Using cyber intelligence to detect and localize botnets

ENRICO BRANCA

Botconf'13

5-6 December 2013, Nantes, France.

IDEA

Create a **cyber intelligence** system able to:

- Analyse network communications
- Detect and identify botnet activities
- Identify malware sources
- Perform passive protocol analysis
- Analyse SSL communication
- Store massive amount of data
- Perform statistical analysis (cross-clusters, multivariate, etc..)
- Operate on a low-end consumer system (500-1000€ PC)
- Process live or recorded information coming from a variety of sources.

TARGET

Build an application to identify, collect, analyze and distill open and public information to generate actionable security information.

Core Team

Security Architect

Enrico Branca

- More than 12 years of hand-on experience across Europe
- Security researcher since 2001
- Designed high-budget solutions for CAC-40 companies
- Implemented innovative solutions across many business

Senior Developer

Federico Figus

- Subject matter expert in Python, C, Java and R
- More than 6 years of experience with Enterprise Coding
- Professional knowledge of Secure Programming
- Speaker in international conference

Legal Expert

Luis Enriquez

- Specialist in Open Source and FL/OSS licensing
- Recognized as point of reference in European legal market
- Author of a book on software licensing

PROBLEM

PROBLEM		SOLUTION
Information is difficult to identify and collect even when you know where to look and what you need.	Data Discovery	**A platform** for exploring information from any source.
Time is a critical concern for customers generating value from information assets.	Automation	**A multi-agent solution** that automates the integration and movement of data.
No easy way exists to extract information from open and public data to generate intelligence.	Profiling	**A system** able to correlate data and recognize patterns.

TECHNICAL PROBLEM

Python low level libraries are not made with security in mind and have no checks or limits

So we have decided to write new python libraries

- New “**os**” library to enable secure read and secure write to disk, streams or sockets
- New “**sys**” library to deal with system specific call and to have an interface to system statistics and counters
- New “**socket**” library able to deal with illegal or malformed communication without having to delete information
- New libraries designed to work with malformed or malicious traffic for “**HTTP, FTP, SMTP, POP, IMAP, NNTP, BitTorrent, SSH, SSL, IRC, Telnet, DNS, SSH, NTP**”
- New libraries to handle **string operation** and **string management** to eliminate memory or encoding attacks

TOOL OVERVIEW

The software does:

- Supports 16 connection protocols
- Remove duplicates from input data
- Organize unstructured data
- Load data in any format even binary
- Extract data and metadata from files
- Correlate data to extract intelligence

The software does NOT:

- Use hacking techniques to find data
- Perform penetration tests on servers
- Remove passwords from archives
- Crack protocols or system’s defenses
- Infiltrate secure data or communication
- Brute force access any kind of resource

PROTOCOLS

- FTP
- FTPS
- IMAP
- IMAPS
- HTTP
- HTTPS
- DNS
- IRC
- SMTP
- SMTPS
- BitTorrent
- Telnet
- POP
- POPS
- SSH
- NTP

DATA TYPES

- Open Office
- MS Office
- PDF / PS
- MBOX
- Apache Log
- PEM/DER
- PCAP
- BRO IDS
- SNORT

TOOL OVERVIEW

DEVELOPMENT

Code Base:

- 21.765 Source Line of Code

Coding Time:

- 14.400 Man Hours

Software Versioning:

- Currently in ALPHA status
- 14 Major and 527 Minor Releases

TECHNOLOGY

Coding Language:

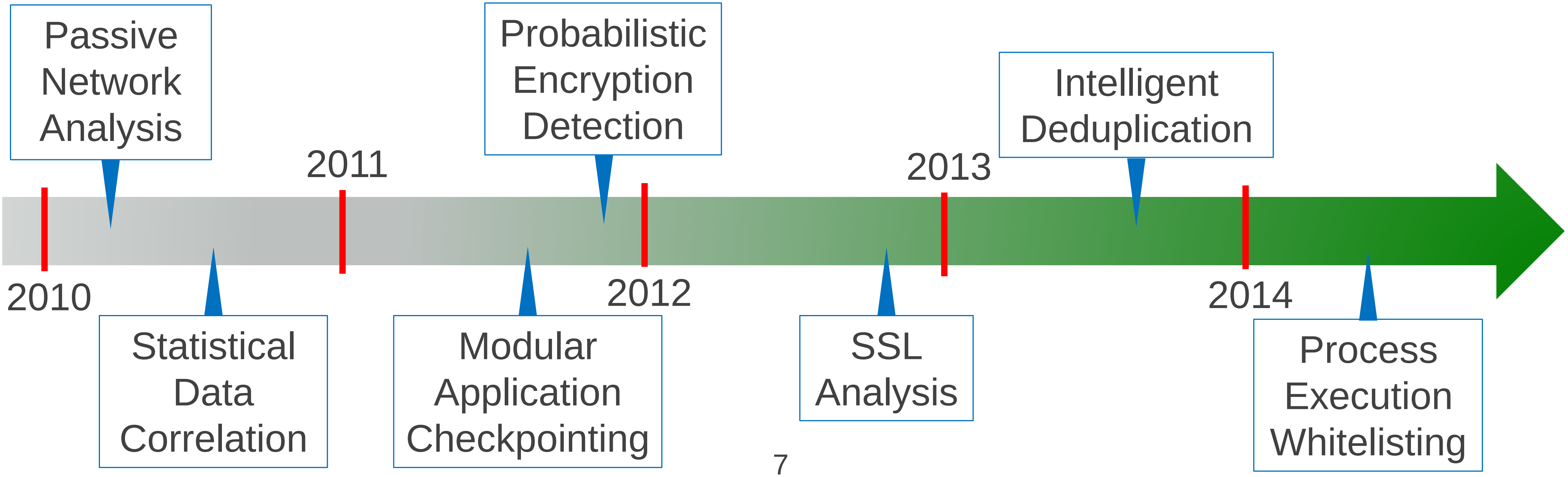
- Python, C, HTML5, JavaScript

System Compatibility:

- Linux (Ubuntu, Debian, Fedora)

OpenSource Components:

- OpenSSL, D3



2010

Passive Network Analysis

Statistical Data Correlation

2011

Modular Application Checkpointing

Probabilistic Encryption Detection

2012

SSL Analysis

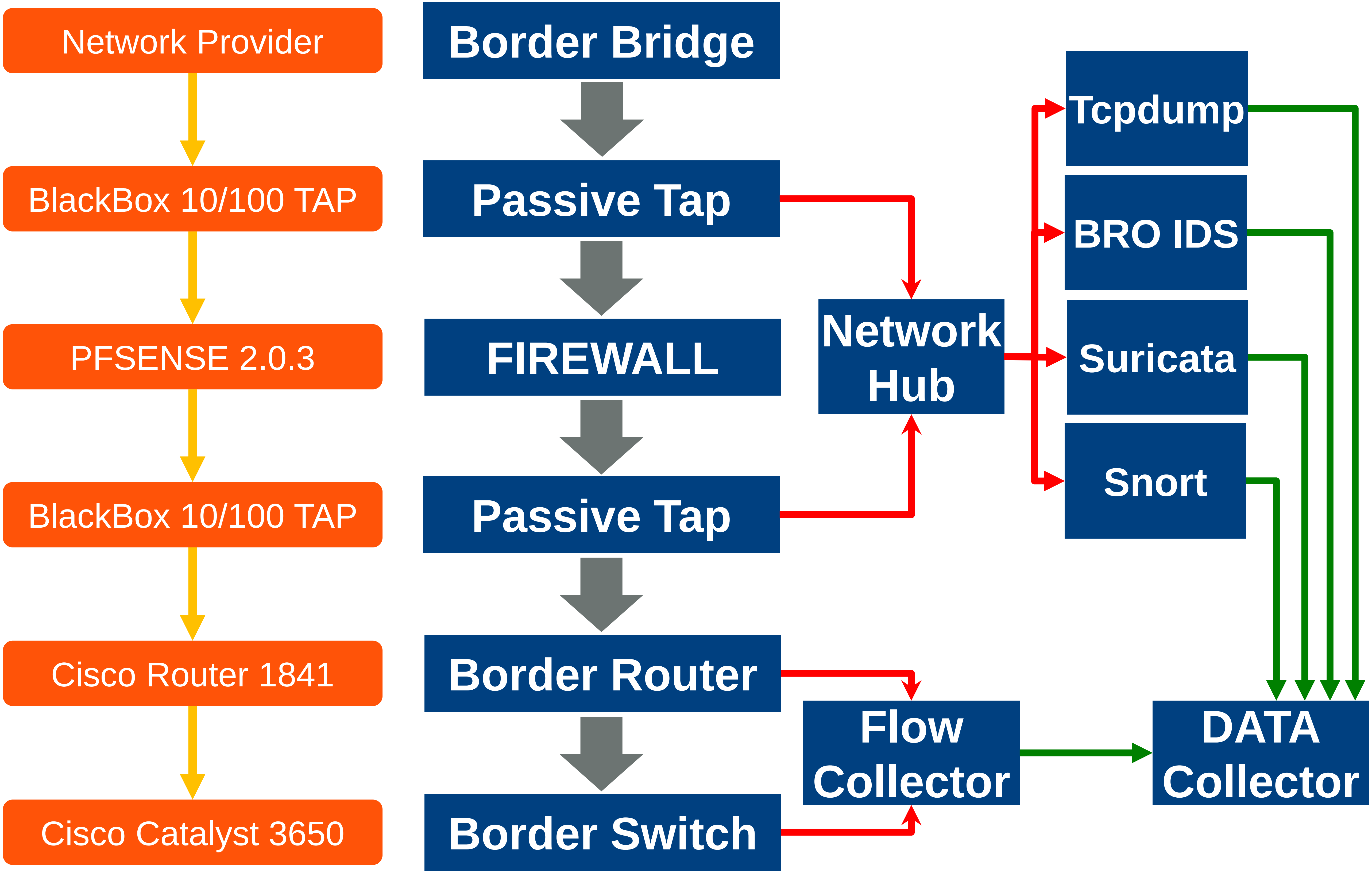
2013

Intelligent Deduplication

2014

Process Execution Whitelisting

TEST LAB



Network Provider

BlackBox 10/100 TAP

PFSENSE 2.0.3

BlackBox 10/100 TAP

Cisco Router 1841

Cisco Catalyst 3650

Border Bridge

Passive Tap

FIREWALL

Passive Tap

Border Router

Border Switch

Network Hub

Tcpdump

BRO IDS

Suricata

Snort

Flow Collector

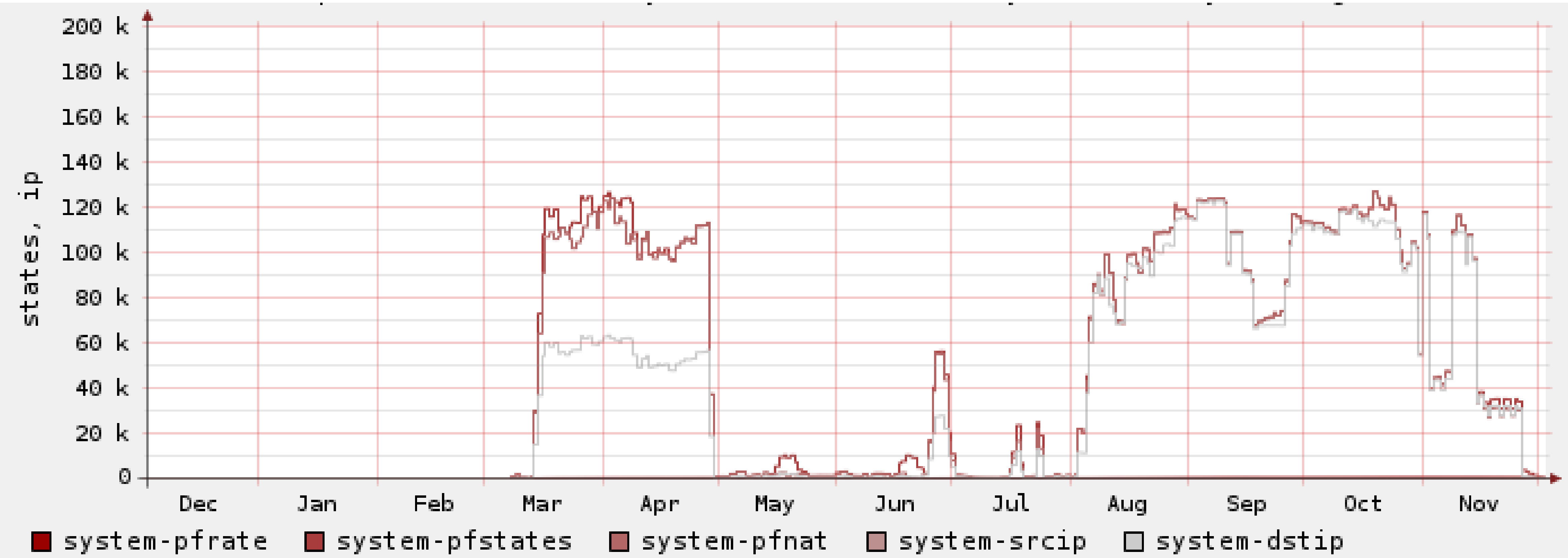
DATA Collector

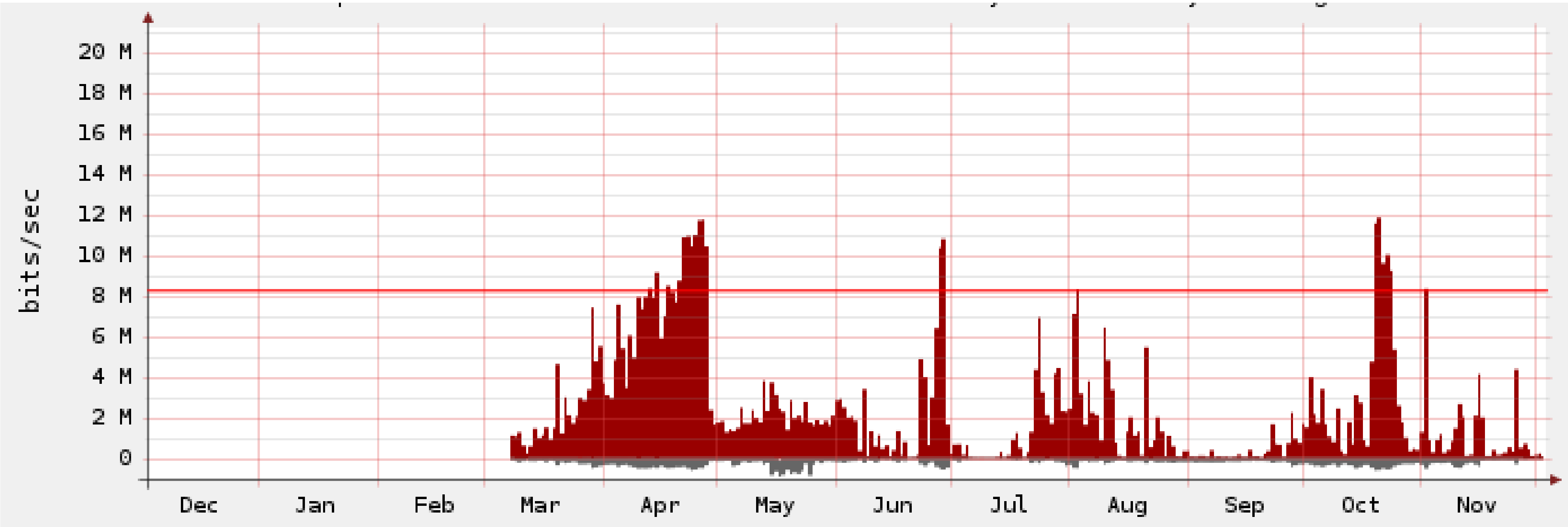
TEST LAB

Setting up the test lab:

- Setup Honeypot DIONAEA and use custom python libraries
- Install VMWARE SERVER with 5 Windows and 5 Linux systems
- Block in each client traffic on ports “1-50, 80, 1139, 3000-7000”
- Expose systems to internet traffic and use them to browse the web
- Subscribe to all RSS feeds of World Top 100 Newspapers
- Subscribe to all RSS feeds of World Top 100 JOB sites
- Subscribe to RSS feeds of Top 10 “Paste tool” sites (i.e., Pastebin)
- Subscribe to 2000 high traffic not moderated mailing lists
- Subscribe to 2000 high traffic moderated mailing lists
- Daily download top million ALEXA site list
- Daily Select top 100.000 websites
- Use AJAX browser to connect to each website and each RSS
- Load static and dynamic/scripted content
- Record all traffic required to visualize website
- (NO CRAWLER-SCANNER-ROBOT used at any stage)
- Save all contents received from website

TEST TRAFFIC





TEST CASE #1

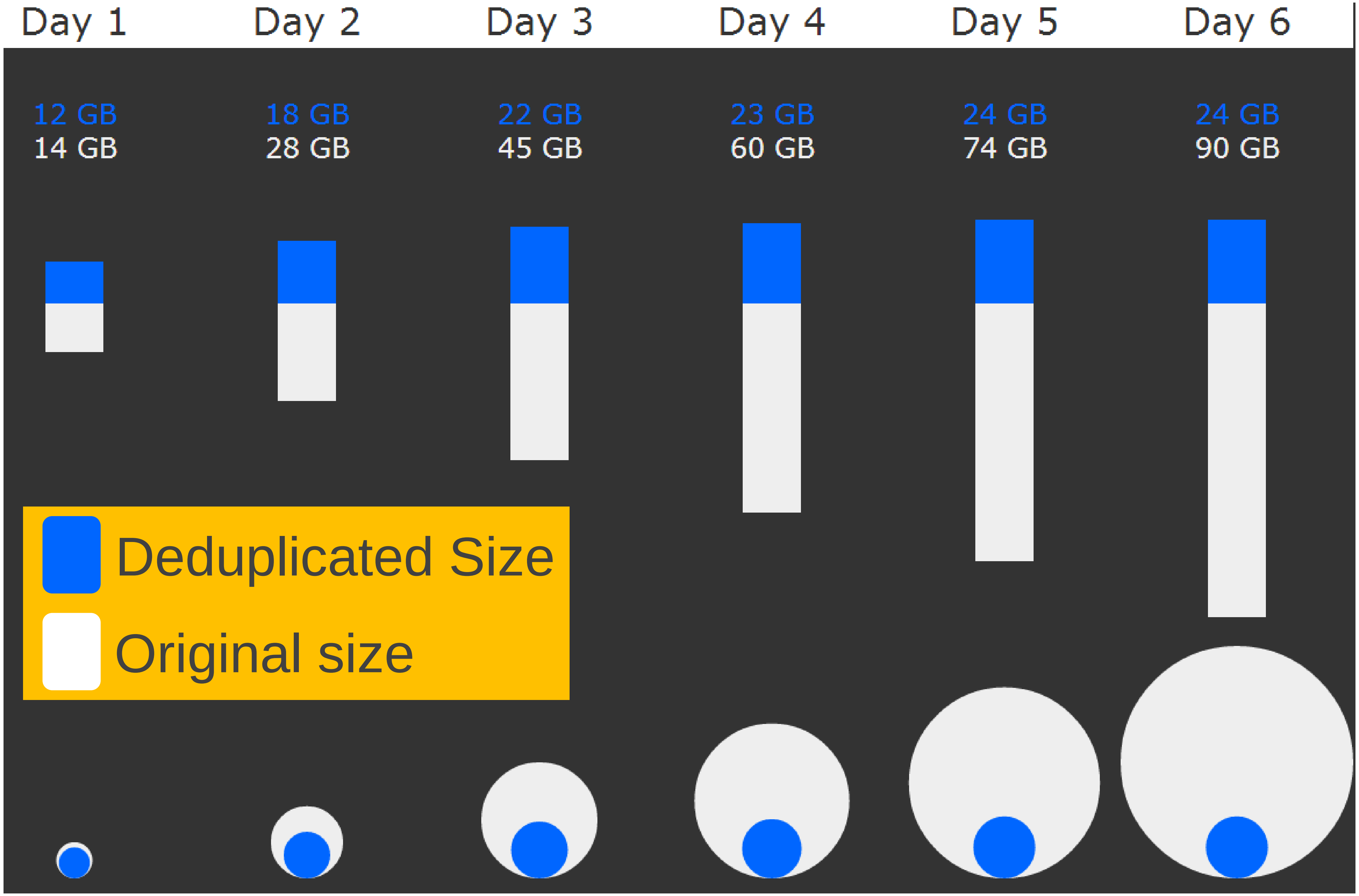
Test Case #1

Identify malware and attack sources by correlating email spam and scripts on high traffic websites using archived traffic

- Collect data from March to June 2013
- Analyse saved flow for temporal patterns
- Analyse saved flow for spatial patterns
- Analyse saved traffic for protocol anomalies
- Analyse saved traffic for data anomalies
- Analyse saved traffic for string anomalies
- Correlate results of each test and aggregate results
- Use aggregated results to identify possible files and sources
- Analyse identified files for viruses/malware
- Analyse identified files for entropy or similarity patterns

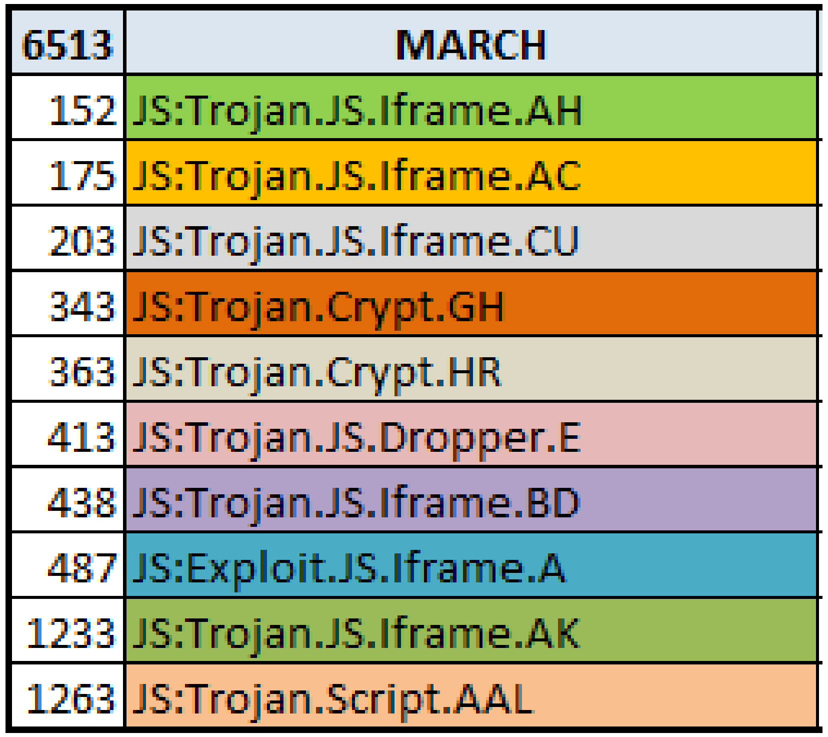
DATA DEDUPLICATION

Data Collection and Deduplication (one week example)

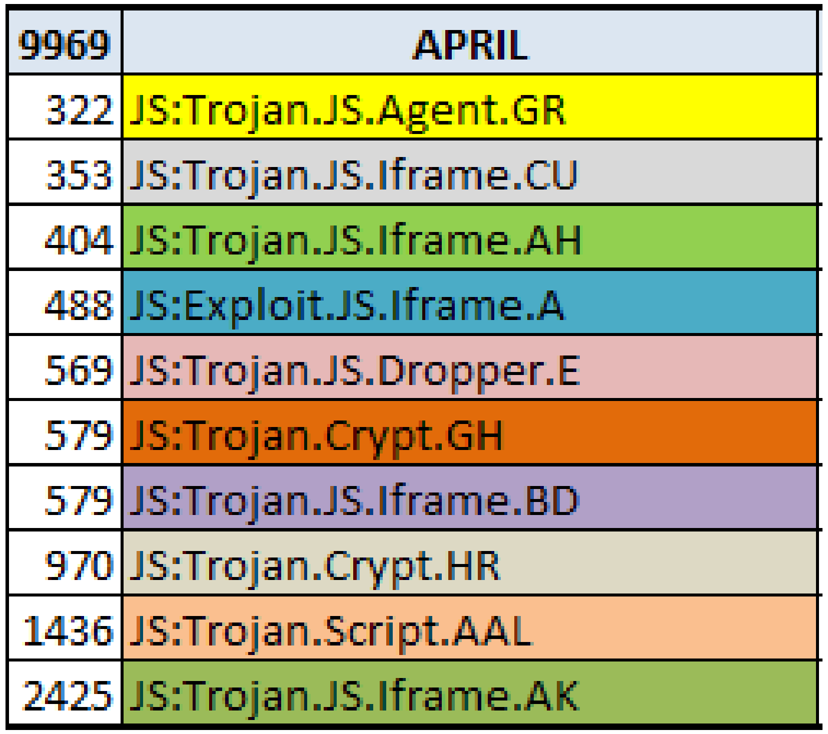


ANTIVIRUS TEST #1

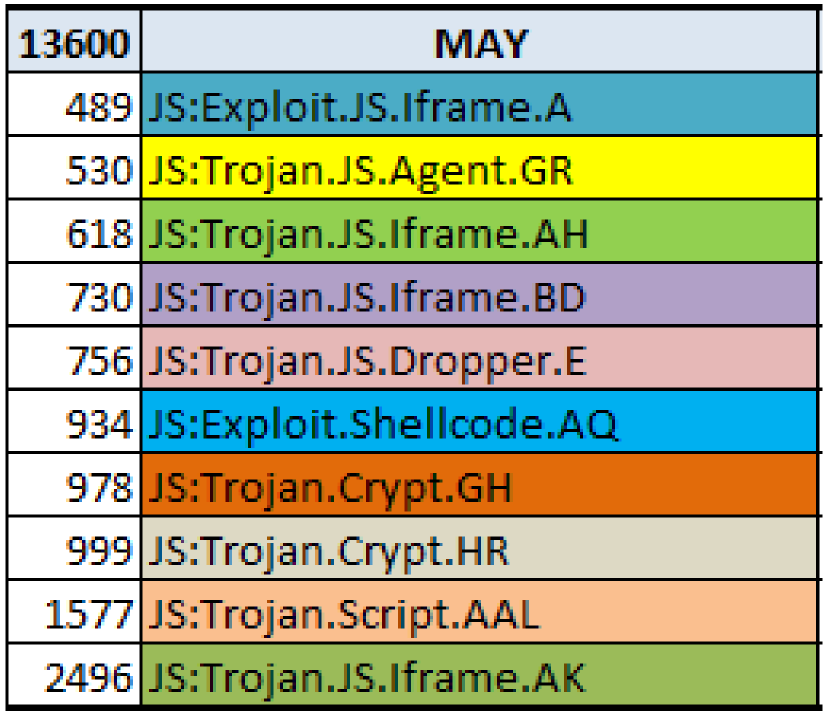
Antivirus: **Bitdefender (top 10)**



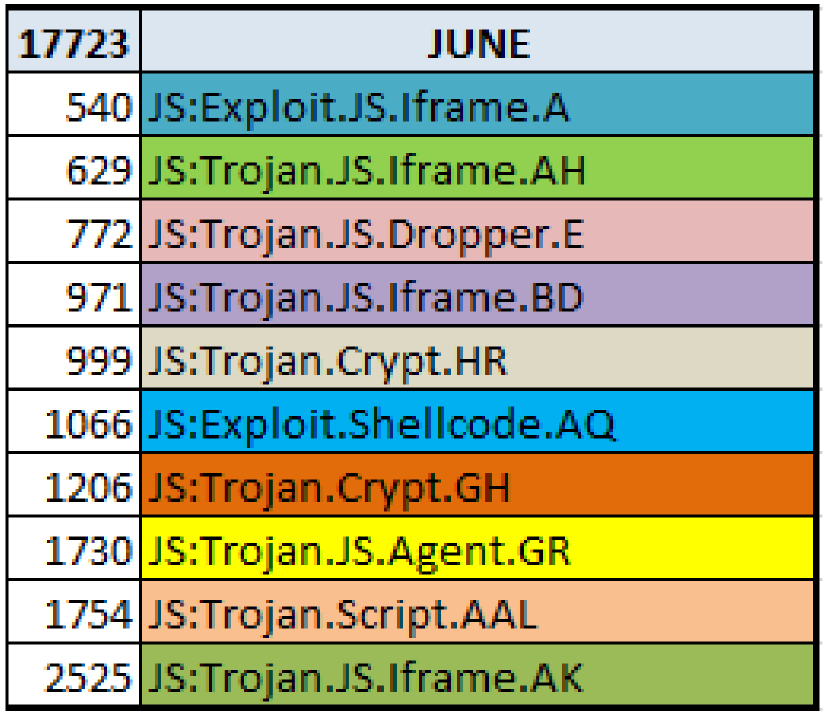
6513	MARCH
152	JS:Trojan.JS.Iframe.AH
175	JS:Trojan.JS.Iframe.AC
203	JS:Trojan.JS.Iframe.CU
343	JS:Trojan.Crypt.GH
363	JS:Trojan.Crypt.HR
413	JS:Trojan.JS.Dropper.E
438	JS:Trojan.JS.Iframe.BD
487	JS:Exploit.JS.Iframe.A
1233	JS:Trojan.JS.Iframe.AK
1263	JS:Trojan.Script.AAL



9969	APRIL
322	JS:Trojan.JS.Agent.GR
353	JS:Trojan.JS.Iframe.CU
404	JS:Trojan.JS.Iframe.AH
488	JS:Exploit.JS.Iframe.A
569	JS:Trojan.JS.Dropper.E
579	JS:Trojan.Crypt.GH
579	JS:Trojan.JS.Iframe.BD
970	JS:Trojan.Crypt.HR
1436	JS:Trojan.Script.AAL
2425	JS:Trojan.JS.Iframe.AK



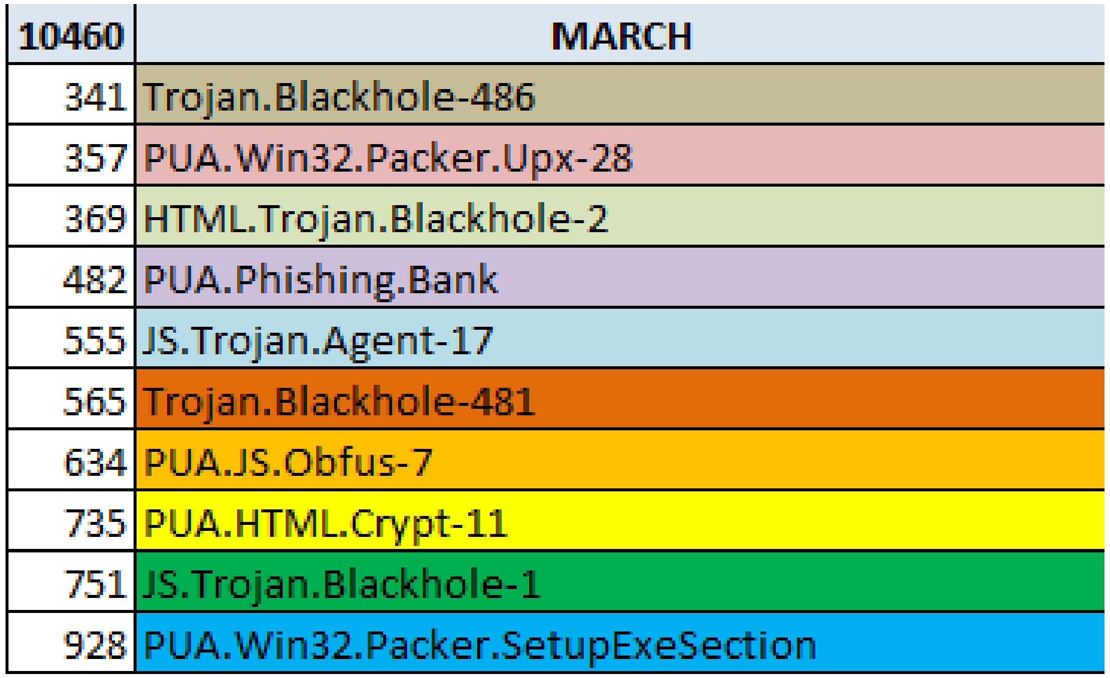
13600	MAY
489	JS:Exploit.JS.Iframe.A
530	JS:Trojan.JS.Agent.GR
618	JS:Trojan.JS.Iframe.AH
730	JS:Trojan.JS.Iframe.BD
756	JS:Trojan.JS.Dropper.E
934	JS:Exploit.Shellcode.AQ
978	JS:Trojan.Crypt.GH
999	JS:Trojan.Crypt.HR
1577	JS:Trojan.Script.AAL
2496	JS:Trojan.JS.Iframe.AK



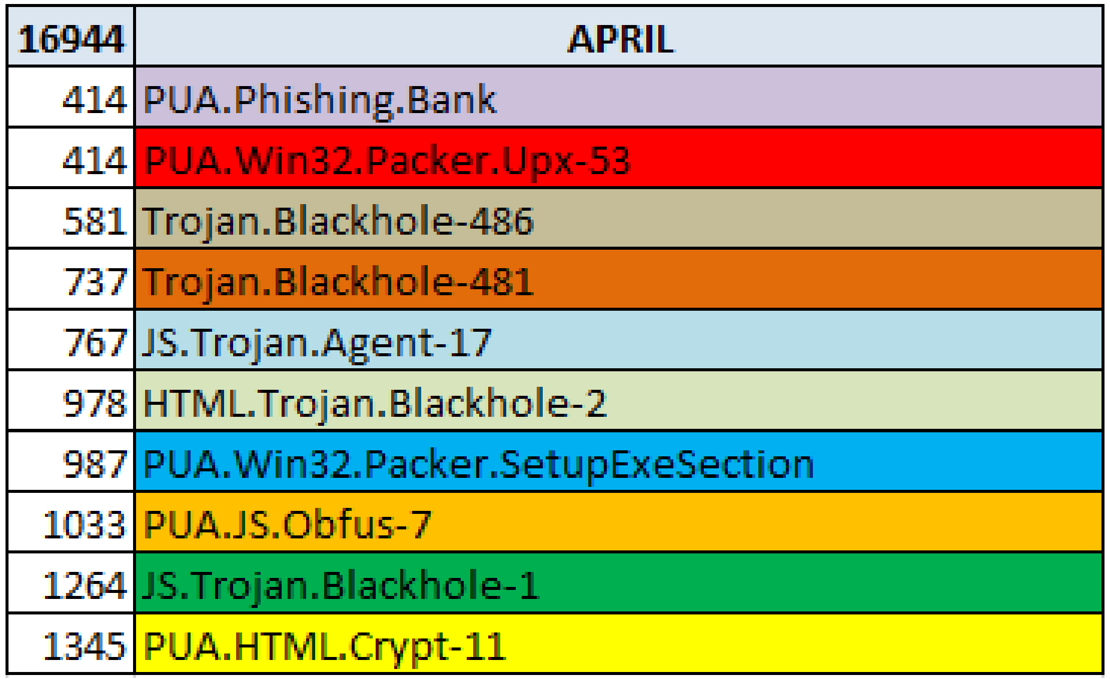
17723	JUNE
540	JS:Exploit.JS.Iframe.A
629	JS:Trojan.JS.Iframe.AH
772	JS:Trojan.JS.Dropper.E
971	JS:Trojan.JS.Iframe.BD
999	JS:Trojan.Crypt.HR
1066	JS:Exploit.Shellcode.AQ
1206	JS:Trojan.Crypt.GH
1730	JS:Trojan.JS.Agent.GR
1754	JS:Trojan.Script.AAL
2525	JS:Trojan.JS.Iframe.AK

ANTIVIRUS TEST #2

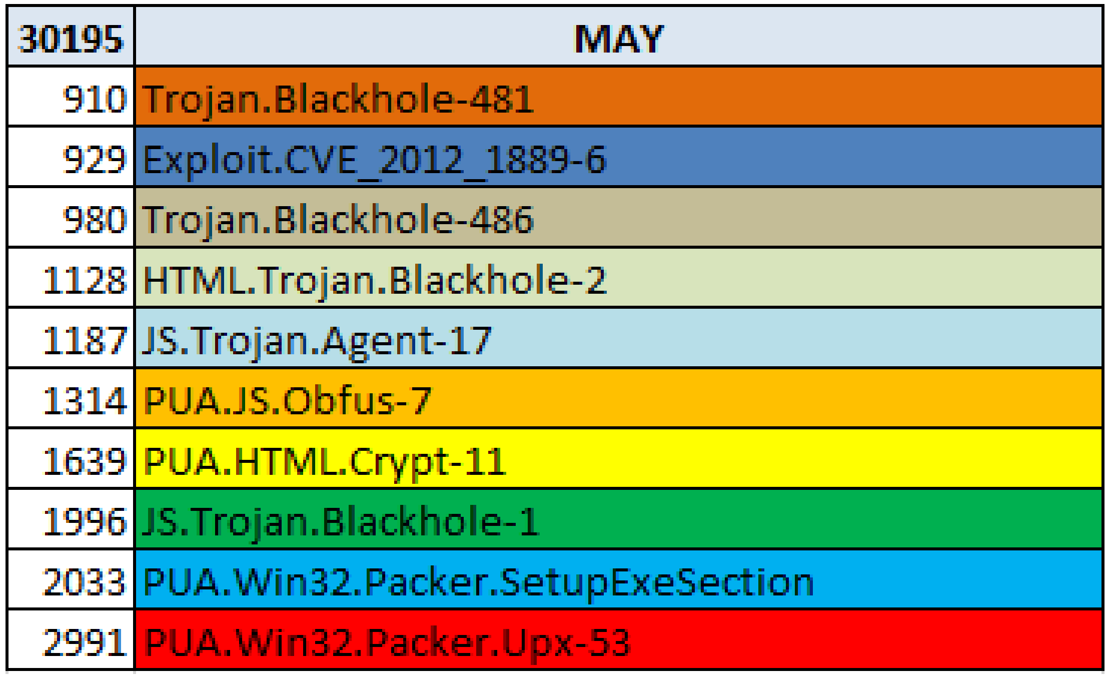
Antivirus: **Clamav (top 10)**



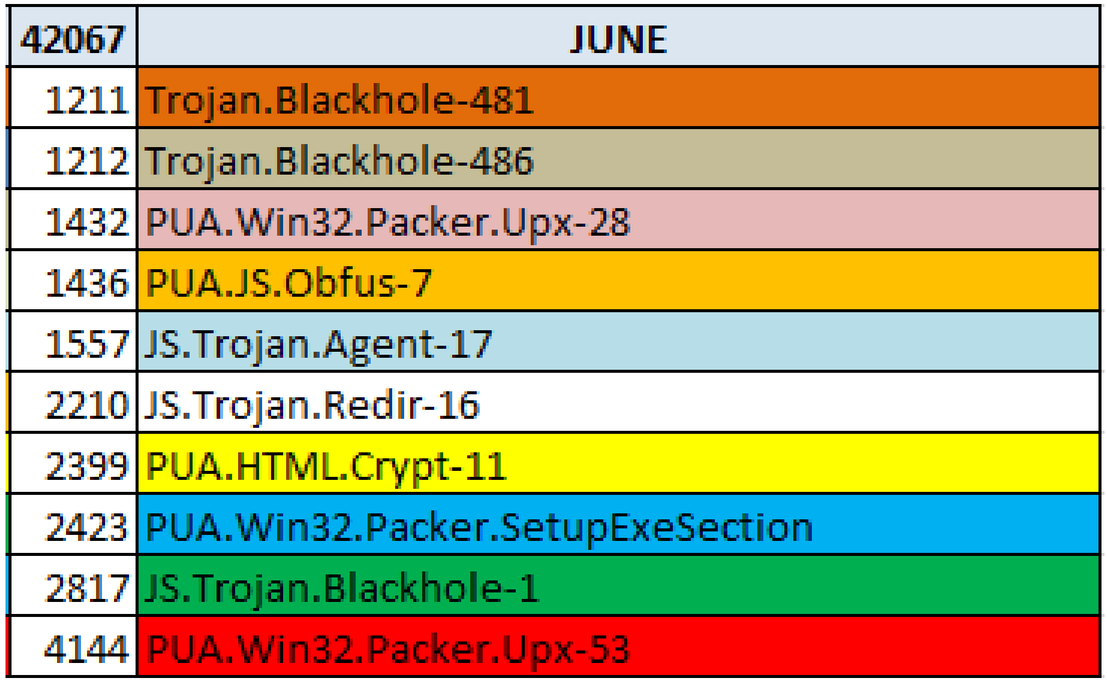
10460	MARCH
341	Trojan.Blackhole-486
357	PUA.Win32.Packer.Upx-28
369	HTML.Trojan.Blackhole-2
482	PUA.Phishing.Bank
555	JS.Trojan.Agent-17
565	Trojan.Blackhole-481
634	PUA.JS.Obfus-7
735	PUA.HTML.Crypt-11
751	JS.Trojan.Blackhole-1
928	PUA.Win32.Packer.SetupExeSection



16944	APRIL
414	PUA.Phishing.Bank
414	PUA.Win32.Packer.Upx-53
581	Trojan.Blackhole-486
737	Trojan.Blackhole-481
767	JS.Trojan.Agent-17
978	HTML.Trojan.Blackhole-2
987	PUA.Win32.Packer.SetupExeSection
1033	PUA.JS.Obfus-7
1264	JS.Trojan.Blackhole-1
1345	PUA.HTML.Crypt-11



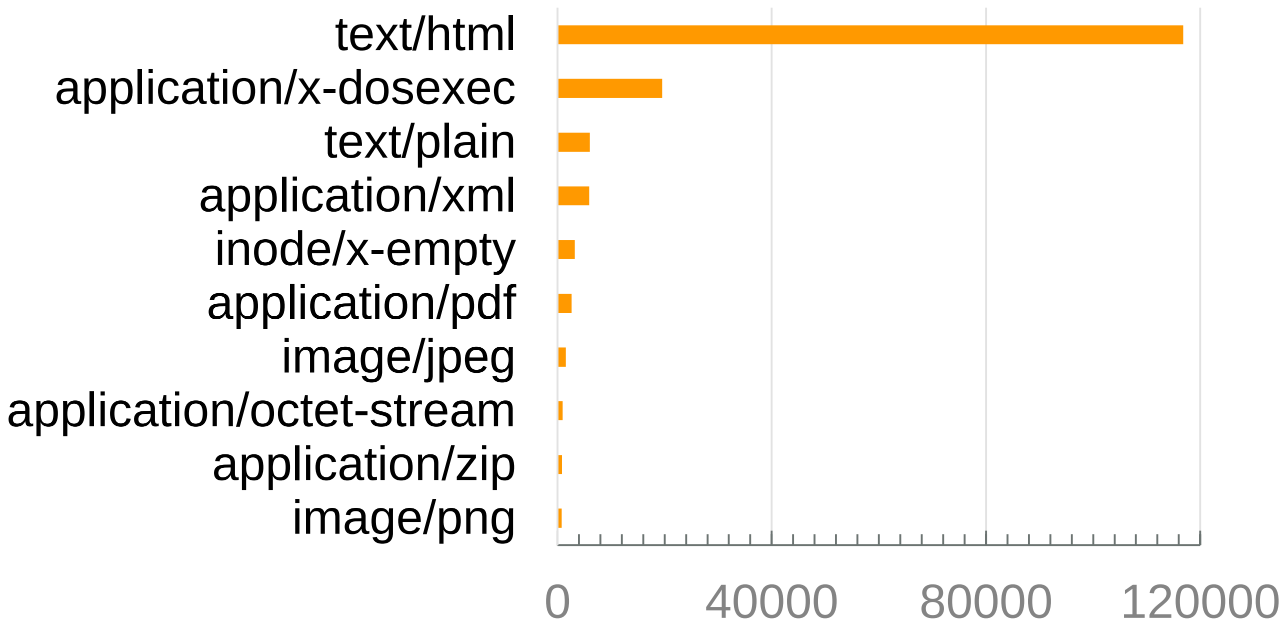
30195	MAY
910	Trojan.Blackhole-481
929	Exploit.CVE_2012_1889-6
980	Trojan.Blackhole-486
1128	HTML.Trojan.Blackhole-2
1187	JS.Trojan.Agent-17
1314	PUA.JS.Obfus-7
1639	PUA.HTML.Crypt-11
1996	JS.Trojan.Blackhole-1
2033	PUA.Win32.Packer.SetupExeSection
2991	PUA.Win32.Packer.Upx-53



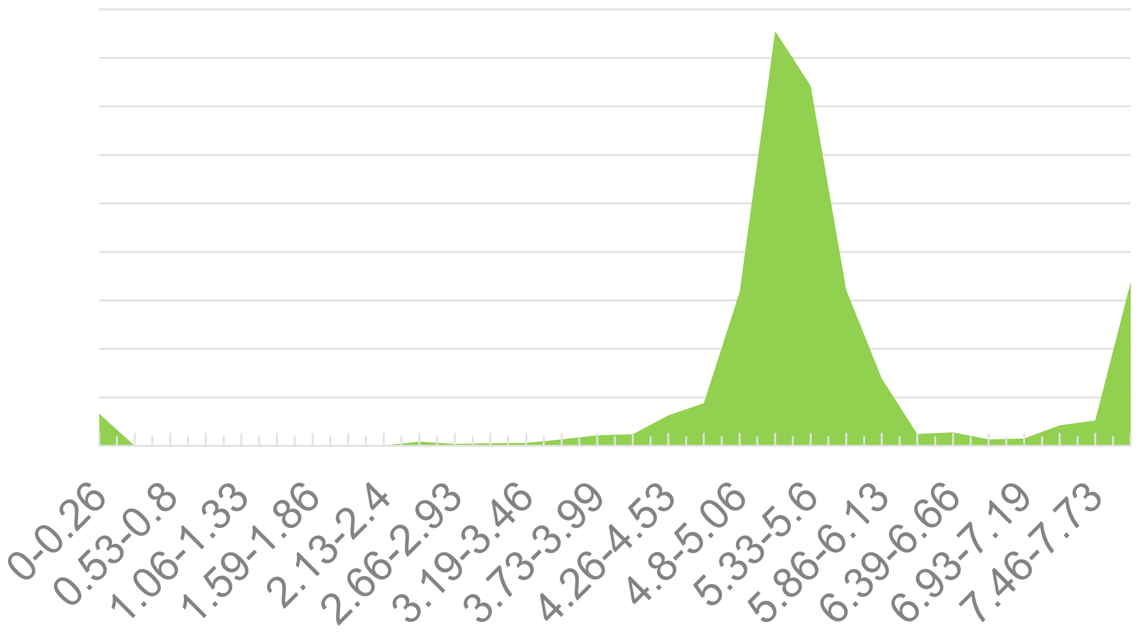
42067	JUNE
1211	Trojan.Blackhole-481
1212	Trojan.Blackhole-486
1432	PUA.Win32.Packer.Upx-28
1436	PUA.JS.Obfus-7
1557	JS.Trojan.Agent-17
2210	JS.Trojan.Redir-16
2399	PUA.HTML.Crypt-11
2423	PUA.Win32.Packer.SetupExeSection
2817	JS.Trojan.Blackhole-1
4144	PUA.Win32.Packer.Upx-53

TEST CASE #1

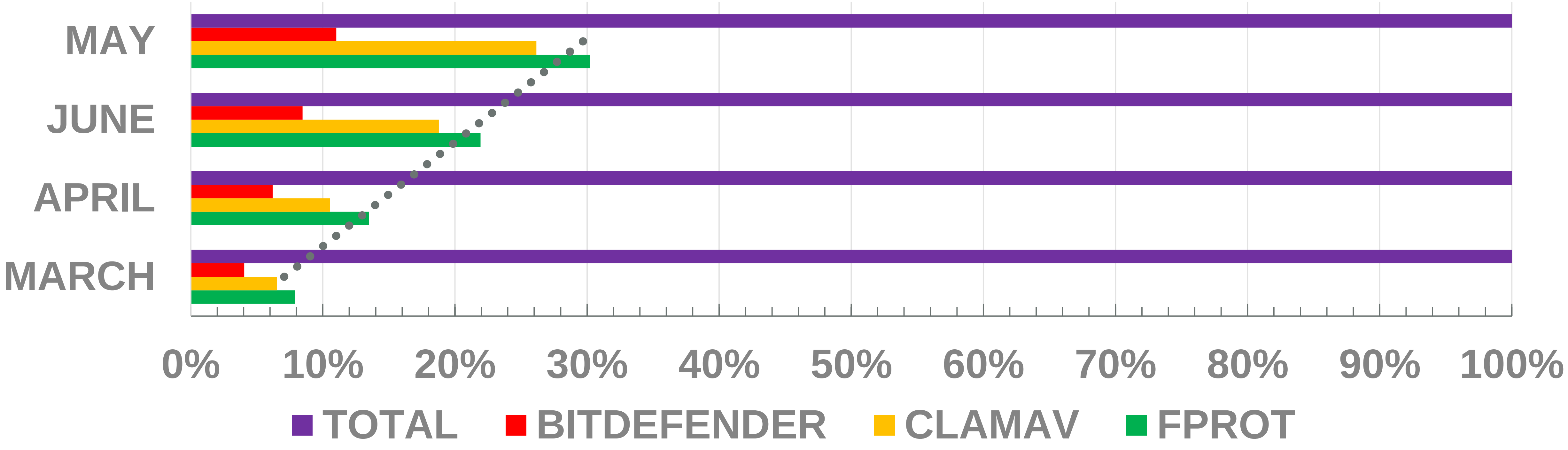
Top 10 MIME types



File Entropy Distribution



Virus Detection Over Time



TEST CASE #2

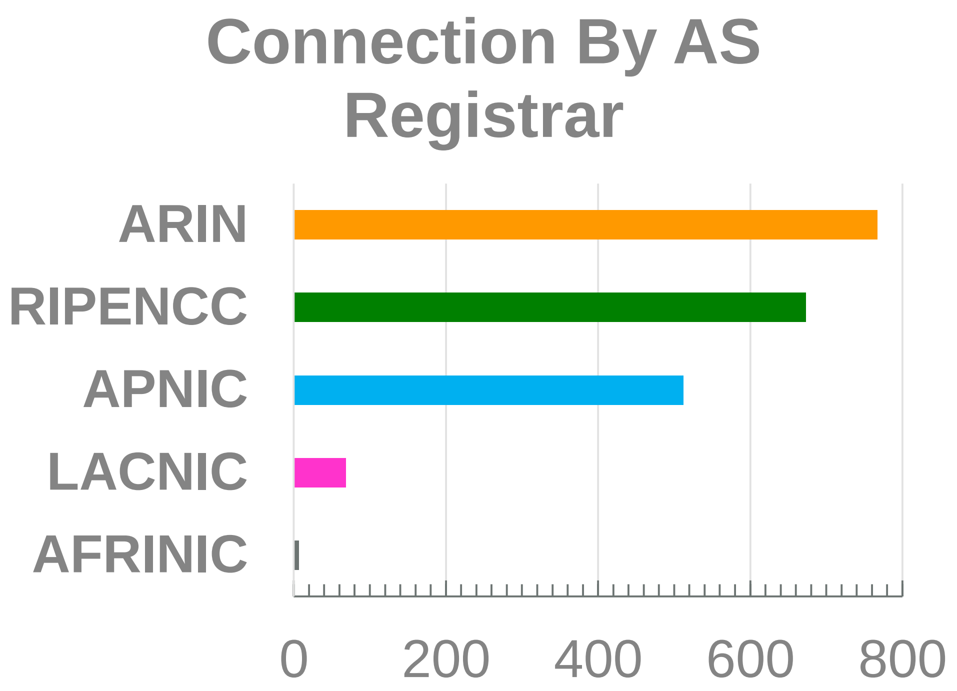
Test Case #2

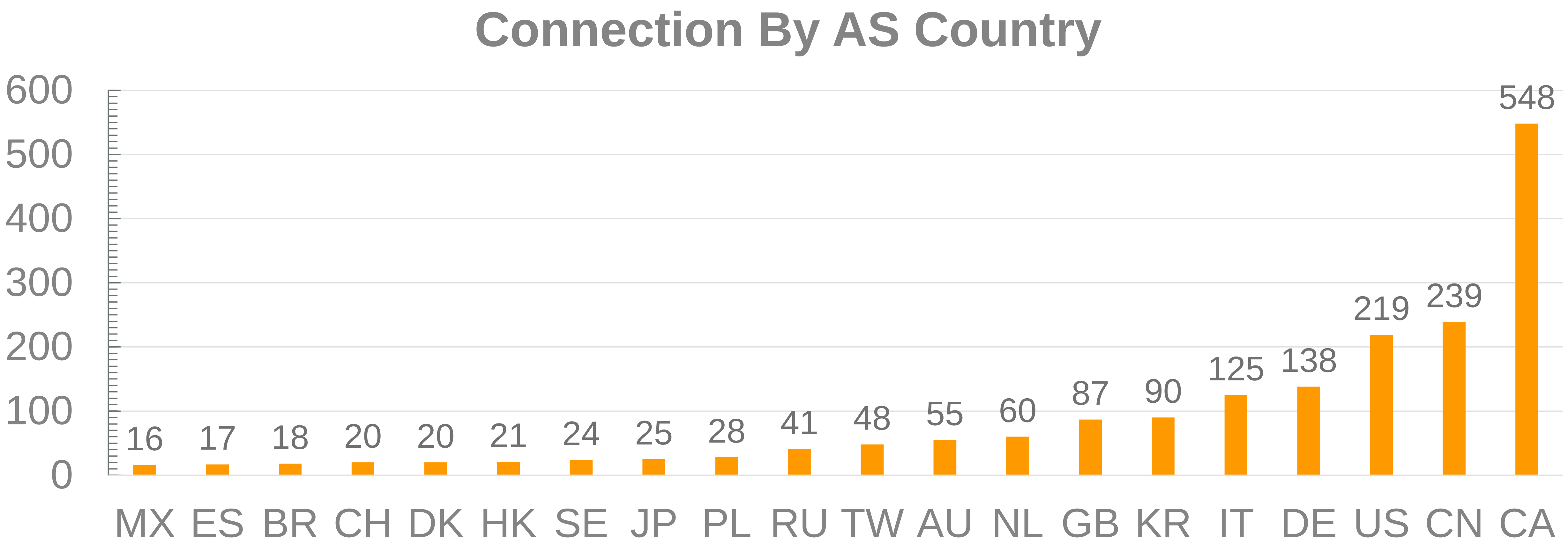
Botnet Tracking using passive network analysis

- Collect data from March to June 2013
- Analyse saved flow for temporal patterns
- Analyse saved flow for spatial patterns
- Analyse saved traffic for protocol anomalies
- Analyse saved traffic for data anomalies
- Analyse saved traffic for string anomalies
- Correlate results of each test and aggregate results
- Use aggregated results to identify possible botnet traffic
- Confirm traffic is related to botnet
- Geolocate IP address and identify Autonomus Systems
- Visualize findings

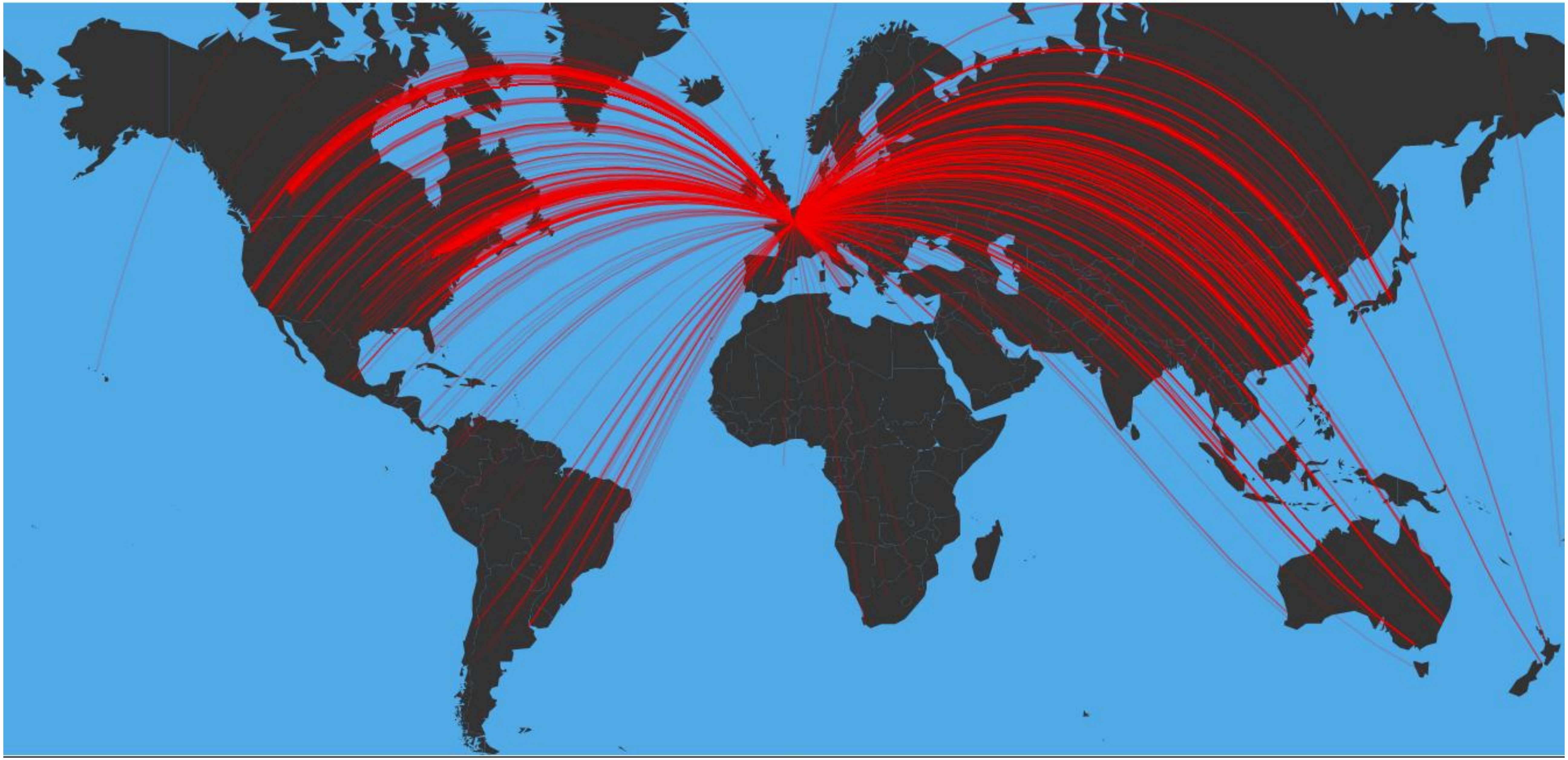
Botnet Connection by AS

AS Description	Count
BARR-XPLR-ASN - Xplornet Communications Inc.	522
CHINANET-BACKBONE No.31,Jin-rong Street	110
ASN-IBSNAZ Telecom Italia S.p.a.	71
CHINA169-BACKBONE CNCGROUP China169 Backbone	55
KIXS-AS-KR Korea Telecom	48
DTAG Deutsche Telekom AG	44
HINET Data Communication Business Group	25
ASN-INFOSTRADA WIND Telecomunicazioni S.p.A.	23
BT-UK-AS BTnet UK Regional network	23
LGI-UPC Liberty Global Operations B.V.	23

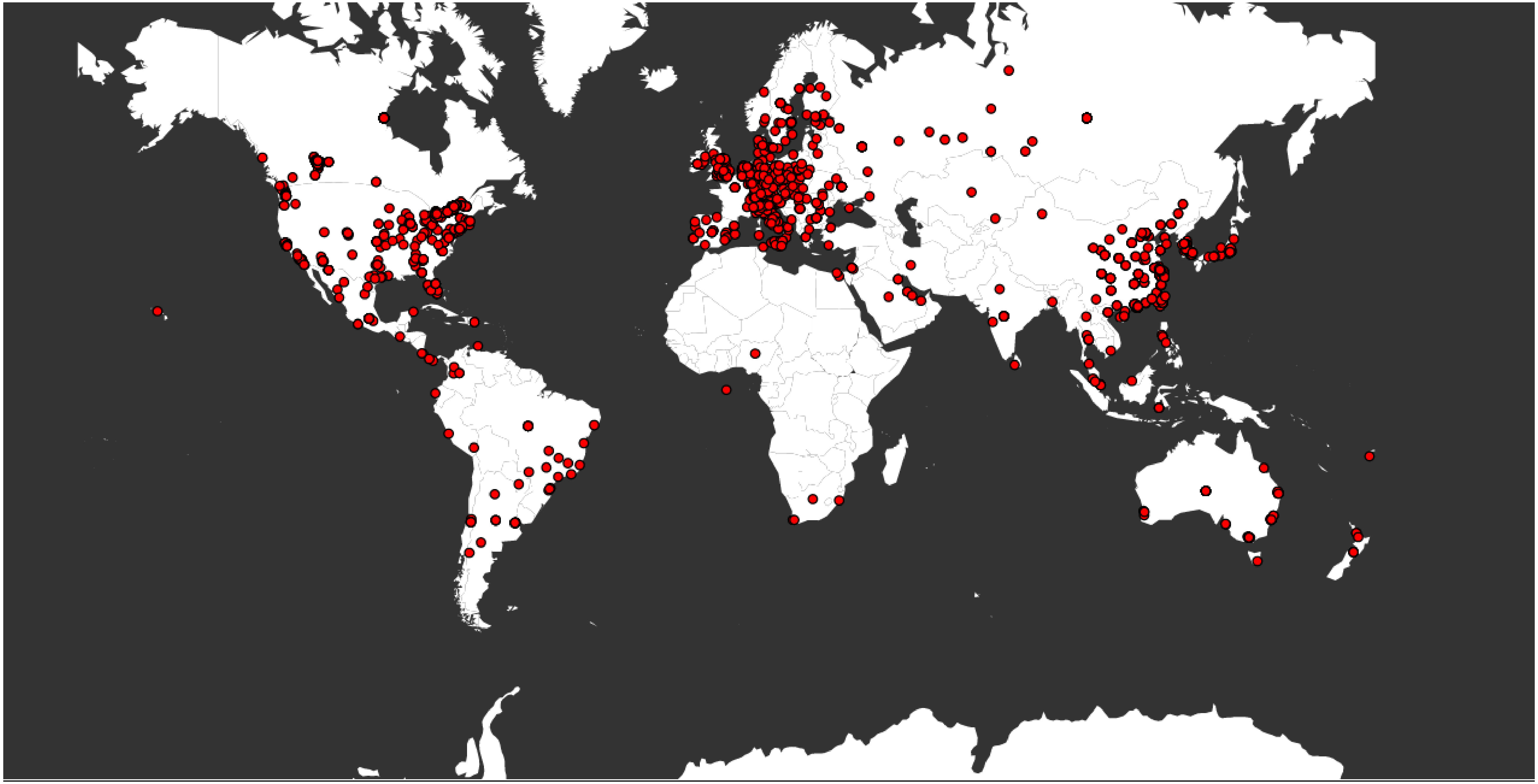




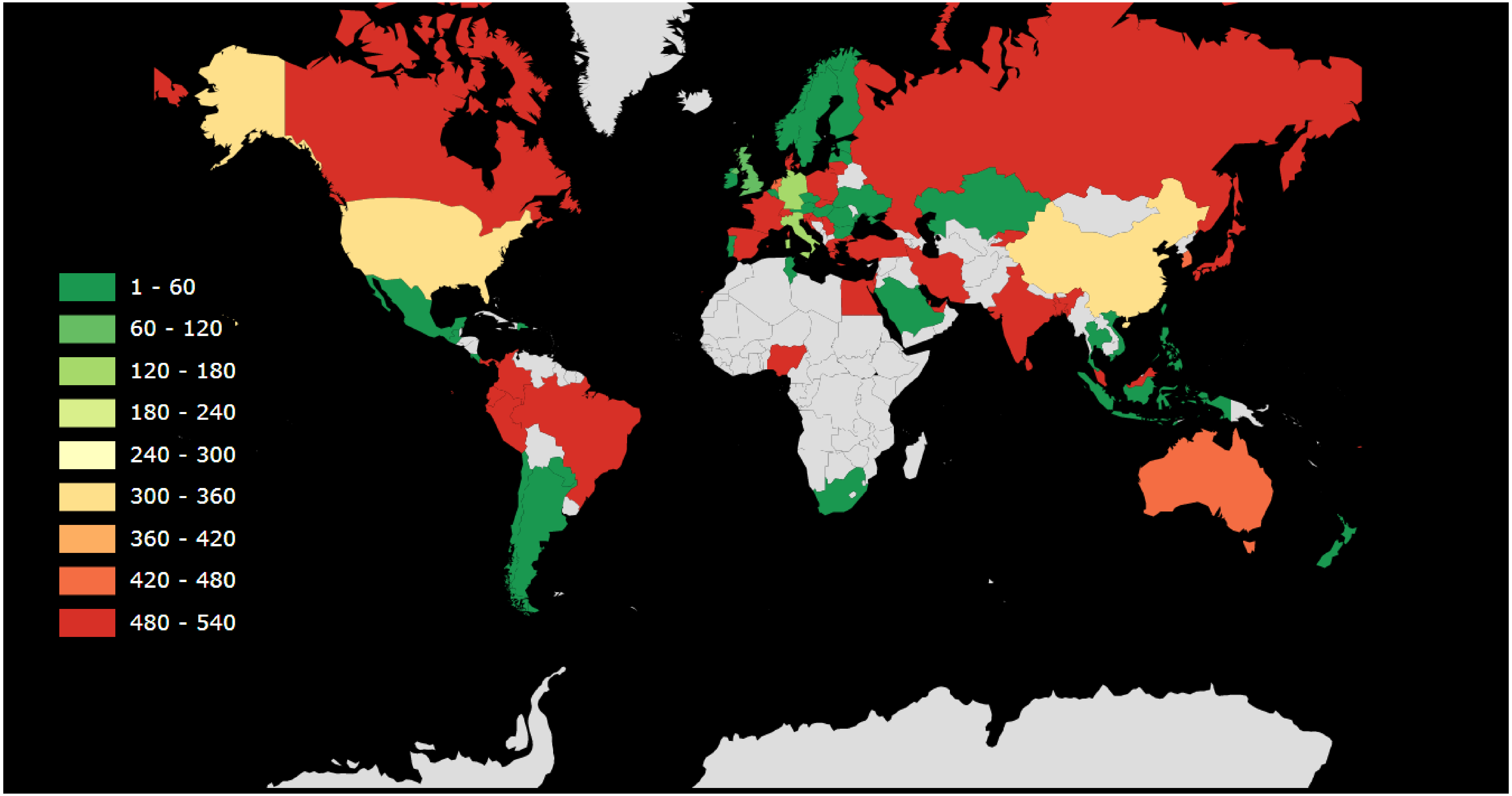
Botnet Connection by Location



Botnet Connection by Location



Botnet Connection by Location



Contact

Enrico Branca

Founder

enrico.branca@awebof.info