



UNIVERSITEIT VAN AMSTERDAM

SYSTEM AND NETWORK ENGINEERING

RESEARCH PROJECT 1

Detecting IP Hijacking Through Server Fingerprinting

Magiel VAN DER MEER
Eddie BIJNEN

May 2, 2014

Abstract

IP block hijacking has been possible for a long time. But as new attacks have been published, IP hijacking has become popular and the need to be able to detect this hijacking has become more urgent over the last years. Greenhost¹ has asked us to take a look at the possibility of finger printing of services to be used to detect changes on a server.

We have created a framework called Artculus that manages multiple endpoints and through the help of third-party tools examines a server. This data is stored in a database for later comparison. After a selectable interval the server is examined again and the information is compared with that stored in the database. If the data is inconsistent, this might indicate a possible IP hijacking attempt. Because changes do not have to be malicious, we have created an algorithm that assigns a weight to each probe and calculates the likelihood that a change is malicious. The sensitivity of this calculation can be configured to allow different levels of certainty before reporting to the user.

The result is that it has become far more work for an attacker to hijack an IP or subnet and stay undetected. Notifications can be set-up to trigger on detected anomalies and notify over various channels like e-mail.

¹<https://Greenhost.nl/about-us/>

Contents

1	Introduction	4
2	Research question	6
3	How does BGP hijacking work and what are the consequences?	7
3.1	Basic BGP operation	7
3.2	How does an attack work?	8
4	Which solutions exist in detecting BGP hijacking?	10
5	Which services can be probed for host fingerprint determination?	11
5.1	DNS Records	11
5.2	Mail services SMTP/IMAP/POP	12
5.3	Secure Shell	12
5.4	World Wide Web Server	12
5.5	Secure Sockets Layer	13
5.6	TCP/IP Fingerprinting	14
5.7	Traceroute	15
5.8	Not researched subjects	15
6	How to analyse the gathered fingerprints to detect an attack?	17
7	Results	19
7.1	Products	19
7.2	Sentinel difference	21
7.3	Findings	21
8	How can an attacker avoid detection?	24
8.1	DNS Records	24
8.2	SMTP/IMAP/POP	24
8.3	Secure Shell	25
8.4	World Wide Web Server	25
8.5	Secure Socket Layer	25
8.6	TCP/IP Fingerprinting	25
8.7	Traceroute	25
9	Conclusion	27
9.1	Future work	27
	References	29

LIST OF FIGURES

List of Tables

1	Articulus terminology	5
2	Services available for fingerprinting	11
3	Weight of available services	17
4	Notification thresholds	18
5	How to avoid fingerprinting?	24

List of Figures

1	Healty BGP example	7
2	Infected BGP example	9
3	Command and Control Sentinels	20
4	Variable output	21
5	Static output	22

1 Introduction

The Internet backbone has been built on trust and faith in the general good of the users. But this enormous growth of the Internet brought malicious participants with the intention to exploit other users and systems. The Border Gateway Protocol relies on this former trust. Once a machine is accepted into the BGP network there is very little stopping the machine from claiming an IP range and redirecting users to a malicious server or snoop on the user-generated traffic.

This is where this research project comes in. The researchers propose a network of sensors that will identify servers from different points on the Internet. When an attacker hijacks a subnet, it will be very difficult to mimic all the characteristics of the original servers inside the original subnet. Subtle changes can be detected by subsequently probing the at-risk targets.

Original request

To derive consistently functional and correct IP routing tables from a fluxing menagerie of BGP advertisements is not a matter of mere collection. Autonomous Systems employ filtering strategies to select the best available route to a given destination. Because the Internet is dynamic in its interconnectedness, routing changes are commonplace, and route filtering can only aspire to produce an ideal routing table, never with absolute certainty. This uncertainty opens a window to malicious route advertisements, in which a claim is made that a given IP subnet (victim subnet) is reachable via an AS with no legitimate claim to that subnet (malicious AS). If such malicious data is accepted into a routing table of an AS (victim AS) then a successful event of 'IP address hijacking' has occurred. At Greenhost, a hosting provider in Amsterdam, we have observed such an attack in the wild.

- *How can we analyze aggregated BGP data from around the world to identify subnets as the potential victims of IP hijacking?*
- *How can we subsequently probe these at-risk subnets to gain additional positive or negative evidence of hijacking?*

Greenhost is exploring possible answers to these questions through the development of analytical programs and distributed network probing agents.

Articulus

Articulus is a framework entirely developed by Eddie Bijnen and Magiel van der Meer, whom shall from here on be referred to as 'The IP hijacking researchers'. All the (Python) code written and the designed database (MySQL) is own work and had no practical involvement of employees of Greenhost. Articulus runs agent software on remote machines to enable them to receive tasks and report information back to the Command and Control (C&C) server. The

C&C server maintains a database with the different results of the scan tasks and the results of each client machine (Sentinel).

The C&C server periodically compares historical and global data with the newly posted information from the Sentinels. When unexpected changes are found, a notification can be sent. This alert contains the change that has been detected giving the user the ability to take action. A management panel gives access to the results, Sentinel management and available scans.

The most important terminology which might not be self-explanatory used throughout the report is shown in Table 1:

Used term	Description
Articulus	Creative Latin translation for 'finger'. Refers to the fingerprints that are being gathered.
Finger	The action of creating a fingerprint of a Node. Includes the external program needed for the fingerprint gathering.
Sentinel	(Virtual) machine somewhere around the world. Gathers the fingerprints of the Nodes.
Node	At-risk host. Services on these machines are fingerprinted by Sentinels.
Server	Controls Sentinels, assigns Fingers to Sentinels. Compares results from Sentinels and possibly notifies.

Table 1: Articulus terminology

All the code needed to continue development from the current state of Articulus can be found at <http://magiel.v-dmeer.nl/projects/RP1/articulus.zip>.

The package includes:

- SQL database
- Server side Python API
- Server side management API
- Management interface in HTML and JavaScript
- Client side Python

2 Research question

Several meetings took place between the researchers and Greenhost. In these meetings it has become clear that Greenhost is looking for alternate ways of identifying possibly compromised IP addresses of at-risk hosts. Greenhost has asked the research group to look into the possibility of using the technique of fingerprinting to identify compromised subnets or IP addresses.

The research question derived from the problem described on the previous page is as follows:

“How can we detect BGP IP hijacking by probing the at-risk subnets to detect suspected changes to hosts and subnets?”

The following sub-questions need to be answered to understand the problem:

1. How does a BGP hijacking attack work and what are the consequences?
2. Which solutions exist in detecting BGP hijacking?
3. Which services can be probed for host fingerprint determination?
4. How can the gathered fingerprints be analyzed to detect an attack?
5. How can an attacker avoid detection?

3 How does BGP hijacking work and what are the consequences?

3.1 Basic BGP operation

BGP (Border Gateway Protocol) is the routing protocol used on the Internet. It exchanges routing and reachability information between autonomous systems (AS). In contradiction to most other routing protocols, BGP decides which routes end up in the routing table based on paths lengths, network policies and/or rule-sets. The latter is configurable by a network administrator, enabling organizations to discriminate connections and, for example, favor a slow but cheap connection over an expensive fast one.

An example of multiple parties peering with each other is shown in Figure 1. Multiple parties (AS_{xx}) are interconnected and provide access to, in this example, Facebook. A client accesses the Internet via AS60. AS60 knows it can reach Facebook over AS20 and AS30 because they announce the Facebook IP space 31.13.24.0/21².

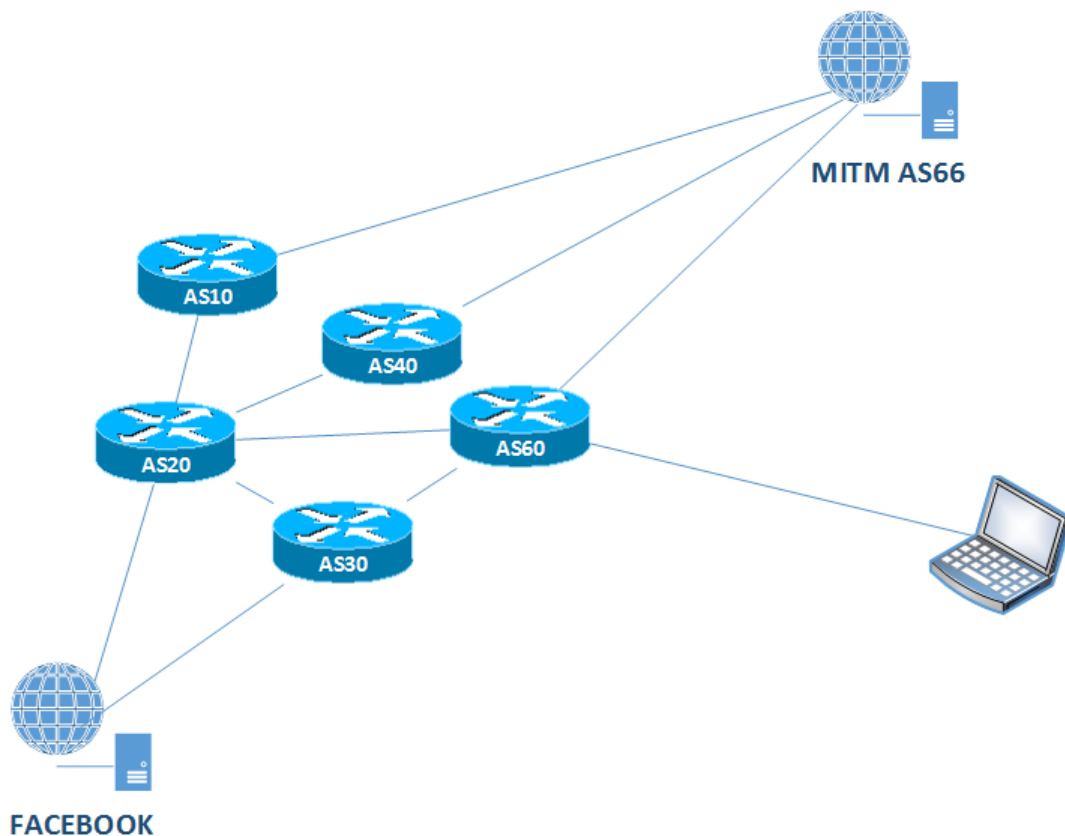


Figure 1: Healty BGP example

²We use this as an example. The practical space used by Facebook is much bigger.

A more detailed explanation of BGP can be found at [5]. This report will only cover the basics needed for the reader to understand BGP for this report.

Two situations leading to problems can be described; configuration mistakes and abuse. An administrator can make one simple typing mistake and, for example, announce 13.37.x.x/16 instead of 13.38.x.x/16 thus attracting all the traffic destined for 13.37.x.x/16. A related and practical example of this can be found at [6] where is described how in the Pakistani government tried to block Youtube for Pakistan but sink holed Youtube traffic from around the world.

The other situation is abuse by inserting prefixes the attacker doesn't own and sinkholing, log or manipulate the victim's traffic. The inserted subnet is mostly called the 'victim subnet' or 'victim host'. The sending host is referred to as 'malicious router'. Examples of abuse can be found at [7, 8].

3.2 How does an attack work?

An attacker can take over the connection from the client to Facebook. If the attacker announces the prefixes 31.13.24.0/22 and 31.13.28.0/22 from AS66, those routes take precedence over the original 31.13.24.0/21 route because they are more-specific. AS66 then attracts all the traffic and can execute various malicious actions on the traffic. This example is shown in Figure 2.

Five options of BGP hijacking are described by Xin Hu and Z. Morley Mao in [2, page 5]. Which BGP attack is used does not matter for this research. The example given above will be used in this research as a reference model for all five possible attacks.

A bit more technical

One cannot claim one or multiple IP address by just configuring an interface with the address(es) and expect it to be reachable. The local AS will accept the IP if configured in the right way, but it cannot be found outside the local AS. This is where Border Gateway Protocol (BGP) comes in play. Edge routers will communicate with each other using (external-) BGP. Routers announce their local and received prefixes to the neighboring routers. If the neighboring routers accept the received prefix, it will include the prefix in its update to his neighbors. This mechanism forms routes from a random point [A] to every possible point [B] on the entire Internet.

In the current system no security is applied. Routers trust each other unconditionally and apply little to no verification of received routes. Several projects are currently in various stages of development to add better security to the BGP protocol but none are deployed wide enough to make a difference and prevent negative consequences from misconfiguration and abuse.

Because of the lack of proper security, any router that has direct access to BGP speaking nodes and is sufficiently trusted by its neighbors can claim to have a route to a victim subnet at a lower cost. Because the malicious route is inserted with a smaller prefix, the original traffic is

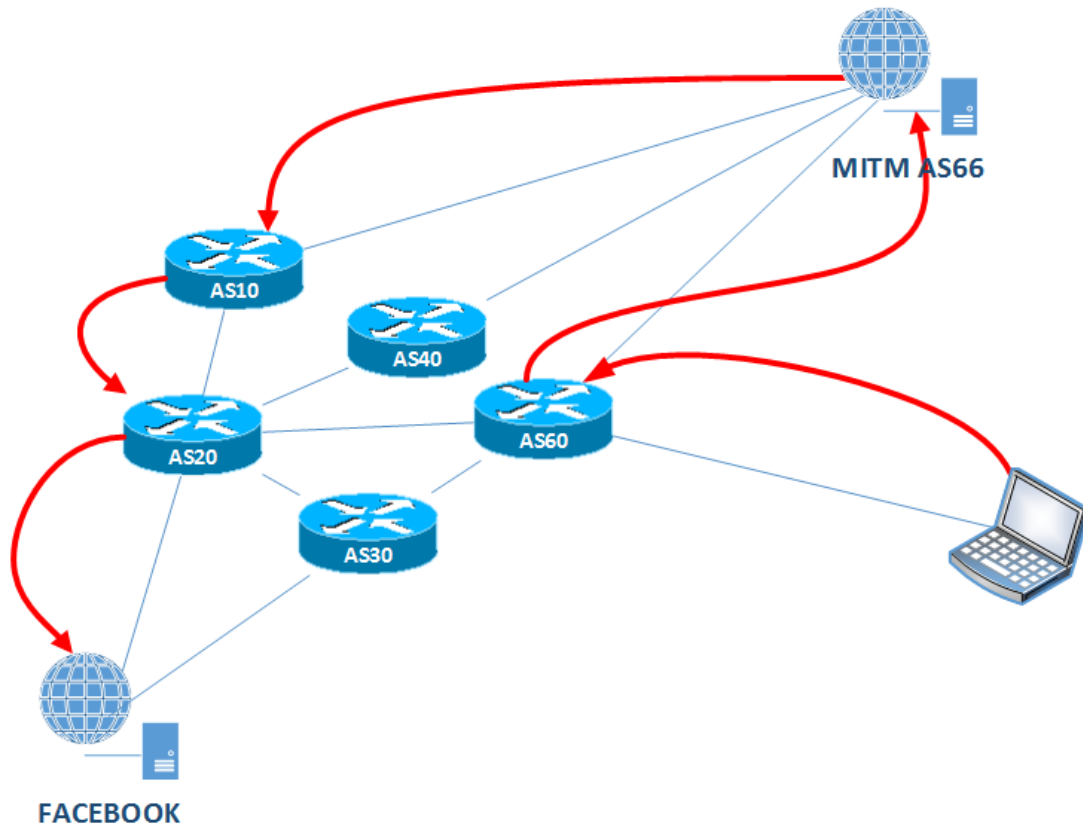


Figure 2: Infected BGP example

being sent to the attacker. The complete decision process involved in route selection is explained in RFC 4271 ³.

In the example above there is no route from the attacker to the original destination. If desired by the attacker the traffic can be forwarded to the original destination by becoming a (transparent) proxy by injecting the victim subnet with AS10 and AS20 as hops in the route. AS10 and AS20 will not accept the route since they think it already passes them. This leaves a route open from the attacker to the original destination. The attacker may also forward the traffic without proxying. In that case it needs to insert a route from the destination to the client which passes the attacker. This route is not necessary if the attacker does not want the returning traffic.

³<http://www.ietf.org/rfc/rfc4271.txt>

4 Which solutions exist in detecting BGP hijacking?

iSPY

iSPY [3] is a proposed method of detecting IP hijacking suggested by Zheng Zhang, Ying Zhang, Y. Charlie Hu, Z. Morley Mao and Randy Bush. They propose probing the IP subnet continually. When a subnet gets hijacked they assume that the route will spread to a relative large portion of the routers forming Internet, This means iSPY probes will no longer reach the legitimate network at which point alarm bells go off. The theory behind the paper seems sound however it requires implementation on each AS that wants to be protected. The researchers were unable to find an implementation of this paper.

Atlas by RIPE

Atlas is a sensor network that has been created by RIPE. It relies on volunteers supplying resources of which several checks can be run. Currently the system supports: DNS resolving, ping, traceroute and grabbing certificates of SSL connections. It however lacks the ability to compare results and only fingerprints SSL. It also requires one to volunteer resources to gain credits which are needed to initiate scans. There is no possibility to add additional scan types or software.

<https://atlas.ripe.net/>

BGPmon

BGPmon is a product that analyses BGP route updates and creates a database how Autonomous Systems are interconnected to one another and announce which subnets. They can make a guess when a subnet changes from AS if they are still going to the same destination. It however does not verify that an IP subnet is hijacked within another subnet and does not use fingerprinting when determining if it is a legitimate change.

<http://www.bgpmon.net/>

Cyclops

Cyclops is a similar product as BGPmon, it analyses BGP route updates. Being able to send alerts when an AS or upstream AS changes. It also looks at which AS publishes which subnet. It does not verify if an IP address changes running services within the AS and does not use fingerprinting when determining if it is a legitimate change.

<http://cyclops.cs.ucla.edu/>

5 Which services can be probed for host fingerprint determination?

Fingerprinting is the term used to identify systems based on certain characteristics. In the case of a real human it is the common conception that each fingerprint is unique. Servers do not have fingers but they do have fingerprints. Each service has specific characteristics: how it responds to certain commands, what features it provides and how it deals with errors.

Being able to remotely detect the version of software is considered unwanted by some and therefore the version of the software is typically not provided. Different versions of software do however respond uniquely to slightly different incoming network packets. Based on these differences, one can determine what software version is running. These unique characteristics form a fingerprint of a server which can be compared from different vantage points and historical information.

Different services needs to be probed in different ways. The following subsections will discuss the services mentioned in Table 2, describe which details are usable to form the fingerprint of the service and how these details are obtained in a technical perspective.

Service description	Short fingerprint description
DNS records	Matches the resolved IP addresses with historical data
Mail services	SMTP header comparison IMAP response comparison
Web services	Web server response to various requests
Secure Socket Layer	Changes in the certificate checksum
Traceroute	The traversed IP path on the Internet and changes in the response time
Open ports	Open ports on at-risk host
TCP/IP characteristics	Host specific TCP responses

Table 2: Services available for fingerprinting

5.1 DNS Records

By resolving a given DNS record at a given list of public DNS servers and comparing the results, it is possible to determine if a DNS server is hijacked. If probed from around the globe, a historical view can be build. If an IP address returned by one or more resolving DNS servers differs from the historical view, the DNS server might be victim of an IP hijacking attack (or otherwise compromised). IP addresses returned might differ based on geographical location.

5.2 Mail services SMTP/IMAP/POP

Mail protocols like SMTP, IMAP and POP are old protocols which are still used for sending and receiving email. The software in use is often returned at the start of the negotiations between the server and client. If the returned version is actually the correct version is of less concern. The fact that it remains the same or changes gives enough information to trigger on or not.

More and more mail services start to use STARTTLS when communication with other mail services enabled nodes. This gives an excellent fingerprint in the form of a certificate. Though the SHA-1 hash of the certificate cannot be considered unique (snake-oil certificates on not configured SMTP servers for example), changes in the certificate can be a sign of hijacking. An attacker would be able to mimic the public certificate when he has access to the private key. One can assume that when the attacker has the private key, he can reach his malicious goals in other manners as well. This goes for all the certificate-based fingerprints. It would also be easy for an attacker to mimic the snake-oil certificates. This is a nice incentive to configure SSL correctly on an at-risk host.

Example of fingerprint for SMTP:

```
25/tcp open smtp      Postfix smtpd
smtp-commands: haarlem.v-dmeer.nl, PIPELINING, SIZE 10240000,
VERFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
```

5.3 Secure Shell

SSH is a network protocol for secure communications between hosts over an insecure network. The underlying security relies on Secure Socket Layer (SSL). Again the certificate returned by the SSH server can be probed for change or differences from different vantage points.

Example Fingerprint for SSH:

```
22/tcp open  ssh      OpenSSH 6.4p1 Debian 2 (protocol 2.0)
ssh-hostkey:
1024 d7:a5:fc:ee:65:30:73:80:42:72:50:19:0a:1d:1e:0f (DSA)
2048 a8:fe:3a:70:7f:ee:a1:0e:89:b2:35:e7:16:1a:77:11 (RSA)
```

5.4 World Wide Web Server

The HTTP protocol and related web servers have been around for a long time. People have added many add-on programs to add functionality or ease content changes to websites. Each of these programs can be detected and identified by their version. Because not all of these programs update automatically, new installs will likely have newer version. These web programs can be

5 WHICH SERVICES CAN BE PROBED FOR HOST FINGERPRINT DETERMINATION?

probed and compared from geographical perspective and historical view. If HTTPS is enabled on the web server the contents of Section 5.5 can be applied as well.

Example fingerprint for a web page:

```
https://Greenhost.nl/ [200]
All-in-one-SEO-Pack[2.1.2]
Apache[2.2.16]
Cookies[PHPSESSID,showpromo]
Country[NETHERLANDS] [NL]
HTML5
HTTPServer[Debian Linux] [Apache/2.2.16 (Debian)]
IP[213.108.104.135]
jQuery[1.10.2]
MetaGenerator[WordPress 3.8]
Script[text/javascript]
Title[Greenhost | Duurzame webhosting]
WordPress[3.8]
UncommonHeaders[x-pingback,link]
x-pingback[https://Greenhost.nl/xmlrpc.php]
```

The fingerprint generated as above does not remain the same on each web site over time. For example, <http://tweakers.net> rotates its HTML <title>tags frequently. An extended test could be implemented in Articulos omitting predefined rows like the ‘Title’.

5.5 Secure Sockets Layer

The idea of fingerprints has been implemented at its core. Identifying the other side is a crucial part, SSL does this by presenting an certificate. A SHA-1 hash of this certificate is created and stored. If there are any changes to the certificate because it was renewed or someone is presenting a false certificate, the system will notice this. SSL has been implemented as a security layer in many other protocols which grants the ability to fingerprint these services with perfect precision.

Example fingerprint for SSL Connection:

```
443/tcp open  http      syn-ack nginx 1.4.4
http-methods: No Allow or Public header in OPTIONS response
(status code 400)http-title: 400 The plain HTTP request was
sent to HTTPS port ssl-cert: Subject: commonName=*.pretwolk.nl
/organizationName=pretwolk.nl/stateOrProvinceName=NH/
countryName=NL/localityName=Duckstad/
organizationalUnitName=pretwolk.nl/emailAddress=
contact@pretwolk.nlIssuer: commonName=pretwolk.nl/
```

5 WHICH SERVICES CAN BE PROBED FOR HOST FINGERPRINT DETERMINATION?

```
organizationName=pretwolk.nl/stateOrProvinceName=NH/  
countryName=NL/organizationalUnitName=pretwolk.nl/emailAddress  
=contact@pretwolk.nl  
Public Key type: rsa  
Public Key bits: 4096  
Not valid before: 2013-06-23T12:13:10+00:00  
Not valid after: 2015-06-23T12:13:10+00:00  
MD5: 9e1a 074d adfe cf68 44de 965f d45a df51  
SHA-1: 5df5 92e2 6ff9 4136 145a 12bb dc4b 4815 3328 8d1d
```

5.6 TCP/IP Fingerprinting

The TCP/IP side of the network packets are handled by the operating system. Each operating system has a specific way of responding to abnormal packets. The sentinel sends 6 specially crafted packets that identify the system by the way they are handled. More information about how these specific packets are crafted can be found at [10].

The operating system can be determined from the TCP/IP response as well as the uptime of the host. The uptime can be calculated by looking at TCP/IP timestamps. By sending multiple packets it can be detected how fast the TCP/IP timestamps increments and by knowing the current value a guess can be made of the current uptime. The current uptime should be not lower than the previous uptime and not significantly higher than the sum of the previous uptime and the check interval.

Example TCP/IP Fingerprint:

```
Device type: general purpose  
Running: Linux 3.X  
OS CPE: cpe:/o:linux:kernel:3  
OS details: Linux 3.0 - 3.1  
  
TCP/IP fingerprint:  
OS:SCAN(V=6.00%E=4%D=1/27%OT=22%CT=%CU=%PV=N%G=N%TM=52E63B01%P  
=x86_64-unknoOS:wn-linux-gnu)SEQ(SP=102%GCD=1%ISR=10D%TI=Z%II=  
I%TS=8)OPS(O1=M5B4ST11NW6%OS:O2=M5B4ST11NW6%O3=M5B4NNT11NW6%O4  
=M5B4ST11NW6%O5=M5B4ST11NW6%O6=M5B4ST11OS: )WIN(W1=3890%W2=3890  
%W3=3890%W4=3890%W5=3890%W6=3890)ECN(R=Y%DF=Y%TG=40%OS:W=3908%  
O=M5B4NNSNW6%CC=Y%Q=)T1(R=Y%DF=Y%TG=40%S=0%A=S+%F=AS%RD=0%Q=)T  
2(ROS:=N)T3(R=N)T4(R=Y%DF=Y%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)U  
1(R=N)IE(R=Y%DFIOS:=N%TG=40%CD=S)  
  
Uptime guess: 159.763 days (since Tue Aug 20 18:36:24 2013)  
TCP Sequence Prediction: Difficulty=258 (Good luck!)
```

5 WHICH SERVICES CAN BE PROBED FOR HOST FINGERPRINT DETERMINATION?

```
IP ID Sequence Generation: All zeros
Final times for host: srtt: 1792 rttvar: 275 to: 100000
```

5.7 Traceroute

A traceroute can be executed over TCP and UDP but mostly ICMP is used. All three protocols give much of the same output. But where UDP and ICMP generally get blocked at the first filtering router or load balancer, TCP used on port 80 gets through these devices can give a small peak inside the network. When changes are detected, an alert will be send. Because legitimate changes are expected different levels of reporting are available. More information on this subject can be found in Section 6.

```
root@sentinelafrica:~# traceroute -n -T -p 80 85.17.176.216
traceroute to 85.17.176.216 (85.17.176.216), 30 hops max, 60
 1 197.85.186.1 0.769 ms 0.937 ms 0.909 ms
 2 196.41.144.34 16.950 ms 16.970 ms 17.558 ms
 3 196.28.178.65 0.832 ms 1.119 ms 1.150 ms
 4 196.28.178.1 1.133 ms 1.145 ms 1.256 ms
 5 197.84.5.225 146.975 ms 197.84.5.226 146.977 ms
 6 197.84.4.197 147.113 ms 197.84.4.32 146.386 ms
 7 176.67.177.131 148.834 ms 148.845 ms 148.799 ms
 8 176.67.177.162 151.219 ms 151.155 ms 151.188 ms
 9 176.67.177.227 148.645 ms 195.66.225.56 152.760 ms
10 176.67.177.133 148.167 ms 147.028 ms 146.973 ms
11 195.66.225.56 153.166 ms 195.66.225.100 159.864 ms
12 31.31.32.69 161.598 ms 62.212.80.74 160.719 ms
13 85.17.176.216 178.371 ms 191.155 ms 181.188 ms
root@sentinelafrica:~#
```

5.8 Not researched subjects

Some available systems have not been researched as possible fingerprint gathering applications.

SSLScan SSLScan can probe SSL enabled nodes for available cipher suites in the SSL stack. The benefit would be the possibility to check for downgrade attacks. However, SSLScan is slow and currently not up to date with TLS1.2.

BGP Looking Glass BGP Looking Glass would show the path through the different AS's and allow monitoring of BGP updates. However, this method of detecting has been done by

5 WHICH SERVICES CAN BE PROBED FOR HOST FINGERPRINT DETERMINATION?

BGPmon and Cyclops, both providing a free and available version. Time constraints do not allow to pursue this avenue.

FPDNS FPDNS from 'DBS Software fingerprinting' would give the ability to detect the version and software used. However, it was incorrectly thought that this software was out of date during the development stages and was not included. By resolving DNS records of the domain to IP, Artculus can already detect false DNS responses.

6 How to analyse the gathered fingerprints to detect an attack?

Not all changes detected on a node are a clear sign of a hijacking attempt. Therefore, each finger has an associated weight. This weight is used in a formula to get a number. This number represents the probability a node is hijacked. In Table 3 is shown which default weight is assigned with a specific finger. The Artculus application allows for changing this.

Finger	Weight
DNS record	150
Mail service telnet EHLO	100
Mail service SSL Certificate	500
SSH service RSA host key	500
Web service	100
Web service SSL cert SHA	500
TCP/IP	100
TCP/IP uptime guess	250
Traceroute hops	250

Table 3: Weight of available services

The threshold limit needs to be relative to the amount of fingers being performed on that specific node. This means that the threshold limit equals of the sum of the weight of the enabled fingers.

Equation 1 calculates the probability a node is hijacked. Assuming all the possible fingers are enabled for a node, the total number of default points for detectable changes on a node is 2000.

$$Points = \left(\frac{weight}{total} \right) * 100 \quad (1)$$

If Artculus detects a certificate change on a random node, the equation (shown in Equation 2) will return 25 as the probability a node's IP has been hijacked. A probability of 25 will trigger a notification, as defined in Table 4. This assumes the host has all the possible fingers configured.

$$25 = \left(\frac{500}{2000} \right) * 100 \quad (2)$$

When a node only has 'DNS record protection' enabled, the values in the equation change as shown in Equation 3. Again this is triggering a notification.

$$100 = \left(\frac{150}{150} \right) * 100 \quad (3)$$

6 HOW TO ANALYSE THE GATHERED FINGERPRINTS TO DETECT AN ATTACK?

When a node has ‘DNS record protection’, ‘SSH service RSA host key’ and ‘Mail service SSL Certificate’ enabled, the threshold limit would be 1150. In this case, the sole change of a DNS record is enough to trigger a notification to a System Administrator but the End User doesn’t receive this. Equation 4 shows this.

$$13 = \left(\frac{150}{1150}\right) * 100 \quad (4)$$

Table 4 contains the lower limits necessary to trigger a notification. These values can be configurable in the Articulus system to tune the needs of the user.

Notification level	Value
Paranoid	1 point
System Administrator	10 points
End User	20 points

Table 4: Notification thresholds

7 Results

The IP Hijacking research group was asked to create an initial framework providing the ability to probe single hosts from multiple vantage points and compare the results. Initial testing during the development has shown that the product is capable of achieving the intended goal but further development is necessary.

7.1 Products

To test the developed system, some constraints have been determined. In this section the following tests have been conducted:

- Sentinel Command and Control
Can the Sentinel be added to the C&C server? Can the C&C server manage the added Sentinels at the same time? Are basic statistics like 'last seen' and the local Unix user running the agent software visible in the C&C server?
- Proof of concept
Is the gathered data usable in such a way that the intended goal can be achieved?
- Sentinel difference
Does the agent software run on a variety of Linux distributions and work with different versions of tools listed on page 28.

7.1.1 Sentinel Command and Control

The Artculus C&C management interface and system API are written in Python and run on an existing web server with Python support. The multi-threaded nature of a web server allows for interaction with multiple Sentinels at the same time. As shown in figure 3, multiple Sentinels are active in the system, are manageable and Sentinel specific data is shown.

7.1.2 Sentinel agent

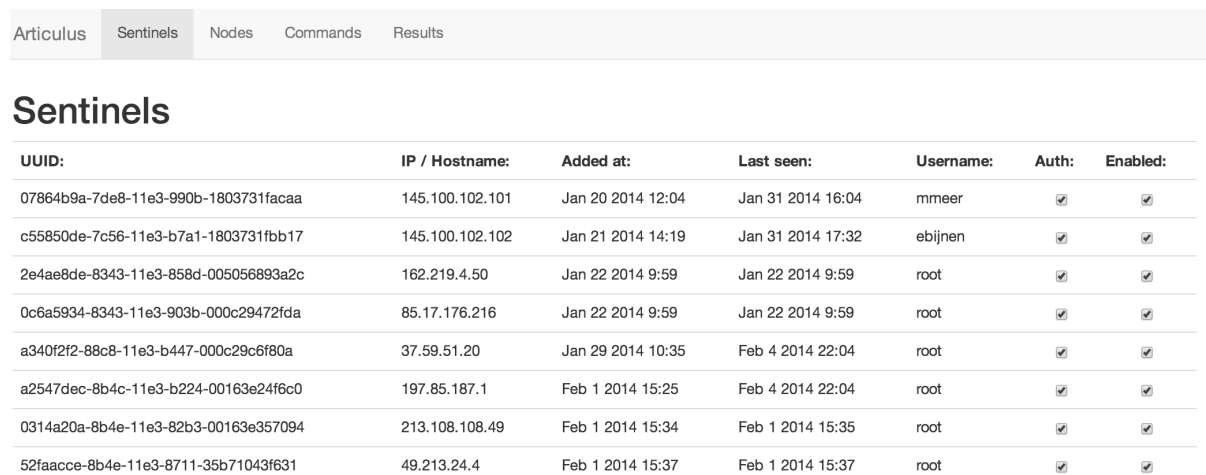
A Sentinel generates a unique UUID which is used as an identifier in communication with the C&C server. This UUID is saved in `/.sentinel/uuid` and only generated if this file does not exist. The first time a Sentinel announces itself to the C&C server, the server considers the Sentinel 'unauthorized' and 'disabled'. An administrator can 'enable' and 'authorize' the Sentinel.

The Sentinel requests its commands to be executed from the C&C server and starts simultaneously using the Python package `clThread`. This package enables multi-threading in Python scripts. The results are posted to the C&C server individually upon completion of the task.

All the communication is encrypted using TLS. The *Curl* package in Python can be enforced to use *certificate – pinning* and verify the remote peer (the C&C server). The CA’s certificate needs to be present on the Sentinel to verify the server.

7.1.3 Result interface

In the current state of Articulo, the entire output of the scanning tool *Whatweb* is hashed and this hash is saved for comparison. The results of one particular web site are shown in figure 4. In this case the hashes are not comparable in history and between Sentinels because too much information is variable to different Sentinels and changes in time. In the figures, the hashes are uniquely colored by using the first six characters of the hash as the HEX value for the CSS color element.



UUID:	IP / Hostname:	Added at:	Last seen:	Username:	Auth:	Enabled:
07864b9a-7de8-11e3-990b-1803731facaa	145.100.102.101	Jan 20 2014 12:04	Jan 31 2014 16:04	mmeer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
c55850de-7c56-11e3-b7a1-1803731fbb17	145.100.102.102	Jan 21 2014 14:19	Jan 31 2014 17:32	ebijnen	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2e4ae8de-8343-11e3-858d-005056893a2c	162.219.4.50	Jan 22 2014 9:59	Jan 22 2014 9:59	root	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
0c6a5934-8343-11e3-903b-000c29472fda	85.17.176.216	Jan 22 2014 9:59	Jan 22 2014 9:59	root	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
a340f2f2-88c8-11e3-b447-000c29c6f80a	37.59.51.20	Jan 29 2014 10:35	Feb 4 2014 22:04	root	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
a2547dec-8b4c-11e3-b224-00163e24f6c0	197.85.187.1	Feb 1 2014 15:25	Feb 4 2014 22:04	root	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
0314a20a-8b4e-11e3-82b3-00163e357094	213.108.108.49	Feb 1 2014 15:34	Feb 1 2014 15:35	root	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
52faacce-8b4e-11e3-8711-35b71043f631	49.213.24.4	Feb 1 2014 15:37	Feb 1 2014 15:37	root	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 3: Command and Control Sentinels

7.1.4 Proof of concept

During the development, one finger was chosen to develop to such a state that it was usable to generate results and contribute to the proof of concept. The tool used for this was *Whatweb*⁴. *Whatweb* scans a webserver and return the results as shown in subsection 5.4.

The results from this tool were variable. In example, a web server sends a different response for different user-agents, sites change their HTML title-tag over time and HTTP load balancing might be in place all resulting in a different output and thus a different hash.

To overcome the problem described above, the finger could be extended to interpret the output shown in subsection 5.4 line-by-line and compare the results individually. Now, for

⁴<http://whatweb.net/>

Articulus	Sentinels	Nodes	Commands	Results
Results				
<div style="background-color: black; color: white; padding: 2px;"> 34 213.230.154.20 tweakers.net </div>				
<div style="background-color: #f0f0f0; padding: 2px;"> └ 21 (37.59.51.20) </div>				
└ fd778d4864e313cd8b0f0e44916ed2ac7f24c88f 2014-02-04 22:06:59				
└ a8679202b885feb3f3b4bdbeb54bbe88108ecdb 2014-02-04 22:05:55				
└ fd778d4864e313cd8b0f0e44916ed2ac7f24c88f 2014-02-04 22:04:51				
└ 60d9c9f8b073a4388541fc034167c5f86183236f 2014-02-04 22:03:48				
└ 1cdd9f8b2e2571452bce54fe8e933ab1a4cd94c7 2014-02-04 22:02:44				
<div style="background-color: #f0f0f0; padding: 2px;"> └ 22 (197.85.187.1) </div>				
└ fd778d4864e313cd8b0f0e44916ed2ac7f24c88f 2014-02-04 22:06:26				
└ fd778d4864e313cd8b0f0e44916ed2ac7f24c88f 2014-02-04 22:05:20				
└ fd778d4864e313cd8b0f0e44916ed2ac7f24c88f 2014-02-04 22:04:14				
└ 60d9c9f8b073a4388541fc034167c5f86183236f 2014-02-04 22:03:08				
└ 1cdd9f8b2e2571452bce54fe8e933ab1a4cd94c7 2014-02-04 22:02:01				

Figure 4: Variable output

example, the version of the content management system, PHP version or jQuery version can be compared historically and between responses of multiple Sentinels. By assigning weights to the individual lines of data, a reliable suggestion can be made about suspicious changes on a server.

7.2 Sentinel difference

As a result of the development of the Sentinel agent software, there can be concluded that output of tools differ between minor versions and this influences the way Articulus deals with the data.

Figure 5 shows a static site which is completely identical to both Sentinels. But the first Sentinel runs a slightly different version of the tool Whatweb and this returns the same results in a different order. This changes the hash and frustrates the comparison of results between Sentinels.

7.3 Findings

The researchers have primarily focused on creating a proof of concept that is able to detect changes on hosts and have gathered little data. The data that has been gathered has shown that detection of changes on hosts is possible by comparing the results. Changing the test environment triggered notifications in Articulus as expected. For instance, replacing the web server software from Apache to Nginx while still serving an identical website does change the footprint of the server and this is detected by Articulus.

7 RESULTS

35	194.71.107.15	thepiratebay.se
21 (37.59.51.20)		
...	6dfa5a27ed70da9b640cf6bb186c342eecf7b1bf	2014-02-04 22:06:59
...	6dfa5a27ed70da9b640cf6bb186c342eecf7b1bf	2014-02-04 22:05:55
...	6dfa5a27ed70da9b640cf6bb186c342eecf7b1bf	2014-02-04 22:04:51
...	6dfa5a27ed70da9b640cf6bb186c342eecf7b1bf	2014-02-04 22:03:47
...	6dfa5a27ed70da9b640cf6bb186c342eecf7b1bf	2014-02-04 22:02:43
21 (37.59.51.20)		
...	6dfa5a27ed70da9b640cf6bb186c342eecf7b1bf	2014-02-04 22:06:59
...	6dfa5a27ed70da9b640cf6bb186c342eecf7b1bf	2014-02-04 22:05:55
...	6dfa5a27ed70da9b640cf6bb186c342eecf7b1bf	2014-02-04 22:04:51
...	6dfa5a27ed70da9b640cf6bb186c342eecf7b1bf	2014-02-04 22:03:47
...	6dfa5a27ed70da9b640cf6bb186c342eecf7b1bf	2014-02-04 22:02:43
38	145.100.96.70	os3.nl
21 (37.59.51.20)		
...	f90957fecc32db0492935565686cd9372ddb69fb	2014-02-04 22:06:59
...	f90957fecc32db0492935565686cd9372ddb69fb	2014-02-04 22:05:55
...	f90957fecc32db0492935565686cd9372ddb69fb	2014-02-04 22:04:52
...	f90957fecc32db0492935565686cd9372ddb69fb	2014-02-04 22:03:48
...	f90957fecc32db0492935565686cd9372ddb69fb	2014-02-04 22:02:44
22 (197.85.187.1)		
...	c8a45aaa08d0131a42417070fd1a6e961b27d3a0	2014-02-04 22:07:34
...	c8a45aaa08d0131a42417070fd1a6e961b27d3a0	2014-02-04 22:06:26
...	c8a45aaa08d0131a42417070fd1a6e961b27d3a0	2014-02-04 22:05:20
...	c8a45aaa08d0131a42417070fd1a6e961b27d3a0	2014-02-04 22:04:14
...	c8a45aaa08d0131a42417070fd1a6e961b27d3a0	2014-02-04 22:03:08

Figure 5: Static output

7.3.1 Routing changes

Adding an extra router in the path is detected as well. Network paths do change regularly for valid reasons and thus not all changes are explicitly suspicious. The act of BGP hijacking is rare and typically on a small scale. During the research project only four Sentinels around the world were available for a short amount of time. Because of this no real data could be gathered showing a route change indicating a possible BGP hijacking attempt.

7.3.2 User Agent inconsistencies

During the testing of the chosen tool to compare web sites, an inconsistency was found. While the same version of 'Whatweb' was used on both Sentinels, the underlying version of Ruby differed. This created a minor difference in the output of 'Whatweb' and thus resulting in a different hash while the examined content was equal. The difference appeared to be the order in which the content of the returned user-agent was handled and given as output. A possible solution is already proposed in this section.

7.3.3 Dynamic Websites

Several of the sites that have been tested were dynamically updating their title with statistical information. For example, *http://tweakers.net* shows the number of visitors and how many are logged in at that moment. This made the testing of changes on the webpage level difficult as it is always changing. As shown in figure 4.

8 How can an attacker avoid detection?

The difficulty of avoiding detection differs between fingerprinting methods. Each fingerprinting method requires their own way of fooling the sentinel. It is possible to generate the right reply so IP Hijacking isn't detected but it requires a lot of resources and non-public information. A list of possible evasion attempts is shown in Table 5.

Service	Possibilities of hiding
DNS	Alternate replies for Artculus and victim. Forward to original.
DNSsec	Access to upstream TLD and alternate replies for Artculus and victim. Forward to original.
Mail services	Same software and modules enabled. MITM. Forward to original.
Web server	Same software and modules enabled. MITM. Forward to original.
TLS services	Access to the certificate and private key. Possible downgrade attack. Forward to original.
Traceroute	Needs to be directly connected to a router in the original path.
Open ports	Port scan original and set up all of the above for enabled services.
TCP/IP characteristics	Run Nmap and running appropriate kernel modules.

Table 5: How to avoid fingerprinting?

8.1 DNS Records

To change the DNS records an attacker either need write access to the existing name servers or hijack the IP subnet that the name server is in. At which point one would need to deliberately return different results to the Sentinels then to other hosts. Changing the IP without this distinction will be detected by the Sentinels.

8.2 SMTP/IMAP/POP

The insecure mail services are relatively easily faked. One only needs to connect to the legitimate mail servers and imitate the replies. Luckily many companies have implemented the secure versions of SMTP/IMAP/POP. Because of the use of SSL and certificates the attacker would

need access to the private key that is stored on the server. If an attacker would have this level of access there would be little need for an IP hijacking attack. Attempting a downgrade attack by disabling the secure versions would be detected as well.

8.3 Secure Shell

To imitate SSH we require the private key. Again if an attacker would have this level of access there would be little need for an IP Hijacking attack. The attacker can however leave the traffic untouched and forward it to the original server leaving the fingerprint intact.

8.4 World Wide Web Server

Avoiding detection for insecure HTTP is relative easy. By using a transparent proxy that presents the original web page and forwards every query. The hijacker can stay undetected and any data transmitted over insecure HTTP can be captured. However when attempting to present a web page that is not the original, great effort must be put into using the same versions of the installed software and plug-ins.

8.5 Secure Socket Layer

Without access to the private key the attack will not be able to imitate the server. However, if the server operators are not solely running TLS 1.2 a downgrade attack is possible. The severity of this depends on the supported cipher suites that are allowed on the server. The attacker may also leave the SSL traffic intact and forward it to the server and rely on the non-secure traffic, leaving the SSL signature intact.

8.6 TCP/IP Fingerprinting

The TCP/IP fingerprint can be manipulated by the use of a special kernel module called 'Fingerprint Fucker'. And the uptime of a machine can also be changed by the use of a program called 'UptimeFaker' Both are not in public active development. But they do show that it is possible to alter the fingerprint. It would require effort and knowledge to maintain these modules and set them up correctly, but in theory, it could be done.

8.7 Traceroute

Traceroute works by sending packets with an increasing TTL. The dropping router responds with a 'time exceeded' message. The only way to fake this correctly is to be directly connected to a router that is in the original path. If the router sends an BGP update and the packets take a route with legitimate routers outside the original route, they will respond with their IP

and be detected by the Sentinels. If the attacker is connected to a router that is in the original route, he can fake and withhold the time exceeded messages. This will stay undetected by the sentinel but limits the amount of users that can be rerouted.

9 Conclusion

BGP IP hijacking attempts as described in section 3.1 pose threats to at-risk hosts or subnets. Changes to these hosts caused by BGP IP hijacking can be detected by creating a fingerprint of them from as many, globally spread out, points of the Internet. The application developed and described in section 1, Artculus, makes it exponentially harder to impersonate a server by probing predefined services on the at-risk hosts. An attacker would need to hide his attempts not just from human eyes but also from a system specifically built to detect the smallest changes in the way an at-risk host responds to requests. By probing services like IMAP, SMTP, HTTP, SSL enabled applications or probe for TCP characteristics the application can make an educated guess if a server is hijacked.

Hiding from Artculus is only possible if the Sentinels are not affected by BGP IP hijacking or by cloning the characteristics of the original server. The attacker can test the original server and mimic these on his own server. The difficulty for the attacker lies in the fact that he does not know which services and/or protocols are available and being fingerprinted and thus needs to take the risk of being detected. Although it is not impossible, the required access to the original server and effort make it very impractical to attempt hiding from Artculus.

9.1 Future work

Artculus

Currently there is no user management, The Artculus management interface has no access control system thus no distinction is made in user access level. Every user that can reach the site can alter all available scans. It is advised to integrate an authentication and authorization system in the Artculus management interface before running it in a production environment.

Creating a program which scans subnets and adds new nodes found in this subnet as a node to the existing Artculus system.

Write and add the finger result comparison Python-scripts to the system.

Add a notification class which can be called by the Artculus Python class so notifications can be sent to the configured receivers.

The weights assigned to fingers are not based on research but on the experience of the IP Hijacking Researchers. Research can be done to tune these to make the results more reliable. The system is developed in a way that the weight is configurable per finger.

Cookie/session checking

If the site that is being monitored is using a login system, one could pass the cookie information to the Sentinels. The Sentinels will continue visiting the site keeping the cookie/session active.

When the owner of the site changes the knowledge of which cookie belongs to which username will be lost, indicating a problem. The effectiveness and reliability of this method is something that would have to be researched.

BGP update monitoring

Currently Artculus does not support BGP monitoring. An addition that could be built is a module that monitors BGP updates through BGPmon open telnet servers. A change in the BGP infrastructure could be directly fingerprinted and compared with results from other Sentinels, allowing Greenhost to screen the Internet live and detect possible IP hijacking.

Used tools

Special thanks goes out to the developers of:

- Nmap
- Whatweb
- Bootstrap
- Python
- Apache
- UNIX networking utils
- jQuery

The development of Artculus relies on these tools and have made the realization of Artculus possible.

References

- [1] *Techniques in OS-Fingerprinting*. Hagenberg, September 2005, <http://nostromo.joeh.org/osf.pdf>
- [2] *Accurate Real-time Identification of IP Prefix Hijacking*. Xin Hu, Z. Morley Mao University of Michigan, date unknown, <http://www.eecs.umich.edu/techreports/cse/2006/CSE-TR-516-06.pdf>
- [3] Zheng Zhang, Ying Zhang, Y. Charlie Hu, Z. Morley Mao, Randy Bush *iSPY: Detecting IP Prefix Hijacking on My Own*. Seattle, August 2008, http://web.eecs.umich.edu/~zmao/Papers/sigcomm08_ispy.pdf
- [4] Matthew Smart, G. Robert Malan, Farnam Jahanian *Defeating TCP/IP Stack Fingerprinting*. Department of Electrical Engineering and Computer Science, University of Michigan https://www.usenix.org/legacy/events/sec00/full_papers/smart/smart.html/index.html
- [5] https://en.wikipedia.org/wiki/Border_Gateway_Protocol
- [6] *How Pakistan knocked YouTube offline (and how to make sure it never happens again)* http://news.cnet.com/8301-10784_3-9878655-7.html Declan McCullagh
- [7] *Someones Been Siphoning Data Through a Huge Security Hole in the Internet* <http://www.wired.com/threatlevel/2013/12/bgp-hijacking-belarus-iceland/> Kim Zetter December, 2013
- [8] *Spam? Not Spam? Tracking a hijacked Spamhaus IP* <https://greenhost.nl/2013/03/21/spam-not-spam-tracking-hijacked-spamhaus-ip/> Mart van Santen March, 2013
- [9] *Stealthy IP Prefix Hijacking: Don't Bite Off More Than You Can Chew* <http://conferences.sigcomm.org/sigcomm/current/papers/p491-mcArthurA.pdf> Christian McArthur, Mina S. Guirguis Texas, August 2008
- [10] *TCP/IP Fingerprinting Methods Supported by Nmap* <http://nmap.org/book/osdetect-methods.html>
- [11] *Articulus - A BGP IP hijacking detection tool*
By Magiel van der Meer & Eddie Bijnen
<http://magiel.v-dmeer.nl/projects/RP1/articulus.zip>