Final Report
Finding Vulnerable Network Gadgets in the Internet Topology

By:
Nir Amar
M.Sc. Student in CS Department, Technion

Advisor:
Dr. Gabi Nakibly, CS Department, Technion

Served as part of the Course "Project in Computer Security" (236349)
18/03/2015
Table of Contents

1. ABSTRACT .............................................................................................................. 4

2. GOALS ................................................................................................................... 5

3. BGP ....................................................................................................................... 6

3.1. AUTONOMOUS SYSTEM (AS) ............................................................. 6
    AUTONOMOUS SYSTEM ................................................................................... 6
    INTERIOR AND EXTERIOR ROUTING PROTOCOLS .......................... 7
3.2. BORDER GATEWAY PROTOCOL (BGP) ........................................... 7
    BGP GOAL ........................................................................................................... 7
    BGP MESSAGES ............................................................................................... 7
    ROUTING INFORMATION BASES .............................................................. 8
    RELATIONSHIPS BETWEEN AS .............................................................. 8
    ROUTING POLICIES AND EXPORT POLICIES IN PRACTICE .... 9
    THE OVERALL IMAGE – THE BGP ALGORITHM .................................. 9

4. ATTACKING BGP .................................................................................................. 11

4.1. QUANTIFYING THE IMPACT OF ATTACKS .................................. 11
4.2. BGP ATTACKS CLASSIFICATION .................................................. 11
4.3. BGP SECURITY PROTOCOLS & THEIR INTUITIVE ATTACKS ....... 11
4.4. SMART – COUNTER INTUITIVE ATTACKS ...................................... 12

5. TECHNOLOGIES ................................................................................................... 14

5.1. PROGRAMMING LANGUAGE - C++ .................................................. 14
5.2. EXPLISAT - GUIDING SAT-BASED SOFTWARE VERIFICATION WITH EXPLICIT STATES ........................................ 14

6. SOFTWARE VERIFICATION .................................................................................. 15

6.1. WHAT IS MODEL CHECKING ............................................................ 15
6.2. EXPLISAT - GUIDING SAT-BASED SOFTWARE VERIFICATION WITH EXPLICIT STATES ........................................ 15
6.3. USING EXPLISAT IN THE PROJECT .................................................. 15

7. THE BGP SOFTWARE MODEL .............................................................................. 17

7.1. PREPARING THE MODEL ............................................................................ 17
7.2. INPUT ............................................................................................................... 18
    A TOPOLOGY ..................................................................................................... 18
    AN ATTACK ........................................................................................................ 18
    IMPORTANT NOTICE ....................................................................................... 18
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.3. OUTPUT</td>
<td>19</td>
</tr>
<tr>
<td>ATTRACTION ATTACK</td>
<td>19</td>
</tr>
<tr>
<td>INTERCEPTION ATTACK</td>
<td>19</td>
</tr>
<tr>
<td>8. FINDINGS AND RESULTS</td>
<td>21</td>
</tr>
<tr>
<td>8.1. TOPOLOGY GENERATION</td>
<td>21</td>
</tr>
<tr>
<td>TOPOLOGY GENERATION IS THE MOST TIME AND MEMORY CONSUMING PART</td>
<td>21</td>
</tr>
<tr>
<td>CHARACTERISTICS FOR REDUCING TOPOLOGIES SIZE</td>
<td>21</td>
</tr>
<tr>
<td>8.2. ATTACK GENERATION</td>
<td>24</td>
</tr>
<tr>
<td>ATTRACTION ATTACK</td>
<td>24</td>
</tr>
<tr>
<td>INTERCEPTION ATTACK</td>
<td>26</td>
</tr>
<tr>
<td>9. PROJECT SCHEDULE</td>
<td>29</td>
</tr>
<tr>
<td>10. CONCLUSION</td>
<td>30</td>
</tr>
<tr>
<td>11. REFERENCES</td>
<td>31</td>
</tr>
</tbody>
</table>
1. Abstract

In response to high-profile Internet outages, BGP security variants have been proposed to prevent the propagation of bogus routing information. The article "How Secure are Secure Interdomain Routing Protocols" [1] presents the main security protocols (origin authentication, soBGP, S-BGP, and data-plane verification) and analyzes them according to different attacks. Then it categorizes the BGP attacks to two categories:

Intuitive attacks: Intuition suggests that an attacker can maximize the amount of traffic he attracts by widely announcing a short path that is not flagged as bogus by the secure protocol.

Counter Intuitive (Smart) attacks: strategies like announcing longer paths, announcing to fewer neighbors, or triggering BGP loop-detection.

The article simulates attacks from both categories on real world AS’s topologies using all of the above secure BGP protocols. Finally, it shows that a clever export policy can often attract almost as much traffic as a bogus path announcement. Thus, the article implies that secure routing protocols only deal with one half of the problem: while they do restrict the paths the manipulator can announce, they fail to restrict his export policies and mechanisms that police export policies (e.g., defensive filtering) are crucial, even if S-BGP (as the most efficient security protocol) is fully deployed.

Another surprising insight is that the articles shows that the smart attack strategy is quite effective, in some cases, even more effective than the intuitive attacks!
2. Goals

Different attacks have been published in which an attacker publishes false BGP advertisements to other ASes, thereby attracting more traffic than it should. These attacks rely on a special topology structure (gadget) between the ASes and the attacker.

The project main goal is to find gadgets and appropriate "smart / counter-intuitive" attacks on those gadgets using SAT-Based Software Verification tool called ExpliSat [2].

The project had two sub goals / stages leading to the main goal:

1. Studying how BGP works and reading related literature.
2. Developing a BGP software model. The model will simulate the BGP logic that is important for our research, i.e. not all the inner BGP messages are simulated. Moreover, the model would be able to be extended in order to simulate malicious AS that has its own export policy and tries to attract traffic which aimed to some victim. The model will analyze the effectiveness of given attacks on given topologies.
3. BGP

In order to understand the importance of the project and its implications, we must first understand

- What Autonomous System (AS) is and how As's communicates
- What is BGP and BGP security aspects
- What are the vulnerabilities in BGP and how can attacker exploit them.

3.1. Autonomous System (AS)

**Autonomous System**

On the Internet, an autonomous system (AS) is the unit of router policy, either a single network or a group of networks that is controlled by a common network administrator (or group of administrators) on behalf of a single administrative entity (such as a university, a business enterprise, or a business division). An autonomous system is also sometimes referred to as a routing domain. An autonomous system is assigned a globally unique number, sometimes called an Autonomous System Number (ASN).

![Figure 1 - Autonomous System](image)

The routing problem is divided into:

1. Routing within a single autonomous system (intra-domain routing).
2. Routing between autonomous systems (inter-domain routing).

An AS can run whatever intra-domain routing protocol it chooses. It can even use static routes or multiple protocols.

Using an inter-domain routing protocol, different ASs share reachability information with each other. A router connecting an AS to the rest of the Internet is a border router.
**Interior and exterior routing protocols**
Routers within each domain speak interior routing protocols. The border routers speak both interior and exterior routing protocols.

Exterior routing protocols have to ensure that:
- Their network is not used for transit, unless authorized.
- Local packets do not traverse unauthorized or insecure ASs

Issues in interconnecting exterior and interior routing protocols:
- The protocols may use different ways to determine link costs
- The protocols may use different routing protocols.

**Internet routing protocols**
- **RIP** (routing information protocol), a distance-vector algorithm, very popular, because it was distributed with BSD Unix.
- **OSPF** (open shortest path first): a link-state algorithm
- **IS-IS**: a link state protocol, and quite similar to OSPF

**Exterior routing protocols**
BGPv4 (border gateway protocol), based on path-vectors, rather than on distance-vectors.

### 3.2. Border Gateway Protocol (BGP)

**BGP Goal**
The primary function of a BGP speaking system is to exchange network reachability information with other BGP systems. This network reachability information includes information on the list of Autonomous Systems (ASes) that reachability information traverses. This information is sufficient for constructing a graph of AS connectivity for this reachability, from which routing loops may be pruned and, at the AS level, some policy decisions may be enforced. (BGP RFC – 4271)

**BGP Messages**
BGP uses TCP as its transport protocol. BGP listens on TCP port 179. A TCP connection is formed between two systems. They exchange messages to open and confirm the connection parameters. The initial data flow is the portion of the BGP routing table that is allowed by the export policy, called the Adj-Ribs-Out. Incremental updates are sent as the routing tables change.
After a TCP connection is established, the first message sent by each side is an OPEN message. If the OPEN message is acceptable, a KEEPALIVE message confirming the OPEN is sent back.

KEEPALIVE messages may be sent periodically to ensure that the connection is live. NOTIFICATION messages are sent in response to errors or special conditions. If a connection encounters an error condition, a NOTIFICATION message is sent and the connection is closed. Routes are advertised between BGP speakers in UPDATE messages.

**Routing Information Bases**

Routes are stored in the Routing Information Bases (RIBs): namely, the Adj-RIBs-In, the Loc-RIB, and the Adj-RIBs-Out:

**Adj-RIBs-In:** The Adj-RIBs-In stores routing information learned from inbound UPDATE messages that were received from other BGP speakers. Their contents represent routes that are available as input to the Decision Process.

**Loc-RIB:** The Loc-RIB contains the local routing information the BGP speaker selected by applying its local policies to the routing information contained in its Adj-RIBs-In. These are the routes that will be used by the local BGP speaker. The next hop for each of these routes MUST be resolvable via the local BGP speaker’s Routing Table.

**Adj-RIBs-Out:** The Adj-RIBs-Out stores information the local BGP speaker selected for advertisement to its peers. The routing information stored in the Adj-RIBs-Out will be carried in the local BGP speaker’s UPDATE messages and advertised to its peers.

**Relationships Between AS**

Customer – Provider (transit) relation – the customer pays to the provider for traffic on the link. An AS will export to its providers paths it learned from its customers. An AS will export to its customer paths it learned from providers, customers and peers.

Peer-to-peer (peering) relation – the link is intended for traffic between two neighbors and their customers. An AS will export to its peers paths it learned from its customers only.
Routing policies and export policies in practice

Local pref (LP):
- Prefer customer routes over peer routes.
- Prefer peer routes over provider routes

AS paths (SP):
- Prefer shorter routes over longer routes.

Tiebreak (TB):
- Use intradomain criteria (e.g., geographic location, device ID) to break ties among remaining routes.

The Overall Image – The BGP algorithm

The BGP algorithm is executed each time the router receives an update message from a neighboring router. It consists of three steps:

1. Import: if the path information for an IP prefix in the Update message is different from the information previously received from that router, the Adj-RIBs-In database is updated.
2. Decision: if it was new information, then a decision process determines which route, of all those presently recorded in the Adj-RIBs-In database, has the best routing path for that IP prefix.

Recall that this algorithm is based on local policy. If the best path chosen as a result of this decision process is different from the one currently recorded in
the Loc-RIB database, then Loc-RIB is updated and internal routers are informed.

3. **Export**: If the decision process found a better path (Loc-RIB was updated) then the Adj-RIBs-Out database may also be updated as well. In such a case the router sends out update messages.

![Figure 3- BGP Policies](image-url)
4. Attacking BGP

4.1. Quantifying the impact of attacks

Before talking about Known Attacks, let’s propose a way to quantify the impact of attacks. It does not suffice to say that one protocol, say S-BGP, is four times as effective as another protocol, say origin authentication, at preventing a specific type of attack strategy; there may be other attack strategies for which the quantitative gap between the two protocols is significantly smaller. Since these more clever attack strategies can just as easily occur in the wild, our comparison must be in terms of the worst possible attack that the manipulator could launch on each protocol.

4.2. BGP Attacks Classification

Attraction vs. Interception attacks – In an attraction attack, the manipulator’s goal is to attract traffic, i.e., to convince the maximum number of ASes in the graph to forward traffic that is destined to the victim IP prefix via the manipulator’s own network. In interception attacks the manipulator may want to eavesdrop or tamper with traffic before forwarding it on to the legitimate destination. In an interception attack, the manipulator has the additional goal of ensuring that he has an available path to the victim.

An attack strategy may include: 1. announcing an unavailable or non-existent path. 2. Announcing different paths to different neighbors. 3. Announcing a legitimate available path that is different from the normal path. 4. Exporting a path (even the legitimate normal path) to a neighbor to which no path should be announced to, according to the normal export policies.

4.3. BGP Security Protocols & Their Intuitive Attacks

BGP – BGP does not include mechanisms for validating information in routing announcements. Attack Example – Export “852-Prefix”.

Origin Authentication - uses a trusted database to guarantee that an AS cannot falsely claim to be the rightful owner for an IP prefix. Attack Example – Export “852, 3248, Prefix”.

SoBGP (Secure Origin BGP) provides origin authentication as well as a trusted database that guarantees that any announced path physically exists in the AS-level topology of the Internet. Attack Example – Export “852, 3303, 3248, Prefix”. (It has Peer-To-Peer path – not available)

S-BGP (Secure BGP) In addition to origin authentication also uses cryptographically-signed routing announcements to provide a property called path verification. Path verification guarantees that every AS a can only announce a path abP to its neighbors if has a neighbor b that announced the path bP to a. Thus, it effectively limits a single
manipulator to announcing available paths. Attack – Export "852, 1239, 286, 3248, Prefix".

Data-plane verification prevents an AS from announcing one path, while forwarding on another. Thus, if the manipulator in Figure 1 wants to maximize his attracted traffic, he must also forward traffic on the provider path. No Intuitive attack exists.

Defensive filtering polices the BGP announcements made by stub ASes. Thus, defensive filtering requires each provider to keep track of the IP prefixes owned by its stub customers. If a stub announces a path to any IP prefix that it does not own, the provider drops/ignores the announcement. Defensive filtering completely eliminates attacks by stubs!

4.4. Smart – Counter Intuitive Attacks

1. Attract more by announcing longer paths.
   Naive strategy – The upper (green) figure shows the "Shortest-Path Export-All" attack strategy, where the manipulator naively announces a three-hop available path, "20984, 702, 6830, Prefix" to his provider 43284. Since ASes 43284 and 13030 prefer the customer path to that manipulator, over their existing peer paths, both will forward traffic to the manipulator. He intercepts traffic from 16% of the Internet, or exactly 5569 ASes.
   Clever strategy – The lower (purple) figure shows the manipulator cleverly announcing a four-hop available path "20984, 43284, 13030, 6830, Prefix" to his provider AS 702. AS 702 will prefer the longer customer path through the manipulator over his shorter peer connection to AS 6830, but this time, the manipulator triples the amount of traffic he attracts, intercepting traffic from a total of 56% of the Internet, or exactly 18664 ASes. In fact, by announcing a longer path, the manipulator earns almost as much traffic as the aggressive prefix hijack.
2. **Attract more by exporting less**

Naive strategy. The "Shortest-Path Export-All" attack strategy requires the manipulator to announce his path to all his neighbors. On the left, when the manipulator announces a path to provider AS 2828, both AS 2828 and its two Tier 1 providers will route through the manipulator. As a result, the two Tier 1's use four-hop paths to the victim, and the manipulator attracts traffic from 40% of the Internet, i.e., 13463 ASes.

Clever strategy. On the right, the manipulator increases his traffic volume by almost 25%, by not exporting to his provider AS 2828. Because AS 2828 no longer has a customer path to the victim, he is forced to use a peer path through AS 1239. Because AS 2828 now uses a peer path, he will not export a path to the two Tier 1 ASes 1299 and 174. The Tier 1s are now forced to choose a shorter three-hop peer path to victim through the manipulator. Because the two Tier 1's now announce shorter paths to their customers, they become more attractive to the rest of the Internet, the volume of traffic they send to the manipulator quadruples, and the manipulator attracts 50% of the Internet, i.e., 16658 ASes.

![Figure 12: Exporting less.](image)

3. **Attract more by gaming loop detection.**

Standard prefix hijack – The manipulator announces the path "25885, Prefix" and attracts traffic from the most of the Internet, exactly 32010 ASes. Notice also that because AS 3561 prefers customer paths, this large AS will chose to forward his traffic along the five-hop customer path through the manipulator.

False loop prefix hijack – The manipulator claims that innocent AS 4436 originates the prefix, announcing "25885, 4436, Prefix" to AS 25653. However, when this false loop is announced to AS 4436, BGP loop detection will cause AS 4436 to reject the path through AS 25653. As a result, AS 3561 has no customer path to the prefix, and instead chooses to forward traffic along the shorter peer path. Now, AS 3561 announces a shorter, four-hop path to his neighbors "3561, 25653, 25885, 4436, Prefix", making him more attractive to the rest of the Internet, and attracting more traffic to the manipulator. For this, the manipulator attracts 360 more ASes than standard prefix hijack, i.e., 32370 ASes.

![Figure 13: False loop prefix hijack.](image)
5. Technologies

5.1. Programming Language - C++

The BGP model has an Object oriented, loosely coupled and extensible design.

This design allows extending the model for:

- Adding user-defined topologies.
- Adding user-defined bgp-attacks.
- Controlling parameters such as: topology size, attack-path size.

5.2. ExpliSat - Guiding SAT-Based Software Verification with Explicit States

Hybrid method for software model checking that combines explicit-state and symbolic techniques. The method traverses the control flow graph of the program explicitly, and encodes the data values in a CNF formula, which then solved using a SAT solver. To avoid traversing control flow paths that do not correspond to a valid execution of the program, the tool introduce the idea of a representative of a control path.

Property of IBM.
6. Software Verification

Given a (model of) hardware or software system and a formal specification, does the system satisfy the specification?

6.1. What is Model Checking

Model checking is a technique for verifying finite state concurrent systems such as sequential circuit designs and communication protocols. It has a number of advantages over traditional approaches that are based on simulation, testing, and deductive reasoning. In particular, model checking is automatic and usually quite fast. Also, if the design contains an error, model checking will produce a counterexample that can be used to pinpoint the source of the error.

By Edmund M. Clarke, Orna Grumberg and Doron Peled.

6.2. ExpliSat - Guiding SAT-Based Software Verification with Explicit States

As presented before: Explisat [2] is a hybrid method for software model checking that combines explicit-state and symbolic techniques. The method traverses the control flow graph of the program explicitly, and encodes the data values in a CNF formula, which then solved using a SAT solver. To avoid traversing control flow paths that do not correspond to a valid execution of the program, the tool introduce the idea of a representative of a control path.

6.3. Using Explisat in the Project

First we needed to develop a BGP software model. The model simulates the BGP logic that is important for our research, i.e. not all the inner BGP messages are simulated.

Moreover, the model was extended in order to simulate malicious AS that has its own export policy and tries to attract traffic which aimed to some victim. The model will analyze the effectiveness of given attacks on given topologies.

The BGP model accept user-defined parameters: A topology and an attack and calculate the effectiveness of the attack on the given topology compared to the intuitive attack on this topology.

The final statement in the software is an assertion:

The intuitive attack is more effective than the user-defined attack. (This is the software specification)
Finally the software uses Explisat. Explisat passes on the parameters in a non-deterministic manner and check whether the software satisfy the specification. If not, counterexample is given.

User parameters: Topology and Attack

Simulate BGP using the SW model

Intuitive Value = the intuitive attack effectiveness
User Value = the user-defined attack effectiveness

Assert (User Value < Intuitive Value)

Expisat  Counter intuitive attack
7. The BGP Software Model

7.1. Preparing the Model

Formal verification requires some preparations to be done on the code under test before starting. Those preparations include the definition of program input values and assertions to be checked. Typically, program input values are set to be nondeterministic, sometimes with additional code providing constraints on those values. Explisat provides a set of API functions for this task.

The API functions are:

- `<Type> nondet_<type> ()`
  Returns a non-deterministic value of a signed type `<type>`, e.g. "int nondet_int ()". `<Type>` can be char, short, int, long.

- `Unsigned <type> nondet_u<type> ()`
  Same as previous, but for unsigned types, e.g. "unsigned char nondet_uchar()".

- `void fv_assume(int condition)`
  Provides an assumption that the execution path which passes through it should obey. If the condition is not satisfied, the execution path is truncated (i.e. this path is considered verified without going further).

- `Void fv_assert(int condition)`
  Asserts the condition. If condition is not satisfied, verification is considered failed. Input / User Parameters
7.2. Input
The software has 2 inputs: A topology and an attack. Each one of those inputs should be constructed from the above API.

A topology
A topology can be thought of as a directed graph.

Without loss of generality, one can decide that the attacker is the first AS (ASN0) and the victim is the second (ASN1).

\( V = \) The AS'es.

\(|V| = \) The number of vertices. can be received from non-deterministic int.

\( E = \) The As'es relation. Each relation in the graph can be either: customer-provider, provider - customer, peer-to-peer or no link (no relation). This information will also be non-deterministic.

An attack
An attack is actually a fake BGP publish message in the form of:

\(< Attacker, As_1, As_2, ..., As_n, Victim >\)

The \(< As_1, As_2, ..., As_n >\) part will be non-deterministic.

The attacker can choose to which neighbors to publish the attack in a non-deterministic manner as well.

Important Notice
The topology and the attack are un-related!

The user can decide that he have a special topology that he want to find a counter-intuitive attack on it. The software allows such thing to happen.

Same for the case that the user have a specific attack (for example – shortest-path-export-all attack) that he would like to test it on several topologies.
7.3. Output

Recall the software specification:

"The intuitive attack is more effective than the user-defined attack."

The software output is according to her specification. If the software satisfy its specification, then the verification passed. Else a counterexample is given (counter-intuitive attack which is more effective than the best intuitive attack!).

In order to quantify the impact of the user-defined attack, the software compares it to an intuitive attack. An intuitive attack is divided to the two categories we seen before:

**Attraction attack**
In an attraction attack, the manipulator's goal is to attract traffic, i.e., to convince the maximum number of AS'es in the graph to forward traffic that is destined to the victim IP prefix via the manipulator's own network.

The most intuitive attack in this category is "Shortest-Path Export-All" (SPEA) attack strategy. Announce to every neighbor, the shortest possible path that is not aged as bogus by the secure routing protocol.

**Interception attack**
In an interception attack, the manipulator may want to eavesdrop or tamper with traffic before forwarding it on to the legitimate destination. In an interception attack, the manipulator has the additional goal of ensuring that he has an available path to the victim.

The most intuitive attack in this category is "Shortest-Path Export-All- with-Connectivity" (SPEA-C) attack strategy. The manipulator announce his shortest legitimate available path to the victim, instead of his normal path. Another limitation is to whom the attacker announce this path to. It must keep in mind that a "black hole" won't be created.
Figure 4 - Aggressive export policies: Taken from article [2]. CCDF of the probability that at least $x$-fraction of the ASes in the internet forward traffic to the manipulator. Probability is taken over a randomly chosen victim, and a manipulator chosen randomly from the class of ASes that have at least 25 customers. The figure shows 3 different strategies:

(a) Announce the shortest available path to all neighbors (SPEA).

(b) Announce the normal path to all neighbors (SPEA-C)

(c) Announce the normal path using the normal export policy.
8. Findings and Results

8.1. Topology Generation

**Topology Generation is the Most Time and Memory Consuming Part**

The topology generation has two non-deterministic decisions:

1. How many As'es are in the topology (\| V \|).
2. What is the relation between each As'es pair?

Recall that for question 2 there are 4 possibilities: customer-provider, provider - customer, peer-to-peer or no link (no relation). The 4 possibilities are for each pair in the topology. Therefore number of topologies that can be is \(4^{\|V\|}\). Meaning that the number of actions/statements the must be perform and the amount of memory needed in case the software will traverse the entire BGP model is \(O\left(4^{\|V\|^2}\right)\). Here comes Explisat, and the model checking theory which helps to reduce this number.

The biggest topology I could generate using the Technion host memory capacity is in size of 5-6 As'es.

Nevertheless, **as much as the software host will have more time and memory resources, he would be able to work about bigger topologies.**

**Characteristics for Reducing Topologies Size**

While developing the topology generation module and applying optimizations in order to shorten the amount of time and memory this process consume, I discovered the main characteristics that a topology have in our context. While maintaining those characteristics, one can reduce his topology size, investigate counter-intuitive attacks on the reduced topology, and deduce the counter-intuitive attack the original, bigger topology.

For example, let us examine the following topology:
Figure 5 - Reducing Big Topology Size - Intuitive attack: The manipulator, AS 20984, is a small stub AS in Basel, Switzerland, that has one large provider, AS 702 owned by Verizon and having degree over 500,000 and one small provider, AS 43284 owned by Industrielle Werke Basel and having degree only four. The victim is European broadband provider AS 6830.

SPEA strategy. The figure shows the "Shortest-Path Export-All" attack strategy, where the manipulator naively announces a three-hop available path, (20984, 702, 6830, Prefix) to his provider 43284. Since AS'es 43284 and 13030 prefer the customer path to that manipulator, over their existing peer paths, both will forward traffic to the manipulator. He intercepts traffic from 16% of the Internet, or exactly 5569 AS'es.

Clever strategy. Figure 6-- Reducing Big Topology Size - Counter-Intuitive attack shows the manipulator cleverly announcing a four-hop available path "20984, 43284, 13030, 6830, Prefix" to his provider AS 702. AS 702 will prefer the longer customer path through the manipulator over his shorter peer
connection to AS 6830, but this time, the manipulator triples the amount of traffic he attracts, intercepting traffic from a total of 56% of the Internet, or exactly 18664 ASes. In fact, by announcing a longer path, the manipulator earns almost as much traffic as the aggressive prefix hijack.

I ran this example in the software, while AS702 had only 3 customers and AS13030 had only one customer. The results were identical to the large original topology and the new topology size is only 9 instead of few thousands!

From the above example, we studied that main characteristics that the reduced topology has to maintain are:

1. The attacker and the victim location in related to one from the other. I.e. in the example, the attack add one customer-peer path to the victim, and one customer-customer-peer path to the victim.

2. The number of links for the vertices on the paths from the attacker to the victim. I.e. in the example, the number of customer-clients-providers the AS702 and AS13030 had.

While maintaining the above characteristics, one can reduce his topology size, hence the memory and time he needs by a lot!
8.2. Attack Generation

**Attraction Attack**

*Creating Counter Intuitive Attacks as Least as Effective As SPEA*

Using my BGP model, I could not create counter intuitive attacks that are at least effective as Shortest-Path-Export-All attack strategy.

Example:

In Figure 8 - Attraction Attack Generation, the intuitive attack is SPEA: the attacker export the path "(Attacker, Victim)". Its effectiveness is: 8 As'es will now pass through the attacker.

Notice that BGP has a built-in defense mechanism called loop detection. It is a basic defense mechanism which says: If an AS receives a BGP path advertisement, it looks if he appears in that advertisement. If so, he would not accept this advertisement. This mechanism is meant to prevent routing loops.

The best counter intuitive attack my model has found is: the attacker export (Attacker, 756, Victim). The attack can be exported to AS720 and to AS 43284, but it been exported (non-determinism) to AS43284 only.

Quantifying Effectiveness – AS702 will prefer going through his customer (the attacker) rather his peer (the victim). Therefore all of AS702 clients will also pass
through the attacker, except AS756. AS43284 will prefer going through his customer (this attacker) rather his provider (AS13030). AS13030 will prefer going through his customer (AS43284) rather his peer (the victim). Therefore AS13030's client will also pass through the attacker. Summary - The attack effectiveness is: 7 As' es will now pass through the attacker. Side note – you can replace AS756 with any of the rest of AS702 clients in the above attack.

Notice that because the loop detection defense mechanism is applied on the network (as it is part of the BGP Protocol), than AS756 will not pass through the attacker, so the attack effectiveness is only 7, i.e. less effective than SPEA!

In general for attraction attacks, my model was not able to find a counter-intuitive attack which is equal or outperform the intuitive attack - SPEA.

Summary – No non-deterministic attack was found which is better than SPEA in attraction attack category. However, I did manage to create counter-intuitive attacks that their effectiveness exceeds a given threshold. As in the example, I set an attack effectiveness threshold to be at least 7 As' es, and I found the above attack.
**Interception Attack**

**Creating Counter Intuitive Attacks as Least as Effective As SPEA-C**

Using the BGP model, one can create counter intuitive attacks that are at least effective as Shortest-Path-Export-All with connectivity attack strategy.

Even if an attack is equal in its effectiveness to SPEA-C, it is good result, because this attack would overcome defense mechanisms that were design to stop the intuitive attacks.

Example:

In Figure 9 - Interception Attack Generation, the intuitive attack is SPEA-C: the attacker export (Attacker, 702, Victim). Its effectiveness is: 3 As'es (AS43284, AS13030, and AS6757) will now pass through the attacker.

The counter intuitive attack is: the attacker export (Attacker, 43284, 13030, Victim). The attack can be exported to AS720 and to AS 43284, but it been exported (non-determinism) to AS43284 only.

Quantifying Effectiveness – AS702 will prefer going through his customer (the attacker) rather his peer (the victim). Therefore all of AS702 clients will also pass through the attacker. AS43284, AS13030 and AS6757 will go through AS13030 to the victim and not through the attacker. Therefore AS13030's client will also pass through the attacker. Summary - The attack effectiveness is: 5 As’es will now pass through the attacker.
Important Notice: We actually find, using Explisat, a counter-intuitive attack which outperform the intuitive attack!

Creating Counter Intuitive Attacks as Least as Effective As SPEA-C with Non-deterministic Topologies

I used the topology creation mechanism to investigate a counter intuitive attack on a non-deterministic topology. Actually it shows the mix of my two main capabilities: Topology Creation and Attack Creation.

Because of the memory constraints explained on chapter 8.1, I use the non-determinism on part of the previous topology.

Example:

In Figure 10 - Interception Attack Generation on Non-Deterministic Topology I applied non-determinism on few of the links between AS702 and who used to be his customers (see previous figure). The intuitive attack is also SPEA-C: the attacker export (Attacker, 702, Victim). Its effectiveness is: 3 As’es (AS43284, AS13030, and AS6757) will now pass through the attacker.

The counter intuitive attack is: the attacker export (Attacker, 43284, 13030, Victim). The attack can be exported to AS720 and to AS 43284, but it been exported (non-determinism) to AS43284 only.

Quantifying Effectiveness – AS702 will prefer going through his customer (the attacker) rather his peer (the victim). Therefore all of AS702 clients and peer will also
pass through the attacker. AS43284, AS13030 and AS6757 will go through AS13030 to the victim and not through the attacker. Therefore AS13030's client will also pass through the attacker. Summary - The attack effectiveness is: 4 As'es will now pass through the attacker.

Important Notice: Again, with non-deterministic topology, we find, using Explisat, a counter-intuitive attack which outperform the intuitive attack!
9. Project Schedule

20.10 – 2014 classes begin.

03.11 – Opening report submission.

24.11 – BGP Studying. Conclusion report submission (3 pages).

01.12 – Reading the article “How Secure are Secure Interdomain Routing Protocols?” Conclusion report submission (3 pages).

03.12 – Mid Semester meeting.

29.12 – Implementing BGP Model using Explisat (Bounded Model Checker for C/C++ programs).

19.1 – Restoration of attacks that were described in the article, using Explisat.

16.3 – Finding new attacks.

25.3 – Final presentations / final report submission.
10. Conclusion

In the end of my project, I have accomplished to achieve the project’s main goal.

Reminder: The project main goal was to find gadgets and appropriate "smart / counter-intuitive" attacks on those gadgets using SAT-Based Software Verification tool called Explisat.

I outlined steps to this goal:

1. Generating non deterministic topologies in order to investigate intuitive vs. counter-intuitive attacks on them.
   I succeeded to generate topologies in my memory constraints, and showed a way to reduce a given topology size while maintaining its important characteristics.

2. Generating non deterministic attacks. The attacks are divided to two categories:

   Attraction Attack – where the manipulator's goal is to attract traffic, i.e., to convince the maximum number of AS'es in the graph to forward traffic that is destined to the victim IP prefix via the manipulator's own network. I compared the non-deterministic attack to the intuitive attack in this category (Shortest-Path-Export-All).
   The non-deterministic attacks that I found had been less effective than the intuitive attack. However, I managed to generate counter-intuitive attacks that their effectiveness exceeds a given threshold.

   Interception Attack – The manipulator may want to eavesdrop or tamper with traffic before forwarding it on to the legitimate destination. I compared the non-deterministic attack to the intuitive attack in this category (Shortest-Path-Export-All- with-Connectivity).
   The non-deterministic attacks that I found were more effectiveness then the intuitive attack!

The topology and the attack creation are un-related!

The user can decide that he have a special topology that he want to find a counter-intuitive attack on it. The software allows such thing to happen.

Same for the case that the user have a specific attack (for example – shortest-path-export-all attack) that he would like to test it on several topologies.

Real world topologies are in danger!
11. References

[1] "How Secure are Secure Interdomain Routing Protocols" by Sharon Goldberg (Microsoft Research, New England), Michael Schapira (Yale University), Peter Hummon (Princeton University), Jennifer Rexford (Princeton University).