450 Report: Formal Verification of 5G Protocols

Alex Grimes, Aaron Nelson, Lyon Zhang, Kaiyu Hou, Yinbo Yu

ABSTRACT

Prior work, LTEinspector, applied formal methods to 4G LTE protocols and found 10 new attacks in addition to 9 known attacks. Nevertheless, LTEinspector only specified and verified the 4G NAS protocol. By leveraging its authors' basic methodology, we improve its test steps and apply the process to the emerging 5G system. We specify three important protocols: radio resource control, emergency handling, and charging protocols, which have not been carefully formally studied before. Vulnerabilities of these protocols are exposed after checking the real world viability of properties. Finally, we discover 7 potential attacks and propose real world testing plans for further research.

KEYWORDS

Cellular Network Protocols, 5G Vulnerability, Formal Verification

1 INTRODUCTION

1.1 Prior Work

Substantial work has gone into analyzing the security and privacy of telecommunication systems. The LTE network has seen substantial growth over the past few years, and research to identify design flaws and weaknesses of the 3GPP standard has developed as well. Work has also been done to identify some of the unsafe practices carried out by stakeholders that puts the security of the system at risk.

The work is done in CNetVerifier [6] utilizes a method for analyzing the soundness of 3G and 4G control plane protocols by first representing the protocols in question with a state machine. Then, once the state machine is complete, the user defines desirable properties and runs permutations of possible model scenarios to find situations in which these properties are violated. Using this methodology, the researchers found six vulnerabilities in control-protocol interaction across three dimensions.

The work described in the LTEInspector [4] presents an improved method for analyzing LTE protocols that employ a property-driven adversarial model-based testing philosophy. Their approach leverages the combined power of a symbolic model checker and a protocol verifier. The authors define a symbolic model for the LTE system and inject a standard symbolic adversary model. The model-checker can then attempt to find a violation of a defined property in the system model. Using this approach, the authors found 10 new attacks in addition to 9 known attacks in the attach, paging, and detach procedures in the LTE protocols.

1.2 Motivation and Challenges

The cellular network system is one that people rely on every day to go about their daily lives. In addition to convenience, government agencies rely on this network to broadcast important messages like weather warnings and AMBER alerts and emergency services rely on the network to provide essential services to those who need immediate help. Exploits of vulnerabilities in this system could lead to inconvenience, widespread panic, or even loss of life.

Most prior work at that applies formal methods to find vulnerabilities in 4G LTE protocols uses the 3G and 4G systems to base their research. Many vulnerabilities and attacks were uncovered in these older generations of the system, but the cellular network system is always evolving. The newest cellular network system, 5G, which aims to solve a lot of security issues of the prior generations as well as provide better and faster service. Formal methods can be used to verify that vulnerabilities in earlier generations have been patched and to uncover new attacks that were introduced as a result of added features and protocol complexity.

There are a few challenges that make research in this area so difficult. The cellular protocols lack formal specification and the standard suffers in many areas from ambiguity. The protocols have many levels of abstraction and interweave functional and cryptographic protocols. The protocols are stateful in nature and need to scale in order to accommodate many participants, many different message types, and many different types of participants. Most of the deployed cellular networks are closed and proprietary, which requires any testing approach to be system-agnostic. Laws prohibit testing and carrying out real-life attacks in commercial networks, therefore requiring much more overhead in order to validate potential attacks.

1.3 Contributions

Notable among our findings are the cell barred attack, which allows an attacker to set up a malicious eNodeB and force a victim UE to attach to it. Through this attack, the malicious eNodeB can launch a DoS attack on the victim UE, causing it to be temporarily barred from the network. Another notable attack discovered is the emergency number MITM (Man In

The Middle) attack, which potentially allows an attacker to cause congestion and confusion at the public-safety access point, therefore denying a victim regional emergency calling. A third and final attack is the forced disconnect DoS attack, in which an attacker spoofs a message to the network charging architecture and forces the user to disconnect from the network. Through this attack, the attack can deny downstream packets to the user and can potentially even cause erroneous charging.

In summary, we make the following contributions:

- We developed 3 models that are available for expansion for the RRC protocols, emergency protocols, and charging protocols, respectively.
- We show the effectiveness of our approach through the discovery of 7 potential attacks.
- We contribute a new tool for the NuSMV and formal methods community that aids in the speed of development, ease of understanding, and customizability of state machines.

2 METHODOLOGY OVERVIEW

We model our methodology after the authors of LTEInspector. In particular, we make use of NuSMV for model checking and follow a similar procedure for constructing models of various aspects of 5G protocols.

2.1 Tools

The main tool utilized in our analysis is NuSMV, which is a popular, open source model checker. The software reads a file detailing various variables and behaviors, then mutates variables so as to check the correctness of any logical properties which are defined. Several variable types are used, including boolean environment and state variables, frozen variables and enumeration variables.

We specify state machines and their transitions in a manner which is nearly identical to that of LTE- Inspector. In fact, so as to facilitate the rapid accurate development of our models, we take advantage of LTEInspector's XML to SMV parser, which is specifically designed for simulating state machines and unidirectional channel communications in NuSMV.

While LTEInspector's parser is useful for its intended purpose, it fails to expose several useful features of NuSMV in its abstracted development layer. In particular, frozen variables are a tool which allows for the specification of varying, static interpretations of the selected 5G protocols. Since this feature is not available in LTEInspector, we modified the parser to expose this functionality. Additionally, while LTEInspector's XML format is more intuitive than the native SMV format, it is still not trivial to develop in. Several nuances exist in the XML format which works against rapid prototyping, such as enumeration of state transitions and "null action" events.

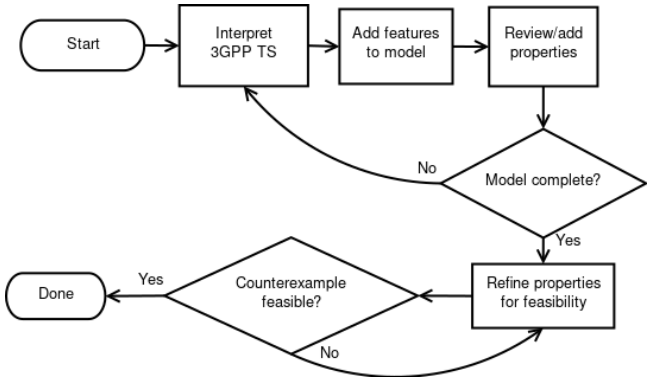

Figure 1: Overview of modeling process.

To reduce the burden of these nuances, a custom YML to XML parser was developed, which greatly improves the readability and development speed of the state machines. Since YML is a strict superset of XML, a naive solution to this simplification is to use existing converters. However, our solution takes advantage of a unique, more intuitive structure, and hides some of the nuances of LTEInspector's format under a layer of abstraction.

2.2 Main Steps

By leveraging the basic methodology of LTEInspector, the protocol specification and verification steps are shown in Figure 1.

- *Read/Interpret 3GPP protocols*: Several technical details are considered and interpreted. This step is particularly time consuming, since each protocol is specified across several documents. The individual aspects of each protocol and their interactions in the overall model must be carefully considered before beginning the construction of the model.
- *Add features to model*: Features are determined after consideration of the technical specifications. In this step, either the state machines representing the protocols are constructed in accordance with the format of the tools.
- *Review and refine model*: Several issues can arise from the abstraction required in the construction of the model. It is essential to carefully review the state machines' soundness in accordance with the model.
- *Create and refine properties*: Properties are used both to verify the soundness of the state machine and to reveal potential vulnerabilities. There are two types of properties, liveness, and security. Liveness properties ensure that the intended functions of the model are

always able to complete, while security properties ensure that certain improper behaviors are not possible in any circumstances.

- *Inspect NuSMV counterexample for the correctness*: Due to both the abstraction inherent to the methodology and the format of the model (state machines dynamics are not perfectly representative of real behaviors), it is essential to verify that any detected property violations are feasible. In the event that such a violation could not be reasonably done in a real network, either additional detail must be added to the state machine, or constraints must be placed on the properties. After this step's completion, any remaining counterexamples generated by NuSMV should be considered for practical testing.
- *Validate counterexamples in the real world*: While real-world testing is outside the scope of our work, we lay down the foundations for testing discovered potential vulnerabilities. This step is of significant importance since vulnerabilities in an abstract model are useless if they cannot be replicated in reality.

3 RADIO RESOURCE CONTROL PROTOCOL

3.1 Background

The Radio Resource Control (RRC) protocol layer controls communications between a UE and an eNodeB (also referred to as a cell in this paper). These protocols operate in the highest layer in the control plane of the Access Stratum (AS). The RRC also transfers messages of the Non-Access Stratum (NAS), which is located above the RRC layer. The messages in the NAS layer are used to control messages between a UE and the Evolved Packet Core (EPC). As a part of controlling the communications between the UE and the eNodeB, the RRC protocols are also responsible for the set-up and tear-down of connections when the UE wants to become connected to the network, handling mobility procedures when the UE moves from one cell to another, and initiating paging notifications when the UE receives an incoming notification.

As the cell boots up, it must initiate the process of selecting a cell, documented in TS 36.304 [1], which enables the UE to begin moving through different RRC states.

The UE and the eNodeB both transition through a series of RRC states, which are documented in TS 36.331 [2]. The two states that are relevant for this paper are the RRC_IDLE and RRC_CONNECTED states. When the UE starts up, will eventually transition from the RRC_IDLE state to the RRC_CONNECTED state as it selects a cell to connect to and connects to that cell. The eNodeB also has RRC_IDLE and RRC_CONNECTED states among others, but the focus of this paper will be on the two states.

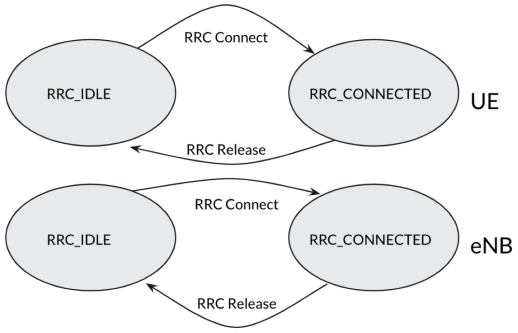

Figure 2: Simplified UE and eNodeB FSMs.

In order to transition from the RRC_IDLE to the RRC_CONNECTED states, a series of messages need to be passed between the UE and the eNodeB. The MIB and SIB messages must be sent by the eNodeB and successfully received by the UE for the connection to be successfully set up and established.

3.2 Motivation and Intuition

The RRC protocols control essential parts of the life cycle of a UE as it starts up, finds an eNodeB to connect to, establishes a connection, sends and receives data, and finally releases the connection. These protocols are involved when any user, regardless of their device, has to use the services provided by mobile internet connections.

The main motivation for analyzing these protocols is that other protocols and functions rely on protocols in this layer being correct and sound. If there are issues and vulnerabilities with these protocols that can be exploited by an adversary, they can cause bigger issues later in the life cycle of the connection. Potential issues include DoS attacks, issues with impersonation, and issues with connection to illegitimate networks.

The main intuition for finding vulnerabilities in these RRC protocols is the lack of encryption in the messages that are sent through the protocols. Firstly, this gives attackers the ability to intercept and read all of the messages that are broadcast by either a UE or an eNodeB. With knowledge of the messages that are being passed around in these protocols, attackers are able to modify messages to fit their agenda. Attackers can also spoof their own messages and inject them into a channel towards a victim device. With knowledge of the protocols, and without any cryptographic knowledge, an attacker could do a great deal of damage. The combination of these three actions gives attackers the power to read the message to get the information they need and spoof seemingly legitimate messages of their own.

3.3 Specification

The protocols for cell selection and connection of a UE to an eNodeB can be modeled by their two participants: the UE and the eNodeB. While during cell selection there can be multiple cells competing for the UE connection, we take the cell with the strongest signal to be the one represented in the model, and therefore the cell that attempts to connect with the UE. Additionally, we only consider AS layer messages in our model, as analysis of NAS layer messages has already been performed in prior work.

We abstractly model a portion of the 5G LTE procedures that are relevant to cell selection and UE connection, without fine-grained implementation details, as two synchronous communicating finite state machines (FSM), shown in Figure 2. The FSMs communicate with each other by sending messages through two unidirectional public communication channels. To bound the message domains, we model information that can be sent in a message with environment variables. For example, the connection_ release message can contain cell_redirection_info, but we simply model the message type as connection_release.

Additionally, all of the states for every process does not need to be included to have an accurate model of the system. For example, the states outlined for cell selection in TS 36.304 include more states than that one needed for a typical cell selection after UE reset. Including all of the extraneous detailed states may not be necessary for finding an attack or vulnerability, and therefore do not need to be considered in the initial model.

3.4 Property Design

Properties are designed with one of two cases in mind: the system never reaches a bad state or the system always can move forward and reach a desired good state. This is the main basis for designing our properties. The properties are identified using the inspection of the technical specifications.

For the RRC state machines, we identified two properties we believe should hold at all points during the execution of the system.

- $p1$: When the UE is connected, we do not want it to ever contain malicious cell redirection information.
- $p2$: We always want the UE to be able to connect to an eNodeB, because otherwise would present an issue for a DoS attack.

3.5 Verification Results

When testing the properties p1 and p2 against the model with an injected adversary, we find some interesting counterexamples.

Counterexample 1: The property that we wanted to verify is that when the UE is connected, it should never contain malicious redirection information. When this property p1 is checked against the model with an injected adversary, it yields an interesting counter- example. When the UE moves into the RRC_CONNECTED state, and the eNodeB decides to release the connection, the attacker can inject their own connection_release message containing malicious cell redirection info. This causes the cell to have malicious redirection information the next time it becomes connected, violating our property.

This counterexample demonstrates the RRC Redirection Attack, originally discovered and shown feasible in [3], so it is not discussed further in this work.

Counterexample 2: The property that we wanted to verify is that the UE should never be barred, which follows that if the UE selects a cell, it should eventually become connected. When checking this property against the model, we find the basis for a potential new attack. After the UE selects a cell to camp on, the connection setup messages begin. The attacker can inject their own MIB and SIB messages with category0_devices_allowed flag set to false. The UE, if it is a category 0 device (an IoT device) will then entered a barred state where it cannot access the network for 300 seconds.

3.6 Potential Attacks

We now present the new potential attacks that we discovered as a result of our counterexamples, and the implications that they present.

Cell Barred Attack. This attack exploits the fact that the messages in the RRC protocols are not encrypted and therefore attackers can spoof legitimate messages from malicious base stations.

Adversary Assumptions. For successfully carrying out this attack, the adversary needs to set up a malicious eNodeB and can collect the identification information (PSS and SSS) from a legitimate eNodeB. Because the messages that contain these identifiers are regularly broadcast without encryption, this is a reasonable assumption.

Detection. We exposed this attack by the first model checking the model with the injected adversary with respect to the property p2. We observed a violation of p2 in the model where the adversary broadcasts fake PSS/SS information to spoof a legitimate eNodeB and then fabricates MIB and SIB messages and sends them to the UE. To validate the adversary's capability to forge these messages, we leverage our domain knowledge to conclude that the cryptographic constraints are met because of the lack of encryption in these protocols.

Attack Description. The steps of this attack are shown in Figure 3. It is similar to the description of the counterexample 2 described above but with more reasoning about how the adversary obtains the information to carry out the attack.

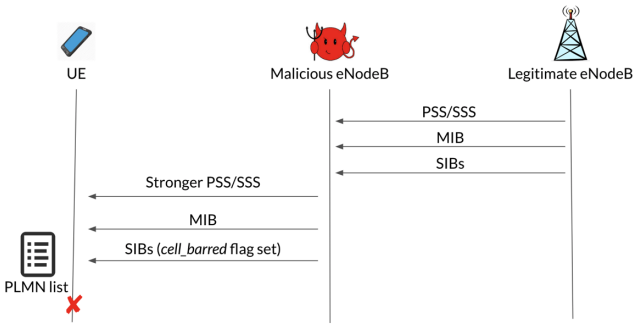

Figure 3: Cell Barred Attack.

The adversary, once having set up their malicious eNodeB, can obtain the PSS and SSS identification information from messages broadcasted by a legitimate eNodeB. It is possible to obtain this information because of the lack of encryption in these messages. The malicious eNodeB can then broadcast these copies of the PSS/SSS with a stronger signal, which a victim UE will pick up because the eNodeB identification is legitimate. The UE will then become camped on this malicious cell.

The eNodeB can then send the MIB message that was obtained from the legitimate eNodeB and send this message to the UE, where the UE will receive this message successfully because the malicious eNodeB is simply spoofing the legitimate eNodeB messages. In order to carry out the last step of the attack, the malicious eNodeB spoofs SIBs messages with the same method as for the MIB, but instead of simply using the same messages, the attacker can modify the SIB message that contains cell barred information to cause the UE to be blocked from the eNodeB. One particular case of this cell barred information is where the cell can block IoT devices, so if the UE is an IoT device, the UE will be barred from connecting to the cell. The UE will then try cell selection again, this time using a different PLMN from its PLMN list, but most UEs only contain 1 PLMN (their home network) from which they are now barred and have no other alternatives for cell selection.

Implications. The most significant implication of this attack is the cell is barred from connecting to the cell for 300 seconds. This cell is in the one PLMN that the cell stores in its list for cell selection, so in many cases, this means that the UE is unable to connect to the internet for these 300 seconds. While this 300 seconds may simply be a nuisance in some cases, in others it could prevent a user from calling the emergency or geolocation services they need.

3.7 Real World Validation/Testing Plan

In order to validate our potential attacks, an essential next step is to test the validity of the presented attacks in a test bed. Because of cost and time constraints, we did not validate the attacks ourselves, but instead include this testing plan for future work.

To show the validity of this attack, there need to be three participants: the legitimate eNodeB, the malicious eNodeB, and the victim UE. The legitimate eNodeB should broadcast its PSS and SSS identifiers to let potential UEs know that it is available for connection. The malicious eNodeB should then eavesdrop on these messages by listening on the same frequency at which these messages are broadcast. The malicious eNodeB can then broadcast this message with a stronger signal than the legitimate eNodeB.

Then the UE should be turned on, and it performs cell selection and will become camped normally on the malicious eNodeB (because of the strong signal). The eNodeB can then obtain the MIB as described above and send it to the UE. Once the UE has received this message, the SIB messages can be sent along as well, but with one of the cell barred flags set. This should then disconnect the UE from the network for 300 seconds.

3.8 Future Work

In addition to the work that can be done with real-world validation and executing the testing plan, we would like to present some areas for future work. One of the weaknesses of this work is that the finite state machines only cover the most general cases of cell selection and UE connection at an abstracted level. For future research, adding states, variables, and messages to the FSMs to dive deeper into the abstractions could lead to more vulnerability discovery. Exploring more properties that hold could also reveal more counterexamples and potential attacks.

4 EMERGENCY PROTOCOLS

4.1 Background

Emergency calling is an essential component of all evolutions of cellular platforms. There are two main paradigms which separate emergency calling from other types of calls:

- Emergency calls take priority over all other traffic
- Emergency calls are possible even from non- subscribing clients

These traits are vital, since emergency calls are often the only thing separating individuals from life and death. Emergency calling is, by design, a relatively insecure component of the 5G specification. This is done not only for legal and liability-driven reasons, but also for ethical principles: there is no greater failure as a carrier than to indirectly cause loss of life.

The manner in which emergency calls work is conceptually somewhat simple (in comparison to other elements

of the specification). If a user is not registered to a network and wishes to place an emergency call, their equipment will attempt emergency registration with any carrier (regardless of who their home carrier is). This registration takes priority over normal registrations, and will even disconnect subscribers if there are not enough network resources to accommodate the emergency session [5]. After the registration, or if the user is already a subscriber of the network in question, the equipment will attempt to initialize an emergency call with the core network. This call carries both a target phone number (such as 911 in the United States) and an emergency flag. Then, the core network determines how to route the call to the most appropriate Public Safety Answering Point (PSAP) so as to ensure legal compliance and a speedy emergency response.

4.2 Motivation and Intuition

There exist some elements of the emergency calling procedure which suggest the possibility of vulnerabilities. First, since the emergency registration process is deliberately possible without authentication, it may be possible to gain access to the network without having any valid subscription to the target network. This requirement additionally removes any cryptographic burden from attackers, at least so far as connecting to the network goes.

Second, since there are at least two factors which the core network uses to determine the routing of emergency calls (the phone number and the emergency flag), it may be possible to abuse combinations of these factors to elicit improper behavior.

In addition to these technical aspects, a PSAP is a particularly interesting target for terror actors or similar groups, as manipulating or denying service to a PSAP can cause confusion, failure of emergency responses, and, in a worst-case-scenario, even mass loss of life.

4.3 Specification

While the technical details for the 5G emergency protocols are nuanced, we can model them in an abstract way by segmenting elements of the interactions into two main elements: the User Equipment (UE) and the Core Network (CN) [1].

Several simplifications are made in the abstraction. Specifically, the model does not consider the existence of competing networks which may be able to provide emergency service to a user. The model also simplifies the communication between the UE and CN into two unidirectional channels without sequence numbering, while real implementations are much more complex. In addition, when modeling an adversary, we make the assumption that it has full cryptographic capabilities as well as the ability to prevent messages from reaching their intended recipients.

Both communications between the UE and CN, and interactions within state machines are also simplified. Since NuSMV is not intended to model complex, stateful messages, each transition and channel message in the model is reduced to an abstract concept. For instance, instead of programming the CN's handling of emergency numbers directly, we instead use immutable variables to dictate certain decision processes and state modifications, such as whether or not the number being dialed is recognized as an emergency number by the CN.

Despite these abstractions, the model is nonetheless an accurate depiction of the emergency calling procedure, albeit with less detail. For this reason, we can assume that only high-level vulnerabilities can be discovered by the model checker. That is to say, it is reasonable to expect finding a procedural error which may lead to a bad state, but the model cannot hope to reveal implementation errors or issues with any protocols which themselves constitute the basic capabilities employed by emergency calling.

4.4 Property Design

In the case of emergency calling, most properties are somewhat intuitive. The properties specified in this work generally follow the following principles:

- Emergency calls should always be allowed to be attempted
- Emergency calls, if accepted, should always route to the PSAP
- Non-emergency calls, if accepted, should NOT route to the PSAP
- Non-emergency calls should be rejected from non-subscribers

4.5 Verification Results

After inserting the properties into the model checker, we are able to produce counterexamples which may suggest the existence of vulnerabilities in real networks

Counterexample 1: The property in question is (1) Emergency calls should always be allowed to be attempted. This property is only violated when the adversarial injection is enabled. In essence, this counterexample suggests that by repeatedly sending connection requests to the CN, it is possible for a situation to arise in which the CN is never able to process the emergency call.

Counterexample 2: The property in question is (2) Emergency calls, if accepted, should always route to the PSAP. This violation occurs when the UE places a call using the emergency flag (believing it is placing an emergency call),

[1] Note that in this abstraction, the eNodeB's actions are considered part of the Core Network

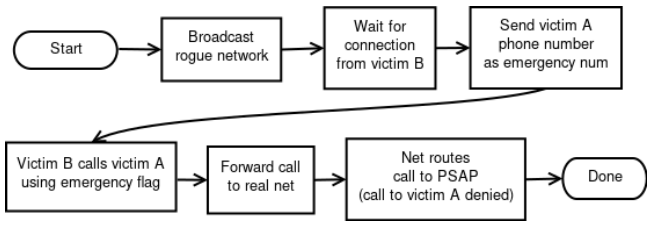

Figure 4: Process flow of PSAP Redirection DoS Attack.

but the CN ignores the emergency flag and instead routes the call according to the number dialed. This represents a security oversight in the CN, since such a call should be rejected.

Counterexample 3: The property in question is (3) Non-emergency calls, if accepted, should NOT route to the PSAP. The property is violated when an improper emergency number is pushed to the UE. The protocol allows networks to send local emergency numbers to devices (intended for international travel), but this can be abused, as it forces the UE to set the emergency flag. If the CN routes all flagged calls to the PSAP, this results in what appears to be a normal call from the user's perspective being answered by the PSAP.

4.6 Potential Attacks

After interpreting the counterexamples to the properties specified, three attacks were discovered, as are summarized in Table 1.

4.6.1 Blind DoS Attack. This attack was rediscovered as a side-effect of the modeling of registration in the emergency state machines. It was initially disclosed in 2019 by Kim *et. al* [5]. Since this attack has already been explored by others, we do not consider it further.

4.6.2 Free Calling Attack. This attack takes advantage of the lack of authentication in emergency registration and loopholes in carrier routing.

Adversary Assumptions: To carry out this attack, an adversary needs to have control over their own UE. This can be in the form of firmware or baseband manipulation, or access to custom SIM cards. No cryptographic assumptions are required.

Detection: This attack was discovered by observing an exception to emergency property 2 in Section 4.4. The counterexample trace indicates that it may be possible, in some situations, for a UE to carry out normal emergency registration, followed by abuse of the emergency call system to place a normal call.

Attack Description: First, the attacker must gain control of their UE, either in the form of rooting the mobile OS, writing a custom SIM, or even developing original hardware. After gaining this control, the attacker must signal their equipment to treat all calls as emergency calls. Upon placing a call, the equipment will attempt an emergency registration, which the CN is required to accept due to legal requirements in most countries. Then, a faked emergency call is placed. In some implementations of the CN, it may route this call to its normal routing destination without regard for the presence of the emergency flag. Upon successful incorrect routing, the adversary is able to place a call on the victim carrier's network without being a subscriber. A flow diagram of this process can be seen in Figure 6.

Implications: The implications of this attack do not initially seem severe. In the United States, unlimited calling can usually be legally obtained at a low cost. In this sense, it is not particularly valuable to obtain the ability to place free calls. This attack becomes more interesting when we consider the possibility of spam calling. While this method cannot produce the volume which is possible with contemporary VoIP-based spambots, it is immune to aspects of state-of-the-art spam detection, since these calls originate from trusted carrier networks.

4.6.3 PSAP Redirection DoS Attack. This attack takes advantage of part of the emergency specification which allows carriers to push emergency numbers to their subscribers. This functionality is useful in cases such as roaming to other countries, where the UE may not be familiar with local emergency numbers.

Adversary Assumptions: We assume that the adversary either has cryptographic knowledge of the system or is otherwise able to downgrade the encryption of the victim's connection. This has been demonstrated to be possible by various means, including the attack discussed in Section 3.6 and the work of [5]. Additionally, the adversary requires access to equipment which is capable of broadcasting a rogue 5G network, and knowledge of a second victim's phone number. Geographic proximity is also an essential component of this attack.

Detection: This attack was discovered by observing an exception to emergency property 3 in Section 4.4. The counterexample trace indicates that it may be possible, in some situations, for an adversary to deceive a UE into believing that an arbitrary phone number is actually an emergency number. In some implementations of the CN, it may be such that all calls to said phone number will instead be routed to the PSAP.

Attack Description: The adversary is required to set up a mechanism for downgrading the encryption between their first victim and the legitimate CN. Then, they will force the UE to transition its connection from the legitimate CN to their rogue network. This is possible in a variety of ways, one

Table 1: Summary of discovered potential attacks in emergency protocols.

Description	New	Adversary Assumptions	Consequences
Registration replay/fabrication	No	Known S-TMSI or cryptographic knowledge Geographic proximity	Total call failure
Improper emergency number configuration	Yes	Malicious UE Network does NOT route emergency flag to PSAP	Free calling for adversary
Emergency number MITM attack	Yes	Has cryptographic knowledge Network DOES route emergency flag to PSAP Geographic proximity	Deny regional calling to victim Confusion at PSAP (resource depletion)

of which is discussed in [5]. Upon receiving the connection from the UE, the adversary will set up a relay to the legitimate network, while sending a false emergency number list to the UE. When the UE attempts to place a call to any number on the false emergency list, it will attempt an emergency call. The legitimate CN then detects that this is an emergency call, and disregards the dialed number in favor of routing the call to the PSAP.

Implications: This attack may seem uninteresting at first, especially due to the assumptions required for its execution. However, since there are existing attacks which grant adversaries these capabilities, it is reasonable to assume that such an attack may be possible given a sufficiently motivated adversary. If executed correctly (and perhaps in combination with a more advanced version of the attack), this attack can be used to cause a large number of calls to be placed to the local PSAP. In many jurisdictions, it does not require a large number of calls to overwhelm emergency operators, which can be used to distract emergency responders from a legitimate incident. Of course, the worst-case scenario for this attack is redirection of resources from a terror attack. In such a case, it is likely that the response time of emergency responders would be increased considerably, which would cause vastly increased loss of life. It is important to note that this case is extremely unlikely, and would require a level of evolution on top of this attack, should it even exist in the real world.

4.7 Real-world Validation/Testing Plan

Testing for these attacks can be difficult, because they require utilization of emergency infrastructure. In the United States, abusing emergency calling is at best a misdemeanor, and is often a jailable crime. Therefore, testing these attacks without full cooperation of the carrier and/or the local PSAP is impossible to conduct legally. That said, certain legal loopholes can be used to do this testing. In particular, it is generally legal to test dialing the PSAP for purposes of verifying the success of emergency routing in VoIP or other non-traditional configurations. It is also typically possible to inform the PSAP of a test call, in which case the call is not a crime.

After securing the legality of testing, the attacks should be tested in accordance with their descriptions in Section 4.6. Since each attack is different, they need to be tested in distinct ways. Perhaps the easiest method of testing the first attack is to modify the emergency numbers listed in a phone's SIM card. By doing this, it is possible to test at least one carrier's handling of illegitimate emergency calls. In order to test multiple carriers, it is necessary to either test in locations for which only one carrier provides coverage, or to modify the baseband to prevent connections to specific carriers.

To test the second attack in this category, a rogue network must be established. This can be done using open source hardware/software in the case of LTE, but it is currently not possible to do so for 5G. Ethical considerations for this test include the aforementioned carrier/PSAP cooperation and limiting the rogue network's signal strength to a trivially small distance.

4.8 Future Work

In addition to real-world validation, future work could evolve on our contribution to the model for emergency calling protocols. Due to the level of abstraction employed in our development, there may exist several gaps in modeling resolution which could affect both the number and accuracy of detected vulnerabilities.

In particular, multimedia emergency protocols (such as SMS, MMS, and video calling) are presently seriously underexplored. While their adoption rates are very low in the real world, SMS emergency access is currently experiencing rapid growth in several regions in the United States. Since these formats are more complex, they are more difficult to model, but may present interesting possibilities in terms of exploitation as they become ubiquitous.

5 ROAMING CHARGING PROCEDURE

5.1 Background

Charging is a natural part of the cellular network and its correct management and implementation are essential to the

continuing agreement between user and data provider. To match the dynamic demands of cellular data usage across the world, there is not one way to monitor and charge data flow. In my work, I focus on PDU (Protocol Data Unit) Session Charging.

A Protocol Data Unit Session is an agreement between a user device and the serving data network that happens before and after any useful data is sent to and from the user device. It monitors data flow in terms of special data units (i.e. packets). Like TCP, a PDU session has to be established before the user receives any data, and this is accomplished via a multi-directional handshake. Similarly, any changes to the parameters of the session and the release of the session are completed through such handshaking.

The standard for roaming is for the data network that the user is visiting to contact the visiting device's home network. It then sets up a special tunnel for interaction between the traveling device and its home network.

5.2 Motivation and Intuition

In general, the most intuitively vulnerable spots in the PDU charging process are in the three major functions: establishment, modification, and release, or, in other words, the handshake between the user device and the network. The vulnerability comes from the fact that the network must behave dynamically in reaction to the flags, requests, etc. of the user, and many different actions are possible. In addition, the user and the network must stay in sync about the status of the session, otherwise a mismatch that could lead to denial of service or erroneous charging could occur.

Out of all different charging scenarios, roaming is one of the more fragile ones both in regards to protocol viability and real world implementation. This is because the visited network and the home network must communicate extremely long distances in order to provide service to the roaming user, which adds another layer of protocol interaction on top of already error-prone communication over longer distances. In addition, the high cost of roaming for both the home network and the user makes the correct management of charging even more important, as mistakes hurt one party even more than normal.

5.3 Specification

For the sake of time, this portion of the project models the PDU Session Establishment and PDU Session release protocols in the home-routed roaming scenario as described above. The state machine does not model any stochastic condition like packet corruption, and instead focuses only on the safety of the protocol itself.

These two protocols are lengthy (20 or so broad steps each), multilayer (most of the steps are further broken down into sub-steps and sub-protocols), convoluted (include interactions between many nodes), and nonlinear (certain steps can be skipped on certain conditions, or done out of order). These properties cause the complete state machine for this protocol to be impractically large for the manpower and time frame of this project. Thus, several simplifications needed to be made to produce an understandable yet sound and comprehensive state machine.

In order to simplify the state machine but still maintain the comprehensiveness of the protocol, the steps that were the most volatile and thus led to the most variability were kept, while all others were abstracted away. All sub-protocols were abstracted away and viewed as a black box that always gave the correct output. Because of the high number of nodes involved in the protocols, the nodes in the state machine do not correspond to the nodes of the charging network, but rather to checkpoints along the path to process completion. Finally, necessarily linear steps were abstracted into one big step that was assumed to function correctly internally.

Together, all of these abstractions come together in a state machine that has about half as many nodes as the number of steps in the protocol, contains bad states that should be avoided, and accurately models real protocol interaction.

5.4 Property Design

The important properties related to charging are self explanatory. The user and network must agree on the amount of data charged. Otherwise, standard security properties such as authentication and availability apply (i.e. charging protocols should not violate general security properties). Thus the following properties were defined:

- If the UE sends data upstream, it should receive a response from the network.
- If the session establishment is deemed successful by the network and the UE receives a notification as such, the UE should eventually receive data.
- If the PDU session should be terminated, the UE should eventually be informed of the release.

5.5 Verification Results

Originally, all three of the properties had counterexamples. However, closer inspection of the first counterexample and the 3GPP protocol showed that the problem in question was actually intended behavior and covered by 5G optimization. Thus these two counterexamples remain:

The first remaining counterexample corresponds to the second property defined above. This failure happens when PDU session establishment happens as expected, and the UE gets the notification that the establishment was successful. The UE then sends its first packets upstream, expecting a response. However, the network can decide that it has

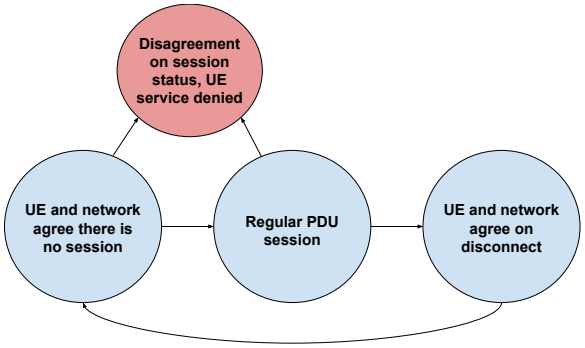

Figure 5: High level abstraction of roaming vulnerabilities.

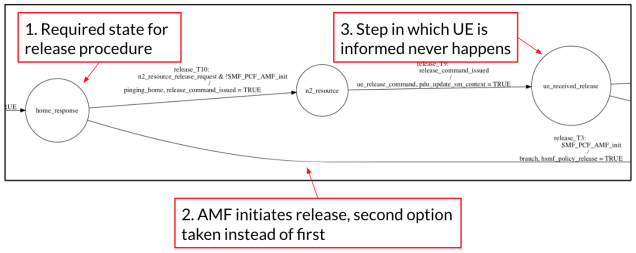

Figure 6: State machine analysis of forced disconnect attack.

insufficient resources to fulfill the request and terminate the connection. The second remaining counterexample corresponds to the third property defined above. This failure occurs after a periodic syncing update between the UE and the network. Normally, the UE should update its status to match that given by the network. However, if the UE indicates that it has no session active in contrast to the network thinking that it does have a session active, the network will immediately terminate the session it thinks it has.

5.6 Potential Attacks

Of the two counterexamples generated above, only the second is feasible as an attack. The reason the first counterexample is infeasible as an attack is because it depends on internal network communications, which are inaccessible or extremely difficult to access for both researchers and hackers. In contrast, the second counterexample simply requires a mismatched syncing message from the perspective of the UE, which is relatively easy to access for attackers. Thus, the new attack presented for the forced disconnect DoS attack is described:

Attacker Requirements: In order to execute this attack, the attacker simply needs to be able to spoof one message from the UE to the serving data network stating that the UE has no PDU sessions active. Specifically how the attacker accomplishes this is arbitrary and not the area of focus for this project, and a multitude of options for identity theft already exist and are known. The following description and consequences show what would happen IF the attacker could gain access to the necessary information.

Attack Description: This attack makes use of status syncing between the UE and its serving data network. In each active PDU session, the UE and network periodically communicate with each other to make sure that both parties are on the same page as far as the status of the PDU session.

If there is a mismatch, specifically, the UE thinks the PDU session has been terminated but the serving network thinks the session is still active, then the network will immediately cut off all resources associated with that UE. After performing this cutoff, the UE is not notified again, as it is assumed that no session exists any longer between the UE and the network. In the case of roaming network, additional steps are taken so that even the resources connecting the home network with the visiting network are terminated.

Attack Consequences: The first and most simple consequence of this attack is denial of service. If the network receives a falsified message that the UE thinks there is no session, the network removes all resources associated with the connection without checking if the message is legitimate. This makes sense from a protocol implementation perspective, as checking if every message was legitimate would greatly increase network resource usage. However, the result is that the network would be disconnected from the UE and the UE would not know it, and would thus try to send packets as normal and not receive a response.

The second consequence is a little more subtle and is a result of the first. 3GPP charging requirements states that it should be possible to charge a user even for failed attempts at data usage. Thus, the failed attempts to receive service from the first consequence could incur an unfair charge on the user.

5.7 Real World Validation/Testing Plan

A real world validation test would require three parties, the same as specified in the attack above. The first two parties are self explanatory, a user and a serving data network that charges based on a PDU session. Whether or not this is a real-world data provider of some testbed infrastructure does not matter. The third party needs to be a second device that has access to the message sending capabilities of the first user device.

The experiment would consist of first setting up a connection as normal (i.e. connecting as normal, using data to

stream, access the internet, etc.). Then, at any point, the attacker should be able to send the spoofed message to the network, which should cause the UE to become disconnected. The UE should not be notified of this, and should instead just hang at the next attempt to use data.

5.8 Future Work

In addition to the physical testbed verification of the attack, more work could be done to expand the theoretical state machine that composed the bulk of the work for this quarter. Firstly, the PDU session modification procedure could be added in the same manner as the establishment and release procedures that are currently modeled. Secondly, some of the omitted details can be added back in. The best detail to add back into the state machine would probably be the sub-protocols that show up in the TS as black boxes but are specified elsewhere (e.g. CHF Selection, see 3GPP TS 23.502).

6 CONCLUSION

This quarter, our team learned how to apply formal methods to the verification of 5G protocols as well as performing research as a whole. All of us succeeded in both implementing the formal methods themselves in addition to showing their viability in exposing protocol vulnerabilities. In doing so, we have exposed 5 potential new vulnerabilities in various areas of the 5G system. Finally, the state machines and specific counterexamples we generated have laid the groundwork for future work that includes improvement of the model and physical verification of the attacks on real testbeds.

APPENDIX: CODE AND RESOURCES

Each of the major protocol categories is in its own GitHub repository due to differences in implementation. Similarly, each category has its own instructions, which can be found in each repository's `README.md`. The links to the public repositories are as follows:

- Radio Resource Control Protocol:
  `https://github.com/alexkgrimes/RRCInspector`
- Emergency Calling Protocols:
  `https://github.com/kaiyuhou/ENL`
  in the `verif` subdirectory
- Roaming Charging Procedure:
  `https://github.com/lyonthezhang/CHFInspector`

REFERENCES

[1] 3GPP. [n. d.]. TS36.304: User Equipment (UE) procedures in idle mode. http://www.3gpp.org/ftp/Specs/archive/36_series/36.304/. 2019.

[2] 3GPP. [n. d.]. TS36.331: Radio Resource Control (RRC); Protocol specification. http://www.3gpp.org/ftp/Specs/archive/36_series/36.331/. 2019.

[3] Lin Huang. [n. d.]. LTE Redirection Attack - Forcing Targeted LTE Cellphone into Unsafe Network. 2016.

[4] Syed Hussain, Omar Chowdhury, Shagufta Mehnaz, and Elisa Bertino. 2018. LTEInspector: A systematic approach for adversarial testing of 4G LTE. In *Network and Distributed Systems Security (NDSS) Symposium 2018*.

[5] Hongil Kim, Jiho Lee, Eunkyu Lee, and Yongdae Kim. 2019. Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane. In *IEEE Symposium on Security & Privacy (SP)*. IEEE, 0.

[6] Guan-Hua Tu, Yuanjie Li, Chunyi Peng, Chi-Yu Li, Hongyi Wang, and Songwu Lu. 2015. Control-plane protocol interactions in cellular networks. *ACM SIGCOMM Computer Communication Review* 44, 4 (2015), 223–234.