IPSCORE: AN INTRUSION DETECTION SYSTEM (IDS) BASED ON STATISTICAL ACCESS PATTERN ANALYSIS

Onur Kocak
Bilkent University
onur.kocak@bilkent.edu.tr

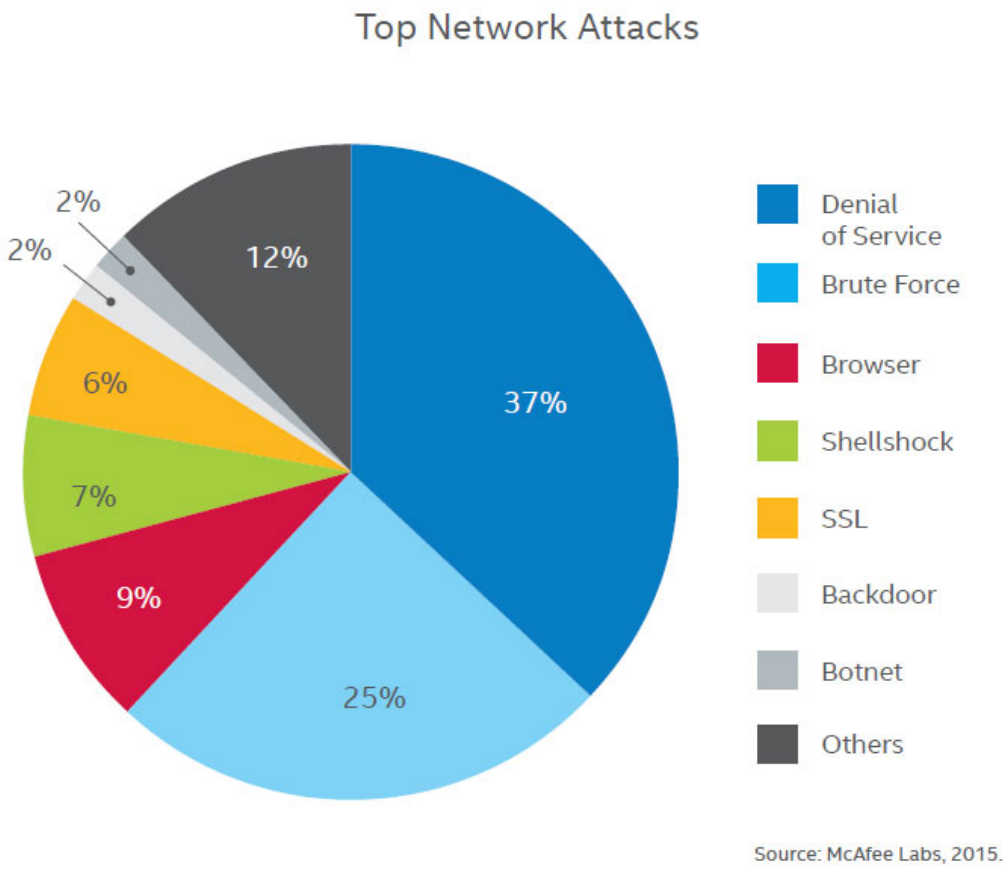
Omer Faruk Aktulum
Bilkent University
omer.aktulum@bilkent.edu.tr

Outline

- Attacks, Attack Types
- Prevention Approaches
- Literature Review
- Motivation
- Proposed Method
- Experiments
- Next Steps
- References
- Questions

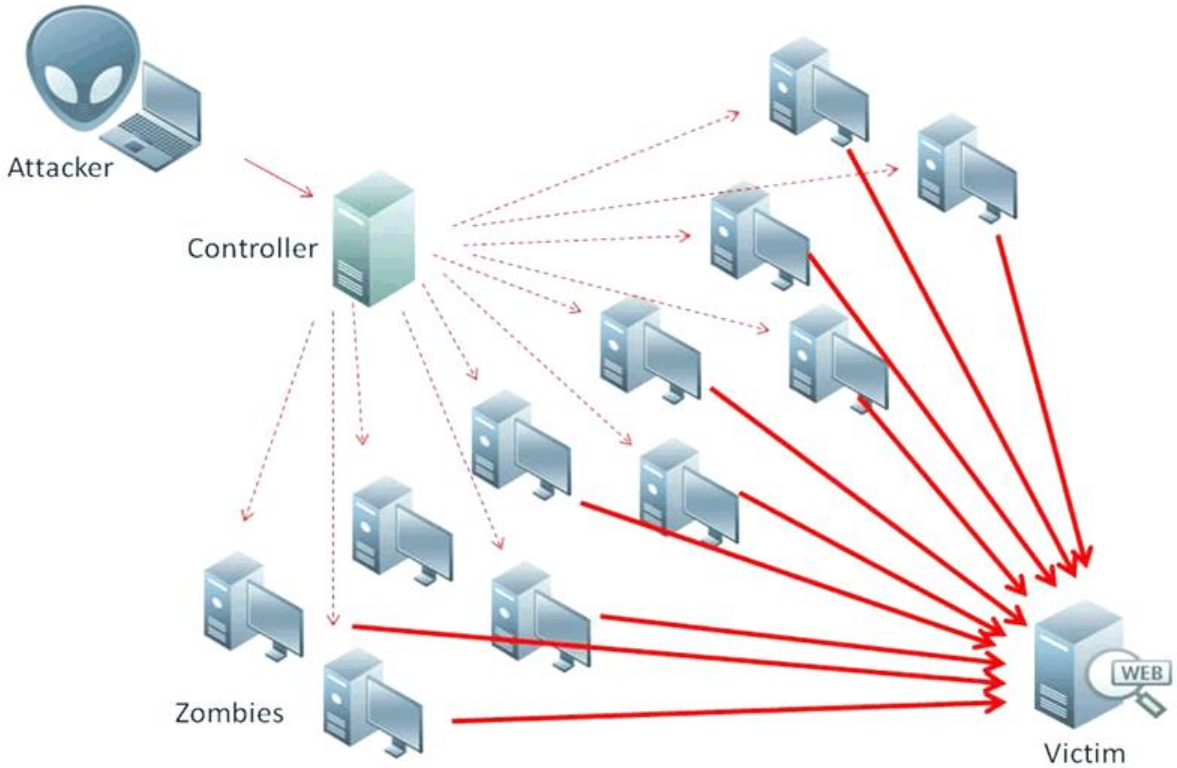
Network Attacks

- Most common attack model: DoS & Brute Force



DoS & Distributed DoS

- A denial of service (DoS) attack attempts to make a resource, such as a web server, unavailable to users.



Types of DoS

1. **Network Layer Attacks**

- TCP, UDP and ICMP Floods (SYN Flood, Ping of Death, etc.)

2. **Application Layer Attacks**

- HTTP GET & POST Flood

Zombie machines attack the victim server through legitimate packets such that packets have legitimate format and are sent through normal TCP connections.

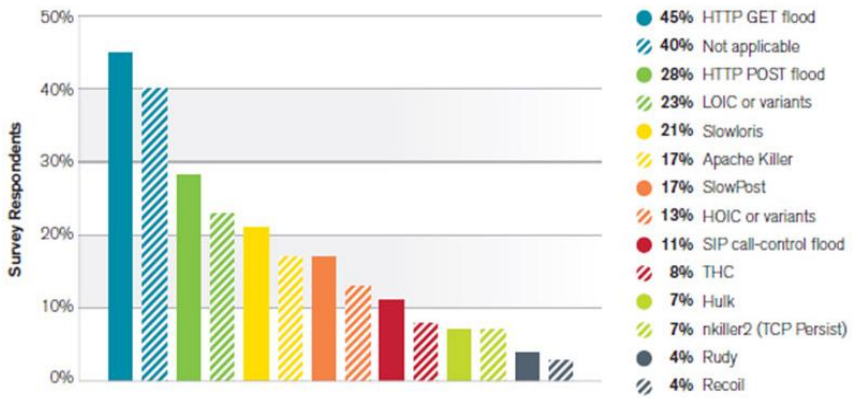


Figure 23 Source: Arbor Networks, Inc.

Preventing DoS Attacks

On Network Layer

- Analyzing # of TCP& UDP Packets in a Statistical Manner
- Analyzing Behavioral Status of Network (Resource Usages, etc.)

On Application Layer

- Analyzing payload of packet (DPI)
- Client-side techniques (CAPTCHAs)

Literature Review

- CAPTCHA Puzzles [1, 2]

A CAPTCHA puzzle is a challenge-response test used in web applications as an attempt to ensure that the response is generated by a human not by a machine.

Weaknesses:

- Several image recognition techniques to break CAPTCHA [3]
- **Labor attacks:** There are free/cheap 3rd party human labor to break CAPTCHAs [4, 5]

- Yatagai's method [6]

Measure session durations and browsing order

Weaknesses:

Vulnerable to advanced DDoS attack scenarios with randomized distribution

Literature Review

- ConnectionScore [8]

Statistical technique to protect from most common application layer DDoS attacks such as HTTP flood. It measures request times, request rates, download rates, browsing behaviour (link depth).

Weaknesses:

24% of the malicious connections do not get negative scores

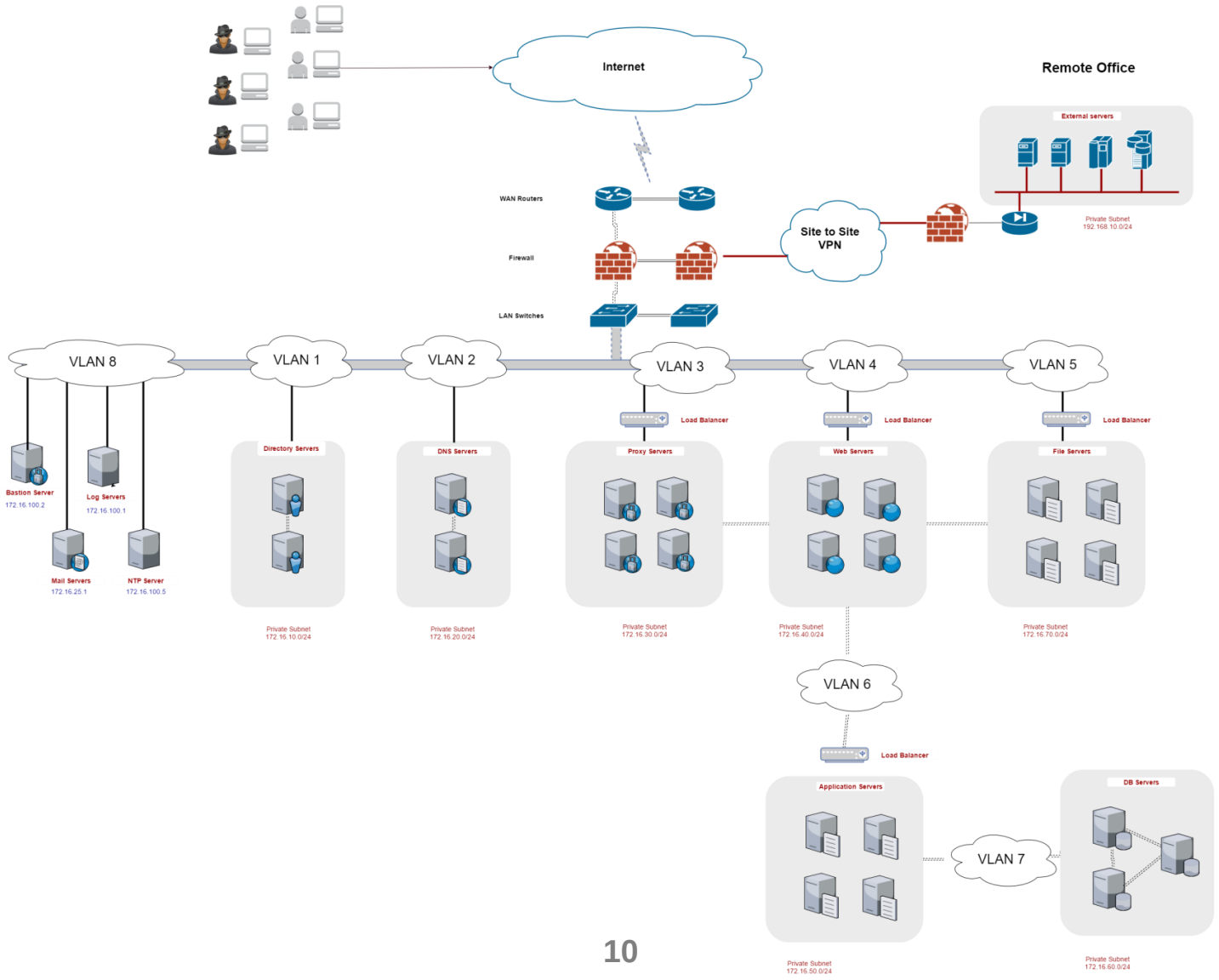
Motivation

- Develop a new statistical method somewhat similar but not limited to Connection Score to prevent application layer DDoS & brute force attacks by analyzing access logs in historic and volumetric distribution.

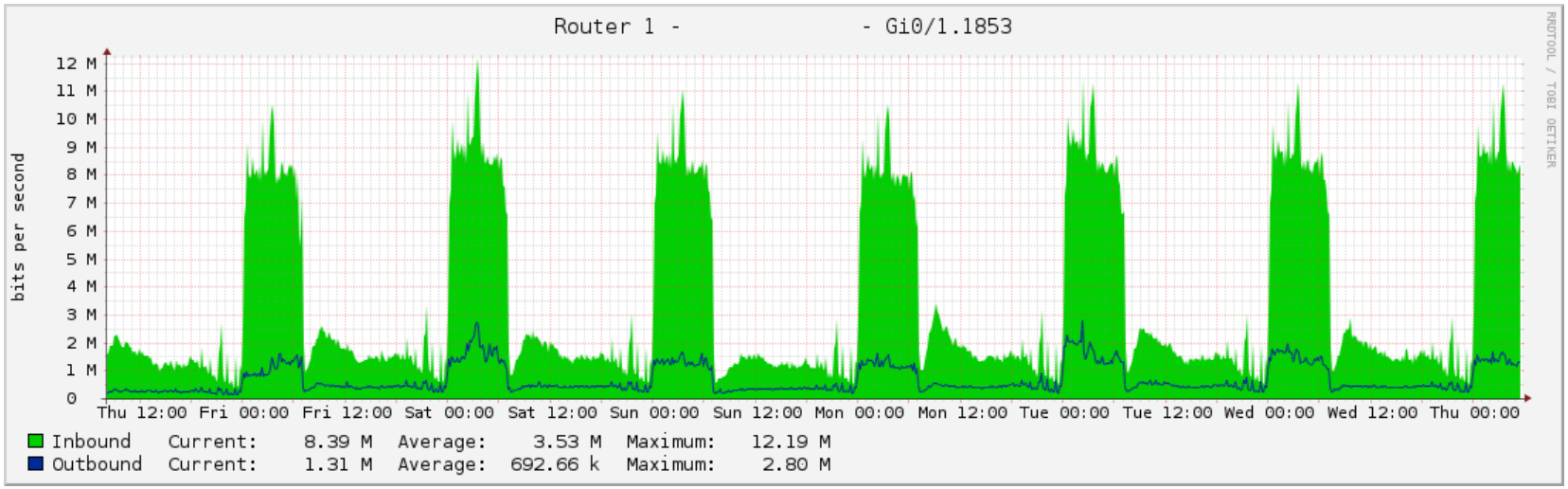
IPScore

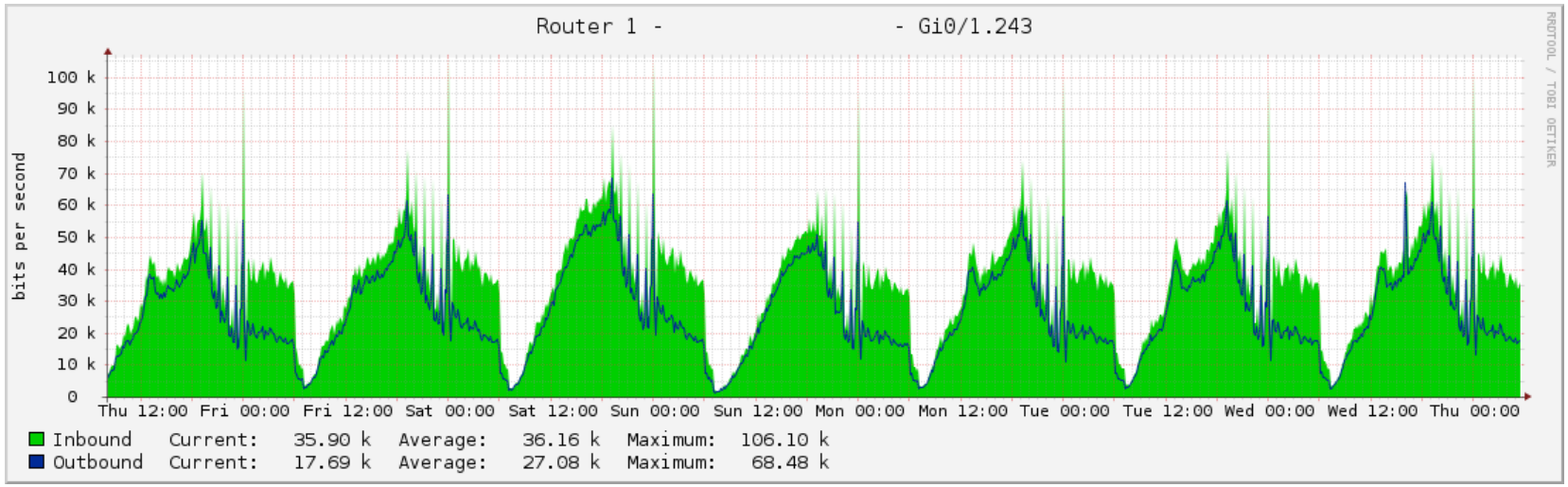
- 5 different metrics
- Some human defined constants to improve success ratio

Network Infrastructure

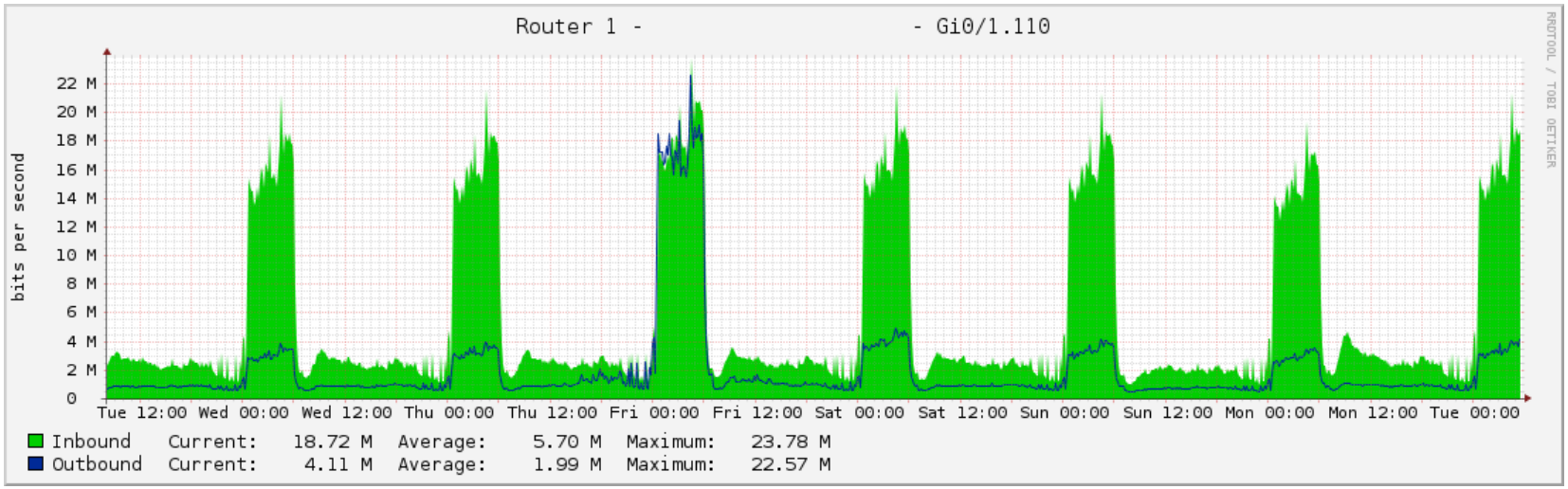


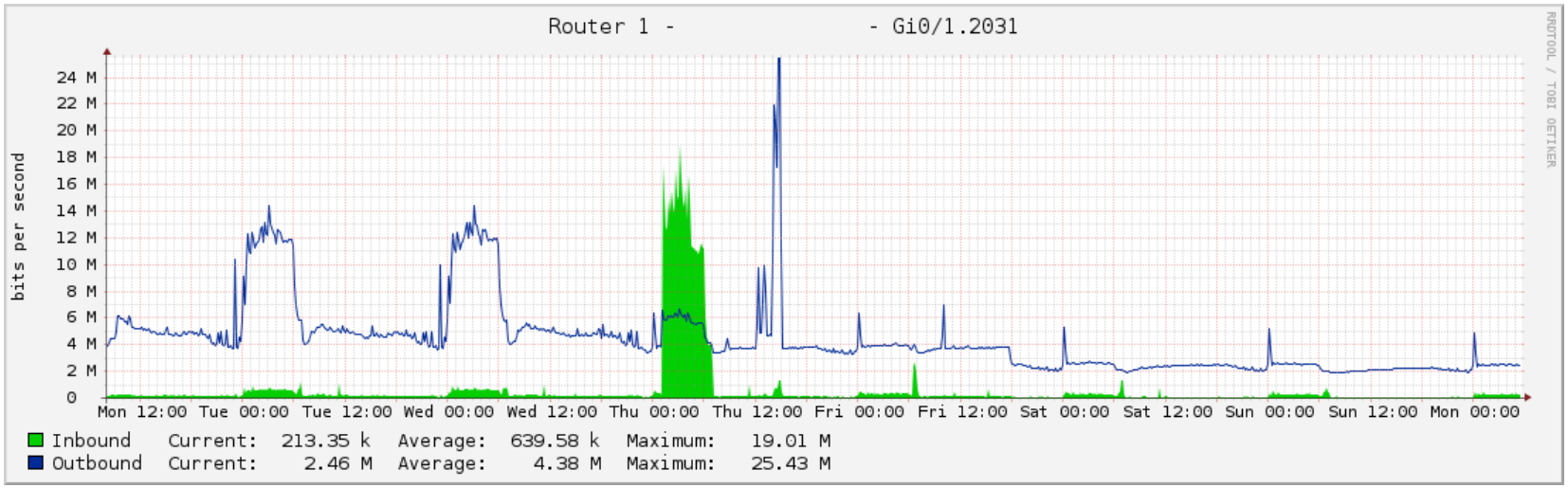
Traffic Patterns (Volume)





Traffic Patterns (Volume)





IPScore

- We proposed a new approach to prevent mainly DoS attacks, brute force and network layer intrusions.

- Our method analyzes network packets in 5 subjects
  - Source of traffic
  - Destination of traffic
  - Volume
  - Variation
  - Time distribution

- Several constants and parameters to calculate IP reputation

IPScore - Constants

Constant	Definition	Description
N_U	Number of users/clients	The number of possible client connections (users/devices) that the system has. External connections to the system are made by users of the services. For instance, for a university system, the population of students and staff can be the maximum number for expected clients. This constant should roughly be defined by the system administrators. Clients can either be from users or devices.
N_i	Number of internal nodes	The number of IP addresses used for the services and servers including the clusters.
N_s	Number of services	The number of available services (such as FTP, HTTP, POP, SMTP, etc.) that the system offers.
t	Tolerance	Tolerance ratio (%) (Default: %15)
L	Threshold Limit	Threshold score for deny operation defined by the system admin depending on the risk appetite of the organization 0.0 – 1.0

IPScore - Parameters

Parameter	Definition
n_{max}	Maximum(peak) number of packets/10 minutes for a service in data set at all times
n_{avg}	Average of number of packets/ 10 minutes for a service in data set
n_{cur}	Number of packets to same service in last 10 minutes
t_{ns}	Number of distinct source addresses in last 10 minutes towards to same port
t_{nd}	Number of distinct destination addresses (ip+port) in last 10 minutes from same source
n_{days}	Number of days that source IP observed in history before the current day
h	Number of days that training data has

IPScore – Score Metrics

1. Static Rule Score (K)
2. Source Score (S)

IF $t_{ns} < N$ THEN $S = 1 - ((1-L) * (t_{ns} / N))$
ELSE $S = L + 1 - (t_{ns} / N)$

3. Familiarity Score (F)

IF $n_{days} > 0$ THEN $F = L + (1-L) * n_{days} / h$
ELSE F= 0

4. Traffic Score (T)

IF $n_{cur} > n_{max}$ THEN T=0 ($n_{max} \pm \%t$)
ELSE IF $n_{cur} > n_{avg}$ then $T = (n_{max} - n_{cur}) / (n_{max} - n_{avg})$
ELSE T=1.0

5. Variation Score (V)

IF $t_{nd} < N_s$ THEN $V = 1 - ((1-L) * (t_{nd} / N_s))$
ELSE $V = L + 1 - (t_{nd} / N_s)$

IPScore – Score Metrics

- Overal Score:

P = K + AVG (S + F + T +V)
IF P > L THEN allow packet
ELSE deny packet

- We keep track of 10-minutes traffic in memory for faster processing

IP Score – Implementation

- Data taken from a payment gateway company
- Processed traffic logs from all network devices
- **Training data:** 1 week / 31 million requests
- **Each day:** ~5 million requests
- ~ 50 internal nodes, 30 services, 400K clients

- Data Sanitization before processing
- Sensitive data removed
- Analyzed data in 10-minute resolution
- Simulate real time analysis on runtime

Experiments

- Success rate depends on training data, risk appetite (threshold) and tolerance ratios
- Training data updated on run time
- Limited computing resource for longer duration run time analysis

Next Steps

- Experiments on different attack scenarios

- **Future goals:** Integrate resource usage for network & server components as a metric and geo-location data for IP reputation scoring. (Out of scope in the project)

References

1. W. G. Morein, A. Stavrou, D. L. Cook, A. D. Keromytis, V. Misra, D. Rubensteiny, Using Graphic Turing Tests To Counter Automated DDoS Attacks Against Web Servers, in: Proceedings of the 10th ACM conference on Computer and communications security, Washington, DC, USA, 2003.
2. J.-F. Podevin, Telling humans and computers apart automatically, COMMUNICATIONS OF THE ACM 47 (2) (2004) 57–60.
3. G. Mori, J. Malik, Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, 2003.
4. E. Athanasopoulos, S. Antonatos, Enhanced CAPTCHAs: Using Animation to Tell Humans and Computers Apart, in: Proceedings of Communications and Multimedia Security, 2006, pp. 97–108.
5. H. D. Truong, C. F. Turner, C. C. Zou, iCAPTCHA: The Next Generation of CAPTCHA Designed to Defend Against 3rd Party Human Attacks, in: Proceedings of IEEE International Conference on Communications, Kyoto, Japan, 2011.
6. T. Yatagai, T. Isohara, I. Sasase, Detection of HTTP-GET flood Attack Based on Analysis of Page Access Behavior, in: Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, 2007, pp. 232–235.
7. Beitollahi, H., & Deconinck, G. (2012). Tackling Application-layer DDoS Attacks. Procedia Computer Science, 10, 432-441. doi:10.1016/j.procs.2012.06.056
8. H. Beitollahi, G. Deconinck, ConnectionScore: A Statistical Technique to Resist Application-layer DDoS Attacks, Tech. Rep. 01-2012-0130, Electrical Engineering Department, University of Leuven, Belgium, http://www.esat.kuleuven.be/electa/publications/fulltexts/pub 2313.pdf (2012).

Questions