CAT – A Practical Graph & SDL Based Toolkit for Vulnerability Assessment of 3G Networks

Kameswari Kotapati, Peng Liu, and Thomas F. LaPorta

The Pennsylvania State University, University Park, PA 16802
kotapati@cse.psu.edu, pliu@ist.psu.edu, tlp@cse.psu.edu

Abstract. This paper presents the Cellular Network Vulnerability Assessment Toolkit - CAT, designed for end-to-end vulnerability assessment of 3G networks. It is the first tool of its kind to model and represent 3G network vulnerabilities and attacks as attack graphs. CAT uses freely available 3G telecommunication specifications written in SDL, the standard Specification and Description Language to produce attack graphs. Attack graphs generated by CAT are unique due to their: (1) global representation of the network, (2) independence from physical deployments, and (3) depiction of the 3G attack graph model and cascading effects.

1 Introduction

Third generation (3G) wireless telecommunication provide circuit switched and high speed packet data services for 3G-enabled mobile devices. These networks have evolved from the isolated 1G & 2G networks by integrating with the Internet. This integration imports the inherent vulnerabilities of the Internet to the 3G networks and gives the end subscriber direct access to the control infrastructure of the 3G network. The goal of this research is to assess the vulnerabilities introduced by this merger by development of the Cellular Network Vulnerability Assessment Toolkit - CAT. CAT models 3G network vulnerabilities and attacks as attack graphs.

Attacks on the 3G network are unique because corruption propagates across the network due to normal end-to-end network operation. This feature known as the *cascading effect*, occurs due to the exchange of corrupt data items in signaling messages between 3G servers. Hence the goal of 3G network vulnerability assessment is to not only identify the attack origin but also the cascading effects caused due to the end-to-end system level vulnerabilities and interactions.

Manual deduction of vulnerabilities and attacks in 3G systems is not feasible because vulnerability deduction requires extensive knowledge of thousands of state machines and end-to-end networking of the telecommunication systems. Also standard Internet based vulnerability assessment tools are insufficient for 3G networks because they present physical vulnerabilities which is not the goal of 3G network vulnerability assessment. The goal is to identify end-to-end system level vulnerabilities and interactions that lead to the cascading effect.

CAT works by taking in 3G data parameters called seeds and goals as input, and uses free [2] technical specifications written in the Specification and Description Language (SDL) to identify system interactions that lead to the cascading effect. SDL is developed by the International Telecommunication Union (ITU)

and is designated as the formal description language for specifying and describing the functional behavior of telecommunication systems [24], [2] by major international standards bodies.

CAT is the first tool of its kind to present end-to-end vulnerabilities for any system with SDL specifications. CAT is practical because of its universal applicability and its independence from physical deployments. Other contributions in this paper include the definition of cascading effect of attacks, generic model of attack graphs for 3G networks and categorization of attacks on 3G networks. The attack graphs generated by CAT show the global network view and are succinct, loop free with low redundancy.

2 Related Work

Our literature survey comprises of vulnerability assessment in telecommunication networks and attack graphs.

The vulnerabilities of telecommunication networks are well addressed in the literature. Telecommunication standards [4], [5], [6] specify 3G security and identify certain security threats. Howard et al. [7], El-Fishway et al. [8], Lo et al. [9], Welch et al. [10], Clissmann et al. [11] have identified threats or attack scenarios on the 3G network while trying to prove the inadequacy of current security schemes, or present new architectures for 3G security. Brookson in [12] motivates the need for security. Mitchell et al. in [13], and Boman et al. in [14] discuss the security features available in current 3G networks. Kotapati et al in [1] present a taxonomy of cyber attacks in 3G networks. *However, the above literature does not present vulnerability assessment solutions.*

3G system administrators typically use conventional tools to assess the physical implementation vulnerabilities. In the literature there are no known techniques for *system level end-to-end vulnerability assessment in 3G networks.*

CAT adapts the attack graph technology which has been well investigated in the literature. This research has foundation in Swiler and Philips et al. [15], [16], [17]. Their work analyses network vulnerability using the graph based approach. Graphs generated by CAT extend their graphs to show normal network activity and subscriber activity in addition to attack activity and provide the global network view of the adversary action. Swiler and Philips [15], [16], [17] generated their attacks by backward exploration from the goal given atomic attacks as input. Unlike the Swiler and Philips model, CAT does not answer any 'what-if' questions regarding security effects of configuration changes. It does not analyze the graph or assign quantitative metrics. It also identifies atomic attacks.

Ritchey and Amman [18] used model checking for vulnerability analysis of a network. Amman et al. in [19] present a scalable method for representing attack graphs with the assumption of monotonocity. Sheyner et al. [3] have shown that model checking may be applied to automatic generation of attack graphs. Jha et al. [20], [21], [22] present algorithms for analyzing attack graphs. Jha et al. [20] present a minimization analysis technique that allows analysts decide which minimal set of security measures guarantee the safety of the system.

All the aforementioned attack graph research focuses on Internet vulnerability assessment. Schneier [23], states that in general, attacks against system may be represented in a tree structure. The literature surveyed reveals that CAT

is the first effort to use SDL specifications to automatically infer possible 3G network attacks in the form of an attack graph. It should be noted that extending the attack graph technology from the Internet to 3G networks is not trivial because Internet case lacks assessment of end-to-end network; and 3G network physical configurations are different between network deployments.

3 Overview

This section gives an overview of the 3G network, SDL and the model of the 3G attack graph.

3.1 3G network

The 3G network is comprised of a number of servers as shown in Fig. 1. The servers in the circuit switched domain of the 3G network include Home Location Register (HLR), Visitor Location Register (VLR), Mobile Switching Center (MSC) and Gateway Mobile Switching Center (GMSC). The 3G network provides service to the subscriber by the exchange of signaling messages among its servers.

All subscribers are permanently assigned to a HLR. The HLR is in the home network and stores permanent subscriber data and current subscriber location (pointer to VLR). VLRs are assigned to a specific administrative area and are associated with one or more MSCs. The VLR acts as a temporary repository and stores data of all mobile stations (user handset) that are currently roaming in its assigned area. The VLR obtains this data from the HLR assigned to the mobile station. The MSC acts as an interface between the radio system and the fixed network. It handles the circuit switched services to and from the mobile stations roaming into its area. The VLR and MSC are either in the home or home network depending on the location of the subscriber.

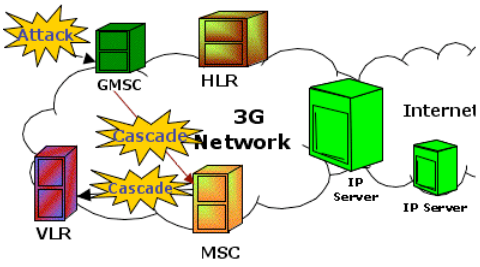

Fig. 1. Speech Attack

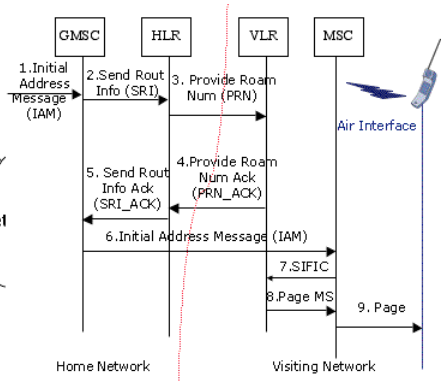

Fig. 2. Signal Flows for Call Delivery Service

The ***call delivery service*** is a *3G service* that delivers incoming calls to the subscriber regardless of the location of the called subscriber and the caller. Calls are sent (signaling message 'IAM' in Fig. 2.) to the GMSC, which is in charge of routing the call to the mobile station and passing voice traffic between different networks. The GMSC checks the called number in the incoming call ('IAM') and resolves it to the assigned HLR of the called party. It signals the HLR of the incoming call using the signaling message 'SRI'. The HLR is aware of the location where the called subscriber is currently visiting and requests

the corresponding VLR for a 'roaming number' ('PRN') to route the call and downloads the incoming call profile to the VLR. The VLR assigns a roaming number for routing the call and passes it on to the HLR ('PRN_ACK'). The HLR passes on this 'roaming number' to the GMSC ('SRI_ACK'). The GMSC uses this 'roaming number' to route the incoming call to the MSC where the subscriber is currently visiting. The MSC requests the VLR for the incoming call profile for the called subscriber ('SIFIC') and receives the profile in the 'Page MS' signaling message. The MSC alerts ('Page') the mobile station.

Each of the above signaling messages comprises of two types data elements: (1) parameters to invoke a function in the destination server; and (2) administrative parameters required for communication such as originating server address and destination server address.

3.2 SDL and Mapping to 3G

This section provides a background on SDL and uses the aforementioned 3G network to illustrate the usage of SDL. SDL is an object-oriented, formal language and is intended for the specification of event-driven, real-time, concurrent distributed systems interacting with discrete signals. SDL is a graphical language and SDL specifications do not indicate an implementation structure. In basic SDL the system description is hierarchically structured. It describes the local and remote behavior of telecommunication systems, as Systems, Blocks and Processes.

A ***System*** is comprised of a number of concurrently running Blocks that communicate by passing signaling messages through a channel. *Blocks* may be of different types and there may be multiple instances of a single block type at a time. The 3G telecommunication network show in Fig. 1 is an *example of the SDL system*. The servers in the 3G network correspond to blocks in SDL.

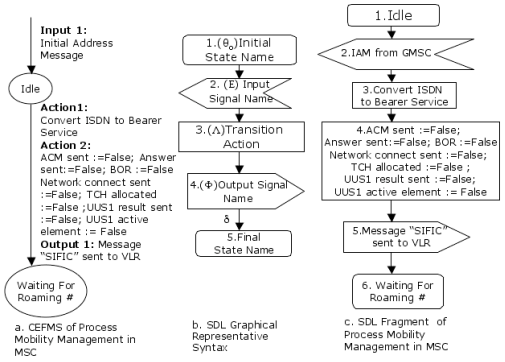

Fig.3. Example of CEFSM and SDL

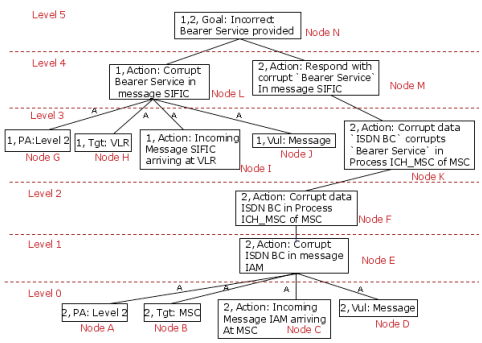

Fig.4. Attack Graph for Speech Attack

A ***Block*** represents a processing unit in a single location. A block is a collection of data and concurrently running processes of same and different types. Blocks provide service and communicate with each other by sending signaling messages through a channel. Data is always associated with or owned by the block. The MSC shown in Fig. 1 is an *example of the block* and it comprises of subscriber data for those roaming in its administrative area.

Processes are basic functional units of SDL systems and perform functions using a service logic and have the capability to change data associated with

the block. A Process may be defined as a communicating extended finite state machine (CEFSM) represented by a 6-tuple $(\Theta, \theta_0, \Xi, \delta, \Lambda, \Phi)$. Fig. 3b shows the graphical syntax of the process represented using SDL. A process in an initial state (θ_0) (node 1 in Fig. 3b) receives an input event (Ξ) (node 2), performs certain transition actions (Λ) (node 3), generates an output (Φ) (node 4), and finally transitions (δ) to another state (node 5). Θ represents the set of finite states in CEFSM. The input event comprises of incoming signaling messages from other processes in same or different blocks. Transition actions comprise of the getting or setting certain conditions or variables or functions. The output generated by the process comprises of signaling messages to processes in same or different blocks. At the end of its regular operation, the process transitions into a final state. *Process types within the MSC block may be broadly classified as Mobility Management, Call Handling, Operations and Maintenance, Fault Recovery, Handover and Subscriber Management.* The CEFSM state transition diagram of *Process Mobility Management* is shown in Fig. 3a. Mapping from the CEFSM to the SDL is one-to-one. Fig. 3c shows the actual SDL fragment for the basic CEFSM of Fig. 3a. The SDL diagram in Fig.3c is representative of the SDL diagrams used in telecommunication specifications.

3.3 3G Attack Graph

An *attack* in CAT may be defined as a network state transitions due to adversary action, where the final transition results in the adversary achieving a goal. CAT deduces possible attacks and presents them using attack graphs. *A 3G network specific attack graph, as shown in Fig. 4, is a network state transition showing the paths through a system starting with the conditions of the attack, the attack, continuing with the cascading effect of the adversary's attack action(s) and ending in the adversary's goal.* The *state of a 3G network* may be defined as the collective state of all its blocks.

The attack graph in Fig. 4 is constructed for the attack shown in Fig. 1. The attack happens by corruption of 'ISDN BC' data parameter in the 'IAM' message. The 3G data parameter 'ISDN BC' is considered as the direct intent of the attack and it is called as the *seed*. CAT builds this graph bottom-up. For description purposes, this attack graph has been divided into levels and assigned node labels followed by an alphabet. Each node has been assigned numbers; these numbers are tree numbers and correspond to the tree to which the node belongs. For example, all the nodes marked with number 2 form the second tree of the graph. Nodes at a particular level with the same tree number(s) are *AND* nodes. Nodes with the same tree number at a layer connected to a node in the layer above indicate AND nodes. For example at Level 0, Nodes A-D are AND nodes, Node L, M at Level 4 are OR nodes. The attack graphs generated by CAT comprise of three types of nodes; condition, action and goal.

Condition Nodes: represent conditions that hold for an attack. *Physical Access* corresponds to adversary's physical access to the network and may be classified as: (1) access to the air interface with the help of a physical device; (2) access to cables connecting central offices; and (3) access to 3G core network blocks in the central office. In the attack example of Fig. 4 the adversary's Physical Access is at Level 2 (Node A). The high level description of an adversary's *target*

is described by a block and indicates all the processes and data sources within a block. In the attack example of Fig. 4 the target is MSC (Node B). An adversary takes advantage of *vulnerabilities*: (1) by attacking the data parameters in *signaling messages* exchanged between blocks; (2) by attacking the service logic of a process in a block so that it behaves abnormally; and (3) by corrupting the data sources in a block. In the attack example of Fig. 4 the adversary corrupts data 'ISDN BC' in message IAM. The vulnerability is message (Node D).

Action Nodes may be events or non-events that causes a network transistion. Examples of events include incoming (Node C) and outgoing (Node M) signaling messages. An example of a non-event includes changing data associated with the block (Node F). Non-events cause a change in state of the network, but do not generate an event. Actions may be further classified as *adversary actions, normal network operations and normal subscriber activities.* Adversary actions comprise of insertion, corruption or deletion of data, signaling messages and service logic (Node E and Node L). Normal network operations include sending and receiving signaling messages (Node C). Subscriber activity may comprise of updating personal data and initiating service.

Goal Node are the final nodes in a tree occurring at the highest level (Node N in Fig. 4). They indicate corruption or derivation of data parameters due to the direct corruption of other data parameters (seeds) by the adversary. The goal is achieved when the corrupt data parameters are propagated to other 3G blocks. The goal of the attack graph is an action that comprises of an event (incoming or outgoing message) with corrupt goal parameter(s) or a non-event representing block level corruption.

The attack graph shows through its edges the implicit transition in network state. The *transition due to adversary action* describes the change in the state of the network as a result of the adversary action and is indicated by a edge marked by the letter A. The *network transition* describes the change in the state of the network as a result of any of the action nodes. By inclusion of normal network transitions in addition to the adversary transitions, the proposed attack graph shows not only the adversary's activity but also the global view or cumulative effect of adversary's action. This is a unique feature of the attack graph. Attacks graphs may be represented using a simplified form of CEFSM using a 4 tuple $(\lambda_0, \Lambda, \tau, L)$ where Λ represents nodes. $\lambda_0 \subseteq \Lambda$ represents the set of intial states. τ represents edges in the attack graph. L represents labels (attack A or non-attack $\emptyset$) for the edges/transitions.

4 Features of CAT toolkit

This section describes the features of the CAT toolkit.

4.1 Architecture

The overall architecture of CAT is shown in Fig. 5a. It is composed of the GUI subsystem, which takes seeds and goals as input from the user, the analysis engine subsystem, which explores the possibility of the seeds reaching the goal, and the SDL database in which the 3G telecommunication specifications are stored. The integrated data structure has structure similar to that of the SDL database described below and holds attack graph results from CAT analysis against the SDL database. Fig. 5b. shows the functional architecture of the

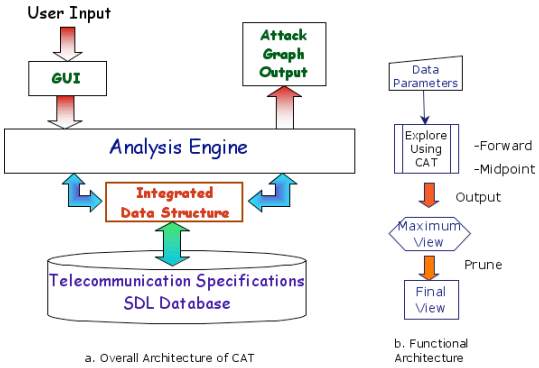

Fig. 5. Architecture of CAT

toolkit. The toolkit takes in seeds and goals as input and explores for the output. The initial output is a 'Maximum View' of the attack graph which is pruned to remove redundancy to provide a 'Final View' of the attack graph to minimize redundancy.

4.2 User Input

CAT works by taking in 3G data parameter seeds and goals as input from the user. Seeds are data parameters which when corrupted by the adversary may or may not lead to the goal. Seeds may merge at different stages of the attack graph. Goals are data parameters that are derived incorrectly due to the direct corruption of seeds by the adversary.

4.3 SDL System Input

The SDL database comprises of 3G telecommunication specifications [25]. Database contains signaling messages and data parameters they contain. They also contains information regarding processes in blocks. Process information is stored as initial and final states, input and output messages and actions. Unlike other methods, CAT does not require network topology, network configuration, adversary profile information or attacks in atomic form, as input. CAT works for any implementation which follows telecommunication standards. CAT assumes that the adversary has the necessary tools to penetrate the network at the different levels of physical access and vulnerabilities as described in Section 3.

4.4 Attack types

Attacks detected by CAT may be categorized as follows.

1. *1-Level Indirect attacks* are those attacks in which, corruption of $Seed_1$ leads to corruption of the goal hence reaching the goal. $Seed_1 \rightarrow Goal$.

2. *N-level indirect attacks* are those attacks in which, given any 'k' *seeds* $\in \{S\}$, $Seed_1$, $Seed_2$, ..., $Seed_k$ and $Goal$, corruption of $Seed_1$ leads to corruption of some seed $Seed_i$ (where $Seed_i \in \{S\}$), which in-turn leads to corruption of some seed $Seed_j$ (where $Seed_j \in \{S\}$) and so on until the $Goal$ is corrupt. $Seed_1 \rightarrow \ldots Seed_i \rightarrow Seed_j \rightarrow \ldots \rightarrow Seed_n \rightarrow Goal$.

3. In the *Collaborative attack*, a single seed cannot reach the $Goal$ but the corruption of multiple seeds allows for reaching of the $Goal$ i.e. $Seed_1$ & $Seed_2$ $\rightarrow Goal$.

4. In the *Multi-stage attack*, a first attack is used to gather information to perform a second attack. The second attack gathers information to perform another attack until the $Goal$ is reached.

4.5 Issues with SDL

CAT algorithms trace the flow of data through the network. Corruption of a data parameter may lead to the corruption of other data parameters which may or may not lead to the goal. When converting SDL specification to data used by CAT there are a number of issues that must be taken into consideration.

Message and Message-ACK Pairs: Issues arise with message and the corresponding message-ack pairs *e.g., message SRI and SRI-ACK, PRN and PRN-ACK*. The message typically comprises of a subscriber key for which the destination block assigns a value. The message-ack contains this value but not the subscriber key. The absence of the subscriber key in the message-ack (PRN-ACK) must be taken into consideration or certain attacks may be missed because the algorithm may lose track of data correlations.

Data Dependencies in Action Attributes: Actions in SDL-specifications (Table 2) may comprise of certain procedures such as 'Check Parameters'. SDL may not specify how the subroutine is to be executed. In such cases (1) the input data-parameters to the subroutine and the output parameters must be clearly specified; and (2) the relationship between the input and output parameters must be specified or else the data dependencies will be lost. In many cases SDL does not provide this information. Uncovering multiple seed attacks such as Collaborative and Multi-stage attacks requires knowledge of the input-output parameters and their relationships. If this information is present in the standard SDL specifications (action attribute of Table 2) then the attacks are discovered. If they are not present in the the standard SDL specifications, these attacks may not be discovered.

4.6 Algorithms

CAT algorithms may be used by 3G System Administrators wanting to detect vulnerabilities in scenarios, such as 'inability of subscriber to hear the voice of a caller in incoming calls'. The goal is known i.e. garbled voice of caller and hence the *seeds and goals* may be assigned: *Goal – Bearer Service* (Bearer Service indicates the type of voice transmit signals between subscribers) and *Seed – ISDN Bearer Capability* (ISDN Bearer Capability is one of the many factors used to determine Bearer Service). The goal is a clue to detecting the major problem i.e. *corrupt Bearer Service*. The output of using the *Goal – Bearer Service*; *Seed – ISDN Bearer Capability* in the algorithm is shown in Fig. 4. With more seeds, the algorithm will produce a much more detailed attack graph signifying that there are many ways to reach a goal. Note that by exhaustively trying every possible goal and every possible set of seeds, CAT can be easily extended to automatically identify all the possible (detectable) attacks associated with a certain service such as the *call delivery service*.

Algorithms take seeds as input from the user. Based on the input provided messages and actions of SDL are explored for generating attack graphs. The following conditions must be satisfied to generate and connect nodes in the attack graph. (1) If the seed occurs in messages or actions of SDL, then the condition nodes (Nodes A-D in Fig. 4) and adversary action nodes (Node E) may be generated and connected. (2) When a corrupt seed occurs in a incoming message of a block (action node) and the same seed occurs in the action of the

block (action node), it may be assumed that the corrupt seed in the message is used by the block hence corruption spreads from the message to the block (indicated by edge connecting Node E and F). (3) When a corrupt seed occurs in the action of the block (action node - Node F) and another seed or the goal itself occurs in the same action, it may be assumed the corrupt seed corrupts the other seed/goal (action node - Node K) leading to the generation of the edge connecting the two nodes. (4) When a corrupt seed or goal occurs in the action of the block (action node - Node K) and the same seed or goal occurs in the outgoing message from the block (action node - Node M), it may be assumed that the corrupt seed/goal in the block spreads to the message, leading to the edge connecting the two nodes.

1. Exhaustive Forward Exploration Algorithm The Forward Exploration algorithm starts 'bottom-up' from the seed and works towards finding the goal. It is exhaustive and checks each and every possible tuple in the SDL database. The Forward exploration algorithm to detect multi-stage attacks is detailed in Algorithm 1 displayed in Fig. 6a. The algorithm works in two phases.

Phase-1: Building Sub Graphs: In this phase, attacks are detected and attack graphs are built based on seed values and types of attacks. This phase also takes care of garbage collection and assigns levels to aid in pruning the graph. Garbage collection is performed when it is found that the seed fails to reach the goal. Levels are assigned for pruning the graph. The goal node is at level X and nodes 'X' edges away from the goal at level 0. Fig. 4. clearly illustrates the concept of levels. This phase is exhaustive and stops when (i) the seed reaches the goal, or the seed has not reached the goal but (ii) it has stopped propagating, or (iii) all messages upto the terminating message are explored.

Phase-2: Integrating and Pruning Sub Graphs: This phase integrates and prunes the graphs built in the previous phase. Low redundancy attack graphs are constructed by collapsing similar nodes and paths at the same level into a single node and path. The Forward exploration is a good approach if the seed actually reaches the goal. Issues arise when the seeds fails to reach the goal.

When the seed cannot reach the goal, the algorithm must *explore all the data sets till the terminating message in that sequence can be reached (via cascading effects)*. This reduces the efficiency of the algorithm. Another issue that arises frequently is the *appearance of loops when the seed fails to reach the goal*. Loops appear frequently due to the unique nature of the SDL data. Looping conditions must be checked at various points to avoid this problem. In the next section the Heuristics based Mid-Point approach is presented. This algorithm does not have some of the efficiency and looping problems evident in the forward exploration algorithm.

2. Heuristics Based Mid-Point Algorithm: This algorithm works 'bottom-up' from the seed and 'top-down' from the goal and terminates when the seed nodes from the bottom meet the goal nodes from the top. When seeds do not meet the goal the given heuristics may be used as a terminating condition to make the

Algorithm 1 Multi-Stage Forward Exploration Algorithm

```
{Seed} : Set of Input Seeds; {Goal} : Goal
for every s_i ∈ {Seed} loop
    for every CEFSM ∈ {SDL} loop
        if s_i in Message .or. Action then
            Add(PA, Tgt,Vul),(Access: Tgt) to Tree
            Call Find-Multistage-Tree(s_i)
        end if
    end loop
end loop
procedure Find-Multistage-Tree (s_i) is
begin
    while true loop
        for every E_i ∈ {Connected − Entity} loop
            if any {Seed} in Action then
                Add(Corrupt Seed) to Tree
                Add(Spy for Next Element) to Tree
            end if
            if {Goal} in Action then
                Add(Corrupt Goal) to Tree
                break
            end if
        end loop
    end loop
end Find-Multistage-Tree
```

(a) Forward Exploration

Algorithm 2 Heuristic Based Mid Point Method

```
{Seed} : Set of Input Seeds; {Goal} : Goal
for every s_i ∈ {Seed} loop
    if !Check-Heuristic(s_i) then
        continue
    else
        if s_i in Message/Action then
            Add(PA,Tgt,Vul) to Tree
        end if
        if Goal in Message/Action then
            Add(Incorrect Goal) to Tree
        end if
        GoalReached=Compare(s_i,SeedElement,GoalElement)
        while (! GoalReached) loop
            track-seeds(s_i) /*tracks the flow of the seed */
            GoalReached=Compare(s_i,SeedElement,GoalElement)
            if !GoalReached then
                /*tracks the flow of the goal */
                track-goal(s_i)
            end if
            GoalReached=Compare(s_i,SeedElement,GoalElement)
        end loop
    end if
end loop
procedure Compare (s_i,se, ge) is
begin
    if se==ge then
        Add(Corrupt Seed s_i Corrupts Goal) to Tree
        return true
    else
        return false
    end if
end Compare
```

(b) Heuristic Based Mid-Point

Fig. 6. Algorithms

algorithm efficient. Algorithm 2 displayed in Fig. 6b shows the Heuristics Based Mid-Point method.

As the network semantics and the SDL data are already known, it possible to assign heuristics as a terminating condition. The following are the heuristics: (1) *Limit Node Heuristics* states that if the number of nodes in a graph exceeds a set number of nodes (E.g. 20 Nodes) the algorithm must terminate. This is impractical and may not detect attacks with lengthy paths. (2) *1-Level Indirect Attack Checking Heuristic* states that if the goal and the seed appear together in any action and the output signaling message has the goal then the seed causes 1-Level Indirect Attack. However this heuristic is not sufficient to eliminate seeds that do not cause N-Level Indirect Attacks. (3) *N-Level Indirect Attack Checking Heuristic* checks each seed with every other seed and the goal for 1-Level Indirect Attacks. A seed may be eliminated from N-level indirect attacks if any of the other seeds it affects fails to reach the goal. This approach is exhaustive and would require N^2 database queries and is hence inefficient. (4) *Parameter Based Heuristic*: contains lists with sets of related and unrelated data parameters. The list of related parameters contains pairs of associated data parameters. For example, if data X is used to derive data Z, and data Y is not involved in computing the Z, the pair 'X-Z' belongs in the related parameters list and the pair 'Y-Z belongs in the unrelated parameters list. If the seed-goal pair occurs in a 'related parameters' list, the algorithm is executed, else the algorithm is

not executed. The advantage of this approach is that it does not require any computation and reduces execution time.

It is not possible to have the perfect heuristic. There is always the problem of mis-detection resulting in Missed seeds and Over-Seeking seeds. Due to inadequate heuristics, seeds that actually reach the goal are assumed to fail to reach the goal resulting in *Missed seeds*. This may result in attacks not being detected and happens because it is not possible to have a complete and exhaustive list of 'related' parameters and 'un-related' parameters. This may be minimized by picking a heuristic that only checks for 'un-related' Parameters. *Over-Seeking seeds* are those that fail to reach the goal but are assumed to be able to reach the goal carry the algorithm works towards matching the seed nodes and the goal nodes. This may result in unnecessary seeking of the goal with a large number of extraneous nodes and increase in execution time. Thereby reducing the efficiency of the algorithm.

4.7 Results

This section explains some representative results produced by CAT.

1-Level Indirect attacks: $Seed_1 \rightarrow Goal$. *$Seed_1$: ISDN BC → Goal:Bearer Capability.* Corrupting the ISDN BC in the IAM message leads to incorrect calculation of the Bearer Capability displayed in Fig. 4.

N-level indirect attacks: $Seed_1 \rightarrow Seed_2 \ldots Seed_i \rightarrow Seed_j \rightarrow \ldots \rightarrow Seed_n \rightarrow Goal$. *$Seed_1$: Alerting Pattern →$Seed_2$: Pre-paging support →Goal:Page type.* Corrupting the Alerting Pattern in the IAM signaling message leads to the retrieval of incorrect Pre-paging support at the GMSC and hence Incorrect Pre-paging support is provided in the SIFIC signaling message to the VLR. At the VLR incorrect Page type is calculated and the subscriber cannot receive calls.

4.8 Performance Analysis

The performance of algorithms is evaluated on the execution time by varying the *Seed Failure Ratio*. *Seed Failure Ratio* may be defined as the ratio of number of seeds failing to reach the goal to the total number of seeds used by the algorithm. The results are shown in Fig. 7. The algorithm is executed with a single goal, 8 seeds and heuristics varied to match the Seed Failure Ratio. When the Seed Failure Ratio is 0, i.e., all the seeds reach the goal, it is found that all the algorithms perform in a similar manner. With *0% mis-detection* (exact heuristic matching) the Mid-Point algorithm performs best and results in up to a 68% decrease in experiment time, when compared to the Forward exploration algorithm. With the *50% mis-detection rate* the Mid-Point algorithm results in up to 11% decrease in time than when compared to the Forward exploration algorithm.

5 Discussion and Conclusion

The introduction of IP based services into 3G networks increases the possibility of attacks on 3G network. CAT has been developed to detect all possible 1-Level and N-Level Indirect attacks. As the SDL specifications do not capture the data dependencies it is not possible to produce an exhaustive attack graph for multi-seed attacks with strong relationships and dependencies not exhibited in the standard SDL specifications. In the future the existing SDL specifications will be augmented with expert input to capture missing data relationships. Research work will be conducted on detecting new and unknown attacks and finding techniques to reduce the vulnerability of 3G network.

Seed Failure Ratio	Forward Exploration (milli secs)	MidPoint Method 0% Mis-detection Time in milli secs (% Time Saving)	MidPoint Method 50% Mis-detection Time in milli secs (% Time Saving)
0	6224	6185 (1%)	6150 (1%)
0.25	6120	5688 (7%)	5612 (8%)
0.5	6011	4903 (18%)	5567 (7%)
0.75	5295	2017(62%)	4897 (8%)
1	4064	1290 (68%)	3637 (11%)

Fig. 7. Time Analysis of Performance

References

1. K. Kotapati, P. Liu, Y. Sun, T. F. LaPorta, A Taxonomy of Cyber Attacks on 3G Networks, in Proc. IEEE Intl Conf. on Intelligence and Security Informatics (Extended Abstract), 2005. Lecture Notes in Computer Science, Vol. 3495, Springer-Verlag, 2005
2. Third Generation Partnership Projects (3GPP and 3GPP2), http://www.3gpp.org/
3. O. Sheyner, J. Haines, S. Jha, R. Lippmann, J. M. Wing, Automated Generation and Analysis of Attack Graphs, Proceedings of the 2002 IEEE Symposium on Security and Privacy, p.273, May 12-15, 2002
4. 3G TS 21.133 V3.1.0 (1999-12) 3G Security; Security Threats and Requirements version 3.1.0
5. 3G TR 33.900 V1.2.0 (2000-01),A Guide to 3rd Generation Security
6. 3G TS 33.120 V3.0.0 (1999-05) 3G Security; Security Principles and Objectives version 3.0.0
7. P. Howard, M. Walker, T. Wright, Towards a coherent approach to third generation system security, Second International Conference, 3G Mobile Communication Technologies, 2001. on (Conf. Publ. No. 477)
8. N. A. El-Fishway, M. A. Nofal, A. M. Tadros, An Improvement on Secure Communication in PCS,Performance, Computing, and Communications Conference, 2003. Conference Proceedings of the 2003 IEEE International , 9-11 April 2003
9. C. C. Lo and Y. J. Chen, Secure communication mechanisms for GSM networks, IEEE Transactions on Consumer Electronics, Vol. 45, No. 4, pp..
10. D. Welch, S. Lathrop, Wireless Security Threat Taxonomy, June 2003 IEEE Workshop on Information Assurance.
11. C. Clissmann, A. Patel, Security for mobile users of telecommunication services, Universal Personal Communications, 1994. Record., 1994 Third Annual International Conference on , 27 Sept.-1 Oct. 1994,Pages:350 - 353.
12. C. B. Brookson, Security in current systems, IEE Colloquium on Security in Networks (Digest No. 1995024), 3 Feb. 1995, Pages: 3/1 - 3/6.
13. C. J. Mitchell, Security techniques, in Proceedings of the IEE Electronics Division Colloquium on Security in Networks, London, February 1995, IEE (London) Digest No: 1995/024, pp. 2/1-2/6.
14. K. Boman, G. Horn, P. Howard, V. Niemi, UMTS security, Electronics & Communication Engineering Journal, Volume:14, Issue:5, Oct. 2002, Pages:191 - 204
15. L. P. Swiler, C. Philips and T. Gaylor, A Graph-Based Network Vulnerability Analysis System, SandiaReport, SAND97-3010/1, January 1998, Sandia National Laboratories, Albuquerque,New Mexico, U.S.A., 1998.
16. C. A. Phillips, L. P. Swiler, A Graph-Based System for Network-Vulnerability Analysis, Proceedings of the 1998 Workshop on New Security Paradigms (NSPW'98, Charlottsville, VA, USA), pp. 71-79, ACM Press
17. L. Swiler, C. Phillips, D. Ellis, S. Chakerian, Computer-Attack Graph Generation Tool, in Proceedings of the DARPA Information Survivability Conference and Exposition II, June 2001.
18. R.W. Ritchey and P. Ammann, Using model checking to analyze network vulnerabilities. In Proceedings 2000 IEEE Computer Society Symposium on Security and Privacy, pages 156-165, Oakland, CA, May 2000.
19. P. Ammann, D. Wijesekera, S. Kaushik, Scalable, graph-based network vulnerability analysis, Proceedings of the 9th ACM conference on Computer and communications security, November 18-22, 2002, Washington, DC, USA
20. S. Jha, O. Sheyner, J. Wing, Two Formal Analys s of Attack Graphs, Proceedings of the 15th IEEE Computer Security Foundations Workshop (CSFW'02), p.49, June 24-26, 2002
21. S. Jha, O. Sheyner, and J. M. Wing, Minimization and Reliability Analyses of Attack Graphs, CMU-CS-02-109, February 2002. Detailed version of paper to appear in Computer Security Foundations Workshop, Nova Scotia, June 2002.
22. O. Sheyner and J. Wing, Tools for Generating and Analyzing Attack Graphs, Proceedings of Formal Methods for Components and Objects, Lecture Notes in Computer Science, 2005.
23. B. Schneier, Attack graphs, Dr. Dobb's Journal, pp. 21-29, December 1999
24. J. Ellsberger, D. Hogrefe, A. Sarma, SDL, Formal Object-oriented Language for Communicating Systems,Prentice Hall, 1997, ISBN 0-13-621384-7, 312 pp.
25. 3GPP TS-23.018 (v3.4.0) Basic Call Handling - Technical realisation, April 99