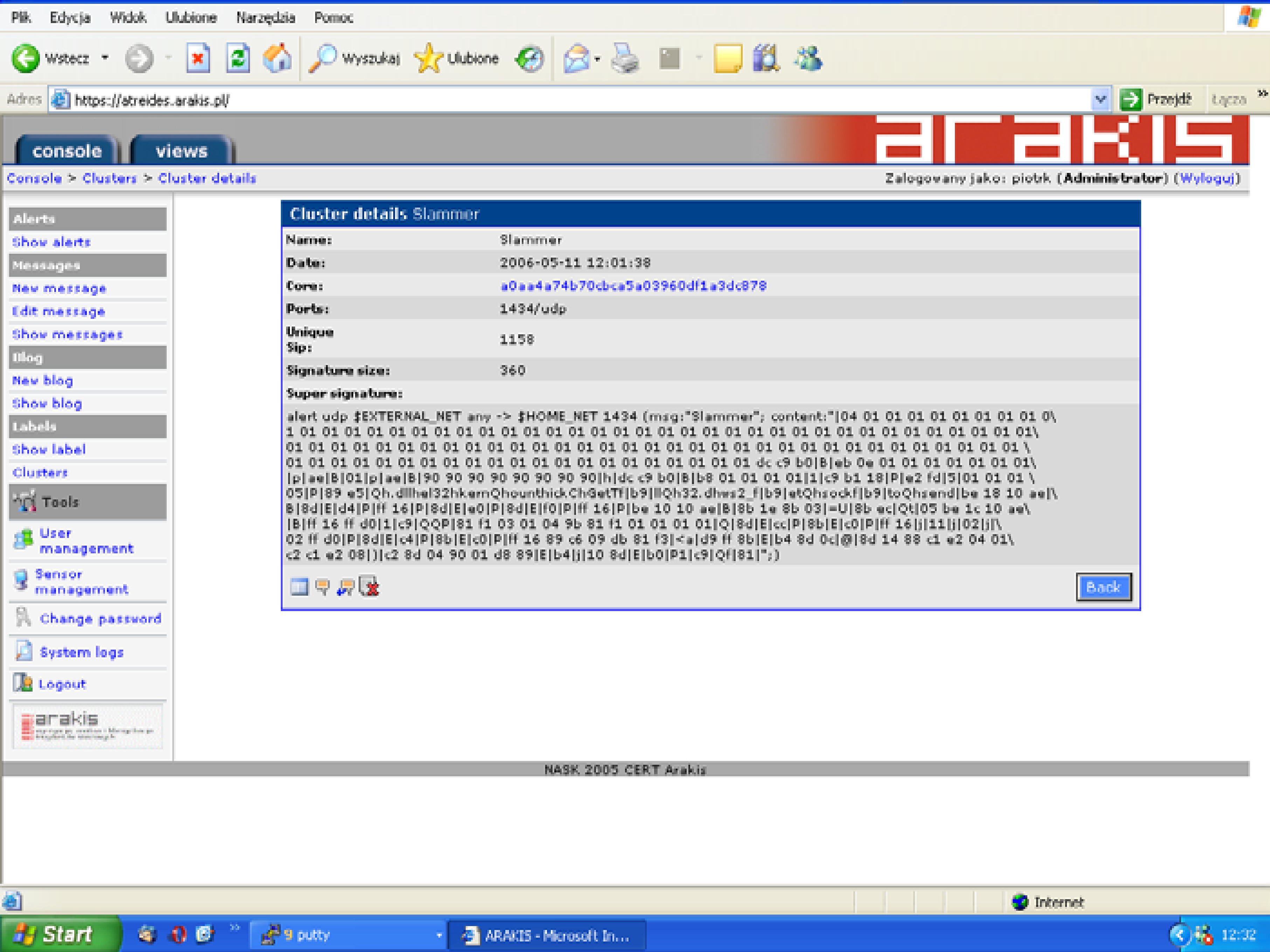
Future

- Current implementation in testing phase
- Application in a an environment other than honeynet
- Application of new algorithms for detection of anomalies and classification of flows
- Implementation of ”off-line” algorithms other than LCS
- Development of methods for signature management


Plik Edycja Widok Ulubione Narzędzia Pomoc
Wstecz
Wyszukaj
Ulubione
Adres https://atreides.arakis.pl/
Przejdź
Łącza
console
views
Console > Alert
Zalogowany jako: piotrk (Administrator) (Wyloguj)
Alerts
Show alerts
Messages
New message
Edit message
Show messages
Blog
New blog
Show blog
Labels
Show label
Clusters
Tools
User management
Sensor management
Change password
System logs
Logout
arakis
Search
From: 2006-05-18 12:14:48
To : 2006-05-23 12:14:48
Priority: All
Search:
In title:
In body:
Reset
Find
2006-05-20 02:15:04 COUNT: port=1433 proto=TCP Center
2006-05-20 01:29:04 COUNT: port=1433 proto=TCP Center
2006-05-20 01:05:02 NRANK: port=57 proto=TCP Center
2006-05-20 01:03:04 COUNT: port=1433 proto=TCP Center
2006-05-20 00:27:05 COUNT: port=1433 proto=TCP Center
2006-05-19 23:55:02 NRANK: port=5110 proto=TCP Center
2006-05-19 23:53:04 COUNT: port=1433 proto=TCP Center
2006-05-19 23:31:16 NCLUS: port=135 proto=TCP Center
2006-05-19 23:31:16 NCLUS: port=135 proto=TCP Center
2006-05-19 23:25:04 COUNT: port=1433 proto=TCP Center
NASK 2005 CERT Arakis
Internet
Start
9 putty
ARAKIS - Microsoft In...
12:10

Date	Name
2006-05-11 12:01:26	SMB initiation
2006-05-11 12:01:35	SMB #1
2006-05-11 12:01:38	Slammer
2006-05-11 12:01:59	SMB #2
2006-05-11 12:02:08	SMB #3
2006-05-11 12:02:08	SMB #4
2006-05-11 12:02:08	SMB #5
2006-05-11 12:02:11	0x31 LSA
2006-05-11 12:02:16	SMB unknown #1
2006-05-11 12:02:18	SMB #6
2006-05-11 12:02:20	SMB JUNK #1
2006-05-11 12:02:20	SMB unknown #2
2006-05-11 12:02:21	SMB unknown #3
2006-05-11 12:02:21	SMB #7
2006-05-11 12:02:24	0x90
2006-05-11 12:02:30	SMB unknown #4
2006-05-11 12:02:30	SMB #8
2006-05-11 12:02:43	0x90 0x31 LSA
2006-05-11 12:07:09	SMB #9
2006-05-11 12:07:12	Winpop Registry Cleaner Spam
2006-05-11 12:07:17	RPC DCOM #1


ARAKIS - Mozilla Firefox
Plik Edycja Widok Przejdź Zakładki Narzędzia Pomoc
https://atreides.arakis.pl:8443/
Idź
Odśwież bieżącą stronę
console
views
arakis
Console > Clusters > Cluster lcs list
Zalogowany jako: piotrk (Administrator) (Wyloguj)
Alerts
Show alerts
Messages
New message
Edit message
Show messages
Blog
New blog
Show blog
Labels
Show label
Clusters
Tools
User management
Sensor management
Change password
System logs
Logout
arakis
Cluster lcs list Slammer
Id
Name
a0aa4a74b70cbca5a03960df1a3dc070
Slammer
d5dca97c2160a0f2b97fc46db3fd9f96
Slammer
Back
NASK 2005 CERT Arakis
Zakończono
atreides.arakis.pl:8443


Plik Edycja Widok Ulubione Narzędzia Pomoc
Wstecz
Wyszukaj
Ulubione
Adres https://atreides.arakis.pl/
Przejdź
Łącza
console
views
Console > Clusters > Cluster details
Zalogowany jako: piotrk (Administrator) (Wyloguj)
Alerts
Show alerts
Messages
New message
Edit message
Show messages
Blog
New blog
Show blog
Labels
Show label
Clusters
Tools
User management
Sensor management
Change password
System logs
Logout
arakis
Cluster details Slammer
Name: Slammer
Date: 2006-05-11 12:01:38
Core: a0aa4a74b70cbca5a03960df1a3dc878
Ports: 1434/udp
Unique Sip: 1158
Signature size: 360
Super signature:
alert udp $EXTERNAL_NET any -> $HOME_NET 1434 (msg:"Slammer"; content:"|04 01 01 01 01 01 01 01 01 0\
1 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01\
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 \
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 dc c9 b0|B|eb 0e 01 01 01 01 01 01 01\
|p|ae|B|01|p|ae|B|90 90 90 90 90 90 90 90|h|dc c9 b0|B|b8 01 01 01 01|1|c9 b1 18|P|e2 fd|5|01 01 01 \
05|P|89 e5|Qh.dllhel32hkernQhounthickChGetTf|b9|llQh32.dhws2_f|b9|etQhsockf|b9|toQhsend|be 18 10 ae|\
B|8d|E|d4|P|ff 16|P|8d|E|e0|P|8d|E|f0|P|ff 16|P|be 10 10 ae|B|8b 1e 8b 03|=U|8b ec|Qt|05 be 1c 10 ae\
|B|ff 16 ff d0|1|c9|QQP|81 f1 03 01 04 9b 81 f1 01 01 01 01|Q|8d|E|cc|P|8b|E|c0|P|ff 16|j|11|j|02|j|\
02 ff d0|P|8d|E|c4|P|8b|E|c0|P|ff 16 89 c6 09 db 81 f3|<a|d9 ff 8b|E|b4 8d 0c|@|8d 14 88 c1 e2 04 01\
c2 c1 e2 08|)|c2 8d 04 90 01 d8 89|E|b4|j|10 8d|E|b0|P1|c9|Qf|81|";)
Back
NASK 2005 CERT Arakis
Internet
Start
9 putty
ARAKIS - Microsoft In...
12:32

Plik Edycja Widok Ulubione Narzędzia Pomoc

Wstecz Wyszukaj Ulubione

Adres https://atreides.arakis.pl/ Przejdź Łącza

console views

Clusters Table

Zalogowany jako: piotrk (**Administrator**) (Wyloguj)

View
- Graphs
  - Global
    - Ranking TOP10 (fv)
    - Ranking TOP10 (hn)
    - Any port
    - AV
  - Sensors
- Tables
  - Global
    - BS Tables
    - AV Tables
    - Flow Tables
    - AS Tables
    - OS Tables
    - LCS Tables
    - Cluster Tables
    - Portsig Tables
  - Sensors
- IP search
- TCPDump

Time period

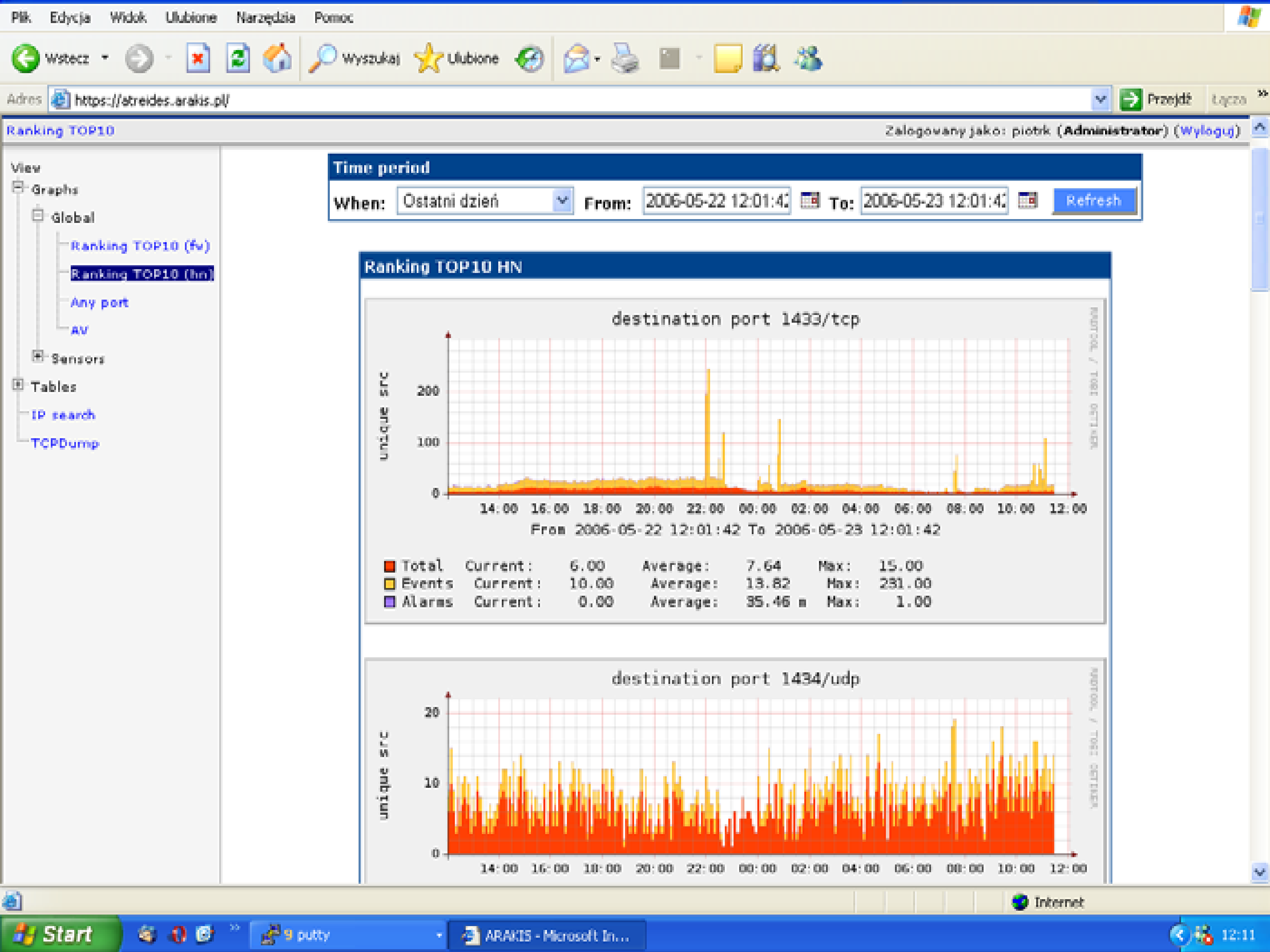
When: Ostatni dzień **From:** 2006-05-22 12:00:0[illegible] **To:** 2006-05-23 12:00:0[illegible] Refresh

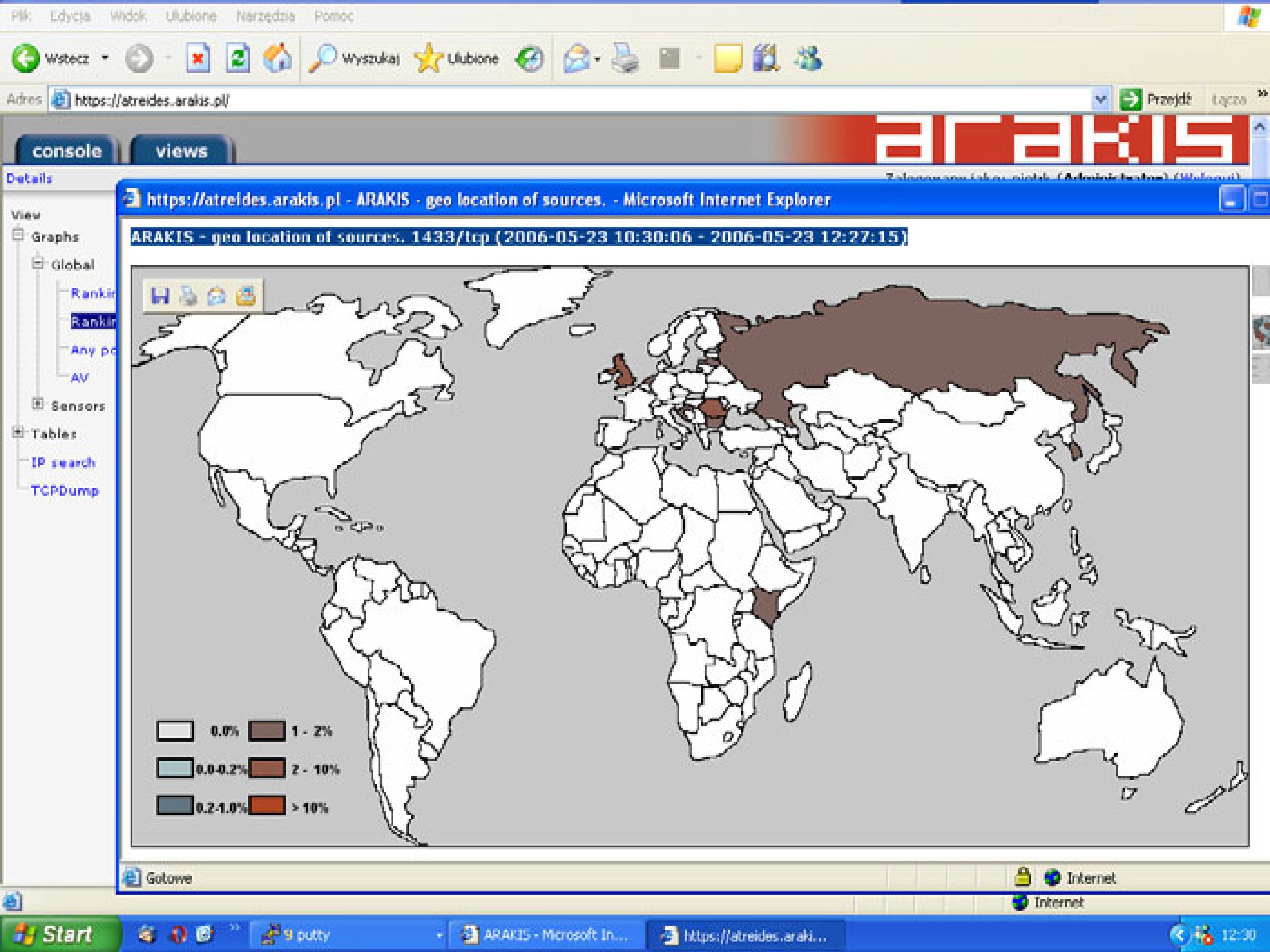
Clusters Table

Cluster name	Flows	% of flows	Port	Protocol	% of unique src
1433 unknown	16898	27.18	1433	tcp	6.63
Slammer	2840	4.58	1434	udp	25.54
SMB initiation	2034	3.27	445	tcp	42.01
0x90 0x31 LSA	1133	1.82	445	tcp	18.92
0x31 LSA	1100	1.77	445	tcp	18.92
SMB unknown #1	1081	1.74	445	tcp	19.01
SMB #1	908	1.46	445	tcp	24.85
0x90	903	1.45	445	tcp	19.70
SMB #2	845	1.36	445	tcp	23.37
SMB #4	786	1.26	445	tcp	22.14
SMB #3	780	1.25	445	tcp	22.14
SMB #5	668	1.07	445	tcp	19.35
SMB JUNK #1	394	0.63	445	tcp	12.55
0x90 LSA	287	0.46	445	tcp	0.28
SMB #6	252	0.41	445	tcp	9.94
0x42 0x43 RPC DCOM	238	0.38	139	tcp	0.28
LSA Korgo x.exe	231	0.37	445	tcp	6.63
0x43 0x44 ASN1	224	0.36	139	tcp	0.11
SMB unknown #2	222	0.36	445	tcp	9.59
0x44 RPC	211	0.34	139	tcp	0.09
0x90 LSA #2	200	0.32	445	tcp	4.62
[no name]	188	0.30	139	tcp	7.59
[illegible]	[illegible]	[illegible]	[illegible]	udp	[illegible]

Gotowe Internet

Start 9 putty ARAKIS - Microsoft In... 12:20


Plik Edycja Widok Ulubione Narzędzia Pomoc
Wstecz
Wyszukaj
Ulubione
Adres https://atreides.arakis.pl/
Przejdź
Łącza
Ranking TOP10
Zalogowany jako: piotrk (Administrator) (Wyloguj)
View
Graphs
Global
Ranking TOP10 (fw)
Ranking TOP10 (hn)
Any port
AV
Sensors
Tables
IP search
TCPDump
Time period
When: Ostatni dzień
From: 2006-05-22 12:01:4
To: 2006-05-23 12:01:4
Refresh
Ranking TOP10 HN
destination port 1433/tcp
unique src
200
100
0
14:00 16:00 18:00 20:00 22:00 00:00 02:00 04:00 06:00 08:00 10:00 12:00
From 2006-05-22 12:01:42 To 2006-05-23 12:01:42
Total Current: 6.00 Average: 7.64 Max: 15.00
Events Current: 10.00 Average: 13.82 Max: 231.00
Alarms Current: 0.00 Average: 35.46 m Max: 1.00
destination port 1434/udp
unique src
20
10
0
14:00 16:00 18:00 20:00 22:00 00:00 02:00 04:00 06:00 08:00 10:00 12:00
Internet
Start
9 putty
ARAKIS - Microsoft In...
12:11


Plik Edycja Widok Ulubione Narzędzia Pomoc
Wstecz
Wyszukaj
Ulubione
Adres
https://atreides.arakis.pl/
Przejdź
Łącza
arakis
console
views
Details
View
Graphs
Global
Sensors
Tables
IP search
TCPDump
https://atreides.arakis.pl - ARAKIS - geo location of sources. - Microsoft Internet Explorer
ARAKIS - geo location of sources. 1433/tcp (2006-05-23 10:30:06 - 2006-05-23 12:27:15)
0.0%
1 - 2%
0.0-0.2%
2 - 10%
0.2-1.0%
> 10%
Gotowe
Internet
Internet
Start
9 putty
ARAKIS - Microsoft In...
https://atreides.araki...
12:30