A collaborative framework for intrusion detection in mobile networks

Mauro Andreolini[a,*], Michele Colajanni[b], Mirco Marchetti[b]

[a] *Department of Physics, Computer Science and Mathematics, University of Modena and Reggio Emilia, Via Campi 213/a, 41125 Modena, Italy*
[b] *Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, Via Vignolese 905/b, 41125 Modena, Italy*

Abstract

Mobile devices are becoming the most popular way of connection, but protocols supporting mobility represent a serious source of concerns because their initial design did not enforce strong security. This paper introduces a novel class of stealth network attacks, called mobility-based evasion, where an attacker splits a malicious payload in such a way that no part can be recognized by existing defensive mechanisms including the most modern network intrusion detection systems operating in stateful mode. We propose an original cooperative framework for intrusion detection that can prevent mobility-based evasion. The viability and performance of the proposed solution is shown through a prototype applied to Mobile IPv4, Mobile IPv6 and WiFi protocols.

Keywords: Network intrusion detection, NIDS state migration, Mobility-based NIDS evasion, WLAN, Mobile IPv4, Mobile IPv6

*Corresponding author
Email addresses: mauro.andreolini@unimore.it (Mauro Andreolini), michele.colajanni@unimore.it (Michele Colajanni), mirco.marchetti@unimore.it (Mirco Marchetti)

1. Introduction

Society has become dependent on a wide array of mobile devices. For example, most credit-card swipes at restaurants are performed with mobile devices. RFID is widespread in inventory management. To lower infrastructural costs and to appease their employees, companies are seeking to enroll so-called "Bring Your Own Device" (BYOD) policies that allow workers to gain controlled access to the internal network resources through their mobile devices (mainly laptops and phones). These devices tipically run a mix of enterprise and personal applications. Cisco predicts that the number of mobile devices will exceed the world population by 2014.

An inevitable consequence of the huge success of user mobility is an increasing exposure of mobile devices and networks to a wide array of attacks. In addition to eavesdropping on wireless transmissions [5, 15, 23], break-in [33, 35], GSM impersonation [16, 13], social engineering [4], we present a novel form of attacks called *mobile evasion* that can be applied to mobile protocols, such as Mobile IPv4, Mobile IPv6 and WiFi. Mobile evasion leverages the intrinsic vulnerability of mobile protocols supporting *transparent mobility* where roaming events do not interrupt established connections [14]. This is a mandatory feature for all applications requiring a stable connection, but it exposes mobile nodes and related networks to so called "stealth" network attacks. They do not exploit vulnerabilities in protocol implementations, but the effects of mobile node migrations on routing. An attacker can route different portions of a malicious payload through different network paths, thus avoiding the possibility of detection by typical defensive network

mechanisms including anomaly-based [26, 12, 25], signature-based [1, 2, 11], location-based [34] intrusion detection systems (NIDS) [30]. Existing defensive mechanisms are not designed for facing transparent mobility, hence they are inherently incapable of detecting a mobile stealth attack that is fragmented, because no portion of the payload can be matched against the NIDS signature database. Even the most advanced cooperative NIDSs (e.g., centralized [31, 32], hierarchical [24, 17], peer-to-peer [21, 19, 37, 20]) are vulnerable because they cooperate by exchanging data pre-processed by one NIDS.

We initially present a model of the mobile evasion attack that can be applied to any well known mobile protocol. Then, we propose an innovative solution that leverages a novel way for NIDS cooperation. The proposed scheme allows sharing of internal state information among multiple NIDSs deployed in different networks or network segments. The overall solution is integrated into a prototype which extends Snort, but it can be easily adapted to any other NIDS because the implementation is based on a lightweight agent and a set of plugins handling different protocols. This modular design guarantees great flexibility in terms of deployment and expandability. We validate the efficacy and efficiency of the proposed framework for different combinations of migration rates and network protocols. We can conclude that the proposed solution can detect mobility-based attacks in all tested real scenarios at a negligible cost in terms of performance.

The paper is organized as follows. Section 2 compares our solution against related work. Section 3 introduces a general model of the mobile evasion attack and instantiates it for three mobile network protocols: Mobile IPv4,

Mobile IPv6 and 802.11g. Section 4 proposes a novel cooperation scheme that is able to thwart mobile evasion for a wide variety of roaming implementations. Sections 5, 6 and 7 present the relevant details for 802.11g, Mobile IPv4 and Mobile IPv6. Section 8 discusses functional and performance evaluation results obtained through experiments. Section 9 outlines main conclusions.

2. Related work

Mobile ad-hoc networks represent an important source of information about intrusion detection systems for wireless environments. For example, Thamilarasu et al. [28] propose a Cross-layer Intrusion Detection System (IDS) in order to mitigate DoS attacks in ad-hoc networks with a focus on collisions, misdirection and packet drops. The cross-layer design is able to detect intrusion at different protocol layers and to exploit the information from one layer to another layer. Zhang et al. [36] propose a distributed and cooperative IDS architecture where each node participates with its information resources. Other authors [22] address the issues related to a rigid response to an intrusion by proposing a flexible scheme that depends on the measured severity of attack and the degradation in network performance. We remark that all these interesting proposals focus on ad-hoc networks that are quite different from mobile Internet-based networks of interest for this paper. For a similar reason, we distinguish from approaches using host IDS [11], and from statistical profiling algorithms (e.g., [34]) that are ineffective against mobile evasion.

We differ also from proposals using distributed intrusion detection sys-

tems (e.g., [31, 32, 6]) that gather data from different sources and send alerts to one aggregator analyzing and correlating all available information. These solutions are based on different IDS architectures: hierarchical (e.g., Emerald [24]), hierarchical and autonomous for cloud systems [17], peer-to-peer (e.g., [21, 19, 37, 20] where the main goal is to avoid single points of failure). All these systems are oriented to exchange high level information, filter and aggregate it with the goal of reducing the amount of transferred data and to increase the intrinsic value of alerts to human operators. On the other hand, in order to contrast mobile evasion attacks, we have to share in an efficient way raw data at the level of the NIDS internal states with minimal pre-processing work. With respect to this objective, our work is more related to NIDS architectures exploiting state migration. For example, parallel NIDS architectures [8, 27, 7, 3, 6] achieve cooperation by migrating the connection state from one NIDS to another. In [27] the authors synchronize one or more NIDS internal variables that are identified in the configuration with the goal of generating a pool of values shared among all the cooperative NIDS sensors distributed among different network links. The same mechanisms for coordinating lower-level analysis were used to implement a NIDS cluster [29], while a different framework provides methods to export/import complete state information from/into a NIDS [8, 7, 3]. All these solutions use state migration to pursue load balancing or to improve performance, while our proposal describes a novel scheme to prevent an attacker to exploit mobility to evade detection. This is achieved by supporting a mechanism where the state information related to a Mobile Node "follows" the Mobile Node in the new network. We improve on our previous paper [9] in several ways:

we design a modular and general framework that supports different mobile protocols, we implement it by extending the Snort software and we present a large set of experimental results demonstrating functional and performance effectiveness.

3. Mobility-based NIDS evasion

We describe the mobile evasion attack by considering the most advanced stateful NIDS architectures, because stateless systems can be easily bypassed by several types of attacks and are now deprecated. In a stateful NIDS, the information of a network packet, which is relevant to intrusion, is used to create and update an internal state about all the active transport level connections. For each connection, a pre-processor maintains several metadata and two ordered lists of payloads (one for each direction) exchanged by the endpoints. The detection algorithm is then executed on the entire state information. As a consequence, although no individual packet contains enough information to detect an intrusion, a stateful NIDS can detect it by correlating information extracted from different packets. The problems originate when we consider a scenario allowing node mobility where an attacker can pursue *mobility-based NIDS evasion* that was introduced in [10]. Here, an attacker exploits network mobility and evasion techniques based on attack fragmentation in order to avoid detection even by the most advances stateful NIDS systems. This attack can be carried out in three scenarios that we describe in the following subsections. We underline that the following scenarios are independent of the mobile node roaming technology. Mobility-based NIDS evasion can be carried out if these realistic conditions are met:

1. the attack is a malicious payload exploiting a remote vulnerability;
2. it is possible to divide the malicious payload in at least two portions *Portion 1*, *Portion 2* such that nor *Portion 1* neither *Portion 2* are detected by NIDS signatures;
3. the roaming process is transparent; active transport level connections are not interrupted by the handover process.

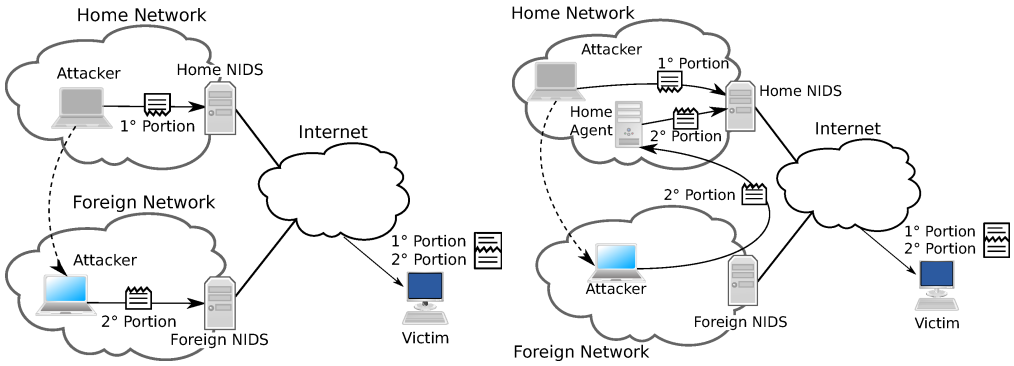
The last assumption is satisfied by several technologies and network protocols, such as Mobile IPv4, Mobile IPv6 and layer-2 protocols for handover across wireless access points and networks.

3.1. Mobile attacker, fixed victim

Figure 1a introduces the first scenario in which two nodes, *Attacker* and *Victim*, can communicate through the Internet. The attacker uses a mobile node which initially operates in a *Home Network* that allows node mobility. To pursue the threat, the attacker will migrate to a *Foreign Network* at some favorable time. Both the home and the foreign networks are monitored by stateful, signature-based NIDSs. Initially, the attacker chooses a remote vulnerability of the victim node and splits the corresponding payload in *Portion 1* and *Portion 2*. Since IP packet fragmentation is discouraged in IPv6 and easily detected by modern NIDSs as anomalous network activity, the attacker packs the two portions inside two not fragmented TCP segments with consecutive sequence numbers. Then, the attacker establishes a TCP connection with the victim and sends Portion 1 from his home network. The home NIDS intercepts and analyzes Portion 1, updates its state information, but it does not have enough information to detect an intrusion. Meanwhile, the attacker roams to the foreign network.

After the migration, packets between the attacker and the victim can be routed through two different schemes: *Direct Communication* or *Tunneling*. Figure 1a refers to direct communication, where the Portion 2 of the payload is routed directly through the foreign network. Portion 2 is intercepted by the foreign NIDS that has not received the previous packet and does not have the necessary state information to recognize the attack. This state information is possessed by the home NIDS, but it is useless since it does not receive Portion 2. As a result, none of the two NIDSs deployed in the home and in the foreign network can detect the attack. We note that this failure is not the result of a bug in NIDS implementations. Node mobility allows the malicious mobile attacker to perform a stealth attack, that is not due to some bug(s) in NIDS implementations. Depending on how node mobility is implemented, only a stateful NIDS installed in the victim node's network may be able to detect the intrusion attempt. In any case, both the home and the foreign network infrastructures can be exploited by a mobile attacker to damage third parties, without any evidence for the security administrator.

Figure 1b refers to the tunneling scheme. The packets sent by the attacker are routed through the home network where a *Home Agent* processes them. This scheme guarantees the mobile attacker to migrate from one network to another while the fixed victim continues to reply to an address within the home network. Depending on the mobility implementation, the attacker might convey its packets through a *Foreign Agent* that is connected with the home agent. In both cases, Portions 1 and 2 of the payload are intercepted by the home network NIDS that can successfully detect the intrusion attempt.

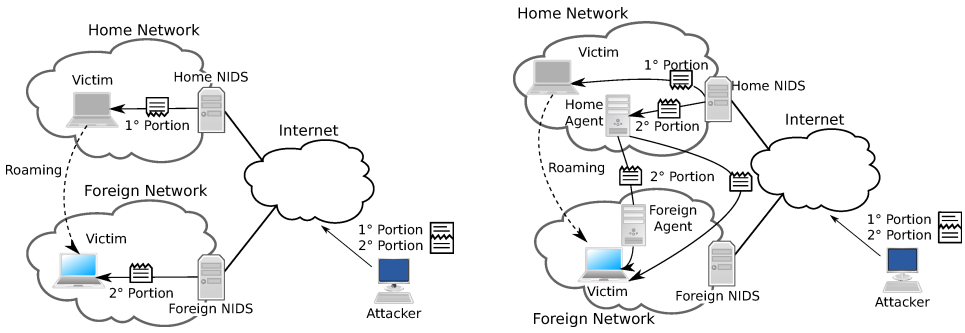


(a) Direct communication (b) Tunneling

3.2. Fixed attacker, mobile victim

In this scenario, the roles are reversed with respect to those in Section 3.1. Here, the mobile node is the victim while the attacker node is fixed. We assume that the home and foreign networks are monitored by two stateful NIDSs, and that the attacker knows when the mobile victim roams across different networks. We show the feasibility for different mobile protocols in Sections 5, 6 and 7. Figure 2a illustrates the attack in the direct communication scheme. The attacker establishes a TCP connection with the victim and sends Portion 1 through its network to the home network. Portion 1 is intercepted and analyzed by the home NIDS, but it does not trigger an alert since it is not recognized as malicious. Then, the attacker waits for the mobile victim to roam to the foreign network. After the migration of the mobile victim, the attacker sends Portion 2 to the foreign network. Here, it is intercepted and analyzed by the foreign network NIDS, but it does not trigger an alert since it is not recognized as malicious. Ultimately, neither the home NIDS nor the foreign network NIDS receive enough information to be able to detect the stealth attack.

Figure 2b shows the attack in the tunneling scheme. Portion 1 is sent directly to the mobile victim, hence it is received and analyzed by the home NIDS. Portion 2 is sent by the attacker after the mobile victim roams to the foreign network, and is routed through a home agent. Depending on the mobility implementation, the home agent can forward packets directly to the mobile victim or through a foreign agent in charge of the foreign network. In this setup we have a partially stealth attack. The home NIDS is able to inspect both portions of the payload, and to detect the attack. However, it cannot sanitize the mobile victim nor protect the nodes within the foreign network from the compromised machine. On the other hand, the foreign NIDS can only analyze Portion 2, thus being unable to detect the intrusion attempt. As a result, the mobile victim has been compromised while connected to the foreign network, the network administrator having no chance to detect the attack.



(a) Direct communication (b) Tunneling

3.3. Mobile attacker, mobile victim

The last scenario is a combination of the previous two cases. Here, both the attacker and the victim are mobile nodes. We assume that all the four

networks involved in this scenario (victim home network, victim foreign network, attacker home network, attacker foreign network) are monitored by stateful NIDSs, and that the attacker knows when the victim roams to the foreign network. The attacker establishes a TCP connection with the victim and sends Portion 1 through its home network to the victim home network. Then, the attacker waits for the victim to roam to the foreign network. Then, the attacker roams to a (possibly different) foreign network and sends Portion 2. It is worth noting that the migration steps can be inverted without changing the attack outcome. Since the attacker roams to a foreign network before sending Portion 2, none of the two NIDSs deployed in the attacker's home and foreign networks receive the complete payload. Hence, they are not able to detect the attack. Moreover, since the victim is also roaming, the detection ability of the stateful NIDSs that monitor the victim's home and foreign networks is reduced. We distinguish between node mobility through tunneling and direct communication.

Tunneling. Portion 2 is forwarded to the victim through a home agent, as shown in Figure 2b. In this case the home NIDS of the victim receives Portion 1 and 2, and it is able to detect the attack. However, the attacker is able to inject the payload into the victim without evidence for the victim home and network NIDSs (partially stealth attack).

Direct communication. Portion 1 and 2 are routed directly to the victim, as shown in Figure 2a. Here both the victim home and foreign NIDS are unable to detect the attack, because they do not receive the complete payload. Hence, none of the four stateful NIDSs can detect the attack, which is completely stealth.

4. Solution through NIDS cooperation

Mobility-based NIDS evasion prevents even modern stateful NIDSs to build a complete state, thus exposing them to the same evasion strategies that were effective only against obsolete stateless NIDS systems. We propose a cooperative solution based on distributed stateful NIDSs exchanging state information. The idea is to extend with three functions the mobility protocols that allow a mobile node to roam: extraction and serialization of state information related to a mobile node from the NIDS deployed in the origin network (*state export*), transmission of the serialized state to the destination network NIDS, deserialization and merging of the transmitted state information within the state of the destination network NIDS (*state import*). The entire process is called *state migration* and allows NIDS state information to "follow" the mobile node in the new network, thus preventing de facto mobility-based evasion.

To implement state migration we introduce the *External Agent*, a modular software deployed in all the networks hosting cooperating NIDSs. For each supported mobile protocol, an external agent detects roaming events and triggers the appropriate state export and import operations among the NIDSs involved in the origin and destination networks. This solution requires only limited changes to the NIDS code base and can be extended to different mobile protocols through new *Plugins*. We discuss three cases, corresponding to possible migrations of the mobile node. In the *First Migration* case, the mobile node roams from the home network to a foreign network. In the *Return to Home* case, the mobile node returns to its home network. In the *Further Migration* case, the mobile node roams from a foreign network to

another foreign network. In the descriptions of these scenarios, we assume that:

1. state migration takes place via a secure channel (e.g., SSL/TLS) to guarantee authentication of the external agents, non-repudiation, message integrity and confidentiality;
2. a trust relationship between the networks among which the mobile node is roaming (otherwise, importing untrusted data into a NIDS could compromise its integrity and thereby network security);
3. the external agent is capable of analyzing the whole network traffic (or at least the portion generated by the mobile nodes) within its network, with the goal of detecting when a mobile node joins the network;
4. the external agent deployed in a foreign network is able to determine the IP address of the external agent deployed in the home network of a mobile node.

Sections 5, 6 and 7 discuss all these requirements for specific mobile protocols.

4.1. First Migration

Figure 3 shows the sequence of messages exchanged during the state migration process. On the left we represent the mobile node migration omitting specific protocol messages. On the right we report each message exchanged among the deployed external agent and the NIDSs monitoring the home and the foreign network. The order in which the hosts are placed left-to-right does not necessarily reflect the real network topology. The *Foreign External Agent* and the *Home External Agent* are the external agents deployed in the foreign and home networks, respectively.

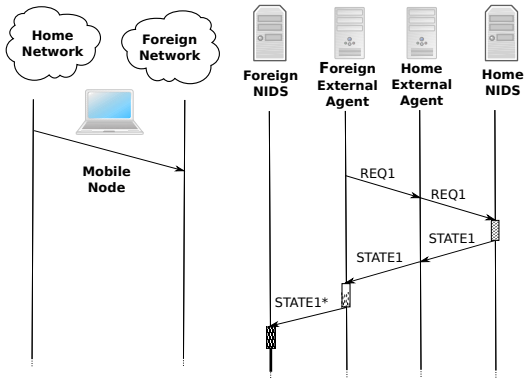


Figure 3: Sequence diagram for the "First Migration" scenario

When the foreign external agent detects the arrival of a new mobile node, it retrieves information about its home network. This operation could be performed in different ways depending on the protocol used by the mobile node for migration (more details on this in Sections 5, 6 and 7). Once the foreign external agent has obtained the address of the home external agent, it sends to it a request asking for the state information related to the mobile node. This request is analyzed by the home external agent, which verifies its consistency (e.g., it checks whether the mobile node has really left the home network and joined the foreign network), then it exports the requested state information stored by the home NIDS and sends the extracted data to the foreign external agent. The latter pre-processes the received information accordingly to the mobile protocol implementation, guarantees the correct normalization of the content and that the format is the same used by the foreign NIDS. Once the foreign NIDS has merged the received data into its internal state, all the communications related to the mobile node are monitored correctly. If an attacker tries to exploit one of the mobility-based

NIDS evasion scenarios described in Section 3, the fragments of the malicious payload are merged into the state information of the foreign NIDS, which now is able to analyze the whole payload and detect the attack. Hence, *state migration* driven by mobile activities prevents mobility-based NIDS evasion.

4.2. Return to Home

Figure 4 shows the sequence of messages exchanged during state migration. Unlike the previous scenario, here all the roles but the mobile node are reversed. When the home external agent detects the return of the mobile node, it asks the foreign external agent for related state information. If the home external agent has previously cached the address of the foreign external agent when the mobile node left the home network, it can now contact it directly; otherwise, it has to infer its address from the captured network traffic. In the latter case, it could sniff messages containing IP address updates (*IP* in Figure 4) and related acknowledgments (*IPA* in Figure 4). When the request reaches the foreign external agent, steps similar to those described in the previous scenario are executed. The foreign external agent replies to the home external agent with the state information exported from the foreign NIDS. Then, the home external agent pre-processes the received data and imports the state information into the home NIDS.

4.3. Further migration

The main difference between this scenario and the first migration phase in Section 4.1 is that the mobile node reaches the destination network from a different foreign network that we call *Origin Network*. Hence the relevant state information is only known by the NIDS deployed in the origin network,

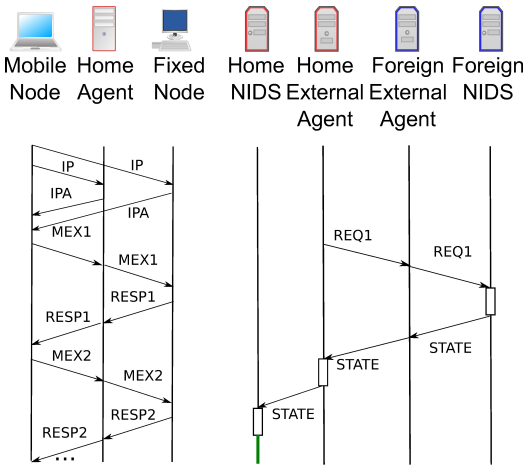


Figure 4: Sequence diagram for the "Return to Home" scenario

the *Origin NIDS*. This can be a problem, because the chance exists that the external agent of the destination network cannot infer the address of the external agent operating in the origin network exclusively from messages sent by the mobile node. Without this address, the external agent of the destination network is unable to issue a request for the relevant state information directly to its corresponding agent in the origin network.

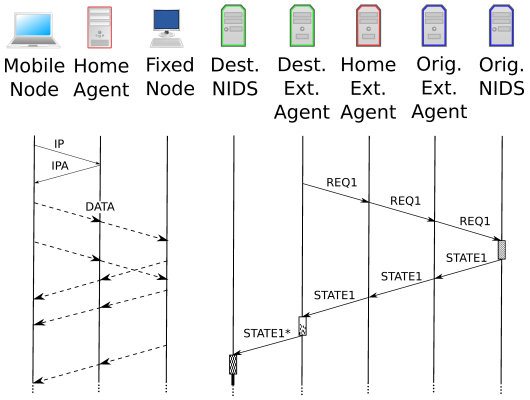


Figure 5: Sequence diagram for the "Further Migration" scenario

To solve this problem, the external agent of the destination network integrates two concurrent approaches. When it detects the arrival of a new mobile node, it analyzes network packets sent by the mobile node (messages *IP* and *IPA* in Figure 5) and tries to extrapolate some useful information to detect the external agent of the origin network. At the same time it looks for packets related to the home network. If the former approach succeeds, the external agent of the destination network directly contacts the corresponding external agent in the origin network and the sequence of steps is the same as in the first migration scenario depicted in Section 4.1.

Otherwise, the external agent of the destination network issues a state export request (*REQ1* in Figure 5) to the external agent of the home network. We recall that this is possible because the external agent of the home network knows the address of the external agent in the foreign network, since the latter has already asked to the former the state information related to the mobile node when it roamed to the origin network. Then, the external agent in the foreign network sends the state export to the origin NIDS that contains the relevant state information. This information (*STATE1* in Figure 5) is finally forwarded through the external agents of the origin and home networks to the external agent of the destination network, where it is pre-processed. The refined state information is then delivered to the destination NIDS (now immune to mobility-based NIDS evasion techniques).

5. WiFi environments

In this section we describe mobility-based NIDS evasion and related countermeasures in a wireless LAN (WLAN) based on WiFi. Here, the main ac-

tors are the nodes equipped with wireless network interfaces and the *Access Points* that allow nodes to connect to the wired network infrastructure. In order to join the network, a node has to successfully complete the authentication and the association phases by exchanging specific packets with the access point. This procedure has to be repeated each time a mobile node decides to migrate to a new access point.

5.1. WiFi evasion

When a mobile node roams from an access point to another one in the same network, the migration is completely transparent to the network layer and above, thus ensuring uninterrupted connections with the other nodes. The mobility-based NIDS evasion technique can be exploited when the access points involved in the roaming process route the network traffic through different paths monitored by different NIDSs. This scenario reflects large organizations where the network is divided into different and independent departments. A unique WLAN for the entire network of the organization (same ESSID and subnetwork) enables mobile nodes to freely move around. In this scenario, each access point routes the network traffic through the gateway of its department, hence the NIDSs deployed in each department are able to monitor only a portion of the network traffic generated within the WLAN. A mobile attacker can roam even without physically moving if he can reach the coverage area of the new access point. All he has to do is force his wireless network interface card to be associated to the new access point. Then he can evade the respective NIDSs by splitting the malicious payload into two portions and sending them through two different access points.

Another option is represented by an attacker targeting a mobile node. To

detect the migration of a mobile node from one access point to another, an attacker can monitor network traffic looking for messages related to roaming (such as packets exchanged during the association to the new access points). Then, the attacker sends Portion 1 while the victim is associated to an access point, and he waits until the association process to the second access point has been completed. Finally, he sends Portion 2.

5.2. WiFi cooperation

To enable the NIDS cooperation scheme proposed in Section 4, the external agents need to detect the migration of a mobile node between access points. In addition, the external agents require information about the home and origin networks in order to request the state information related to previous network activity of the mobile node. The 802.11g standard does not require a specific agent for managing the roaming processes of the mobile nodes. Access points have to maintain a local table containing the information of the associated nodes and they need to know when one of them is not associated anymore. Other components of the network do not know neither through which access point it is possible to reach a wireless node nor in which department the wireless node is operating. This means that it is difficult to trace movements of mobile nodes migrating from network to network. In order to solve this problem several approaches exists, including (but not limited to) the following.

A first approach introduces two extra software components: an agent deployed within each department in charge of reporting the detected wireless nodes; a centralized module which collects information from each agent and can reply to the requests concerning wireless nodes locations. Then,

the external agent is configured to contact the collector in order to obtain information about the origin networks of the mobile nodes.

Alternatively, one can provide to each external agent a list of the nearby external agents. Here the assumption is that mobile nodes can migrate only between adjacent access points. When an external agent detects a new mobile node within the monitored network segment, it broadcasts a state export request to all the neighbors. Then, it receives a reply only from those having state information related to the signaled mobile node.

Finally, one can leverage the characteristics of specific protocols. For instance, in a WiFi network which requires user authentication and provides authentication and authorization services through the RADIUS protocol, it is possible to trace the position of mobile nodes by deploying an extra agent within the RADIUS server. This agent is in charge of registering the origin of the accepted authentication requests, while the external agentss can be forced to contact it in order to obtain the requested addresses.

6. Mobile IPv4 environments

The goal of Mobile IP is to allow a mobile node to communicate with other hosts after changing its point of attachment to the Internet without changing its IP address. Mobile IP allows a mobile node to be always reachable at its *Home Address* also when attached to a network different from the home network. To pursue this goal, the mobile node needs to support Mobile IP. Moreover, the home and the foreign networks have to deploy the mobility home agent and the foreign agent, respectively. No software module is required for the fixed nodes which communicate with the mobile

node. Mobile IP has two communication schemes for mobile nodes which are within a foreign network. *Reverse Tunneling* routes all the packets sent by the mobile node through the origin network, where the home agent is in charge to process them. *Triangular Routing* enables the mobile node to directly send packets to a corresponding node, while receiving replies through the home network. This approach reduces communication latency at least in one direction, without extra requirements for the corresponding nodes.

6.1. Mobile IP evasion

In case of tunneling the mobile evasion attack follows the same scheme described in Section 3. The more interesting scenario is triangular routing, which we discuss next. We refer to the three scenarios described in Section 3.

Mobile victim. From the attacker's point of view, when the mobile node is the victim, the tunneling mode and the triangular routing scheme are similar because both fragments sent to the mobile node pass through the home network. In fact, when triangular routing is enabled, only the packets sent by the victim are directly routed to the attacker, while the packets sent by the attacker pass through the home network, but these differences do not change the attack scheme.

Mobile attacker. When an attacker enables triangular routing, he can directly send Portion 2 to the victim, thus the NIDSs monitoring the attacker are unable to detect the malicious payload, which can be detected by the NIDS deployed in the victim network.

Mobile victim and attacker. If both the attacker and the victim are mobile nodes using triangular routing, the attacker can skip detection by the NIDSs monitoring his network and by the victim's foreign NIDS, but the whole

malicious payload could be detected by the home NIDS of the victim. From the point of view of the attacker, Portion 2 is directly sent to the victim, but it actually passes through the victim home network because the victim receives packets through his home agent.

In summary, when Mobile IPv4 is adopted, a mobile evasion attack is detected at least by the home NIDS of the victim, but not by the foreign NIDS of the victim nor by that of the attacker. We also highlight that both routing schemes of Mobile IPv4 introduce a noticeable communication delay, and an attacker can exploit this latency in order to detect the migration of the victim from the home network to a foreign network.

6.2. Mobile IP cooperation

In a Mobile IP network, when a mobile node joins a foreign network it has to register its new position to the home agent. This task can be accomplished through two procedures through the *Care-of-Address* that is, a temporary IP address that allows a home agent to forward messages to the mobile device. If the mobile node receives its care-of address from the foreign agent, it is mandatory to send registration messages to the home agent through it. Otherwise, if the mobile node uses a co-located care-of address obtained without the intervention of a foreign agent, it has to exchange registration messages directly with the home agent. In both cases, two specific messages are involved: *Registration Request* and *Registration Reply*. The main fields contained in the former are the home address of the mobile node, the home agent's address, the identification field, the care-of address and its lifetime. The request is sent to the home agent which may or may not accept the registration, but in both cases it has to inform the mobile node sending a

registration reply message.

These two registration messages can be exploited by the foreign external agent to detect the arrival of a new mobile node and to extrapolate the necessary information to initiate the state migration process. In particular, the home address, the care-of address and the home agent's address can be retrieved from the registration request message, while the registration reply is analyzed to verify if the registration succeeded or failed. In the former case, the foreign external agent sends the state export request to the home external agent asking for the state information related to the mobile node home address. Then, the received data is pre-processed by the foreign external agent and imported into the foreign NIDS.

7. Mobile IPv6 environments

Mobile IPv6 is an extension for IPv6 which allows nodes to remain reachable while migrating from one network to another through its home address and a new care-of address obtained from each joined networks. The Mobile IPv6 protocol allows two communication modes between mobile node and its correspondent node (be it fixed or mobile). The former, *Bidirectional Tunneling*, is similar to Mobile IPv4 tunneling, where the mobile node and the correspondent node communicate through the home agent, and can be used when the correspondent node does not support Mobile IPv6. The latter, *Route Optimization*, establishes direct communications between a correspondent node and a mobile node also when the latter is in a foreign network. This operation is enabled through the *Care-of-Address*, that is, an IPv6 unicast routable address within the foreign network which is automatically assigned

to the mobile node when it joins the network and can be used to route packets directly to the mobile node.

7.1. Mobile IPv6 evasion

Bidirectional tunneling is similar to the tunneling mode previously described. Also the route optimization scheme is similar to direct communication, but some specific details deserve a discussion. When the mobile node migrates from the home network to the foreign network, it generates a unique care-of address belonging to the foreign network address space, and transmits it to the home agent through a *Binding Update* message. The mobile node also challenges the correspondent node's ability to support IPv6 mobility. If the correspondent node supports Mobile IPv6, then it replies to the binding update message received from the mobile node. All the following network packets between the correspondent node and the mobile node are directly routed between them and do not pass through the home network. On the other hand, if the correspondent node does not support Mobile IPv6, it keeps sending packets to the home address. They are received by the home agent and tunneled to the mobile node in the foreign network. From the attacker's point of view, the messages exchanged to enable route optimization (e.g., binding updates) are useful to detect the migration of the mobile node, and they represent a more accurate solution compared to the time based detection in Mobile IPv4. Moreover, the time interval between the phase of joining the foreign network by the mobile node and the establishment of the route optimization can be used to plan a more complex attack. Indeed, an attacker could split the malicious payload in three different portions and then send the first fragment before the migration of the mobile node, the second after

the migration, but before the route optimization and the final fragment after route optimization has taken place. The three packets follow three different paths which could require extra work for the deployed NIDS.

7.2. Mobile IPv6 cooperation

7.2.1. Migration detection in IPv6

The binding update of the home agent is simple: the mobile node sends a special packet containing its home address and its care-of address to the home agent. Then, the home agent replies with a *Binding Acknowledge*, confirming the receipt of this information. While this last acknowledgement is optional, the former is necessary for the mobile node to maintain the active connections alive. Thus, the foreign external agent can start the state migration as soon as it detects the binding acknowledge addressed to the home agent. Furthermore, from the same packet it can extract the IP address of the home agent which, according to our last assumptions in Section 4, corresponds to that of the home external agent. Then, the foreign external agent can send a request to the corresponding home external agent specifying the mobile node's home address in order to obtain the related state information. Once the foreign external agent has received the reply, it pre-processes the state information and finally imports the data into the foreign NIDS.

7.2.2. Route optimization

IPv6 provides a route optimization scheme which enables direct communication between the mobile node and the correspondent node. However, route optimization can be activated only if the correspondent node supports it and only after the correspondent node receives the binding update. Until

that moment, all the mobile node's active connections to and from any correspondent node are tunneled through the home agent using an IPv6-in-IPv6 encapsulation. Thus, the foreign NIDS sees some already active connections between the mobile node and its home agent and some new connection beginning between the mobile node and the correspondent node. However, if the NIDS does not explicitly support MIPv6, it does not merge together the state information of those connections even if they are the same connections (at least at the transport level). For this reason a new scenario is required to manage IPv6-based mobile migrations. Figure 6 shows the messages exchanged during state migration after route optimization.

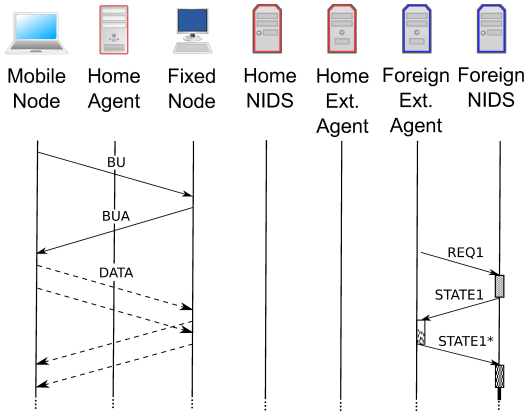


Figure 6: Route optimization sequence diagram

When the foreign external agent detects the binding update and binding update acknowledgement messages exchanged between the mobile node and the correspondent node (be it fixed or mobile; in Figure 6 it is depicted as a fixed one), it starts the state migration process. The sniffed messages contain several pieces of information, in particular the home address of the mobile node, the address of the correspondent node and the address of the home

agent. Then the foreign external agent is able to issue an export request (REQ1 in Figure 6) to the foreign NIDS, asking for all state information related to the home agent. This state information (STATE1 in Figure 6) is pre-processed by the foreign external agent in order to extract only the state information related to the network packets exchanged between the mobile node and the correspondent node that have just performed route optimization. This selection allows the management of connections established with different correspondent nodes, which may or may not support route optimization and which in the former case switch to direct communications in different time intervals. Finally, the pre-processed state information (STATE1* in Figure 6) is sent to the foreign NIDS in order to be imported.

8. Performance evaluation

To show the viability and the effectiveness of the proposed solution to mobile evasion, we design and implement a prototype which provides state migration support among multiple NIDS sensors. This framework consists of three main components.

- The implementation of the *state.import* and *state.export* functions that extract and insert state information related to a specific IP address, respectively. This patch is integrated in the source code of *Snort* that is a popular open source NIDS.

- The external agent that invokes the *state.import* and *state.export* functions. It is implemented as a software module that interacts with the local Snort and other external agents working on cooperating Snort modules.

- A set of plugins that manage the specific details of each mobile protocol. In this version, we have three plugins for WiFi, Mobile IPv4 and Mobile IPv6 protocols. There are no conceptual limits to integrate future protocols or different versions of existing mobile protocols.

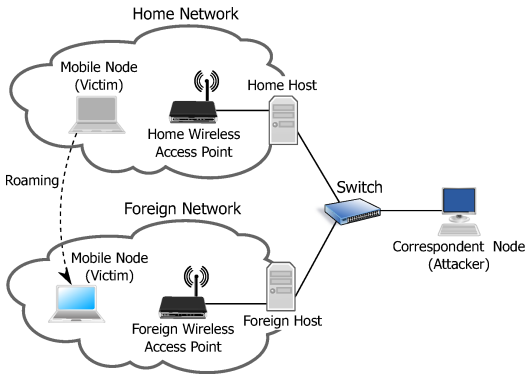


Figure 7: Experimental testbed

Figure 7 shows the considered testbed, where the two clouds on the left denote a home network and the foreign network, which provide mobility support based on Wi-Fi, MIPv4 or MIPv6. Each network is monitored by the *Home Host* and *Foreign Host* which, for the sake of simplicity, execute both our modified Snort and an instance of our external agent. Hence, the home host acts as home NIDS and home external agent, while the foreign host acts as foreign NIDS and foreign external agent. These machines are connected to a Cisco Aironet 1100 access point providing wireless connectivity to mobile nodes. The mobile node is a laptop with a wireless network interface, while the correspondent node is a fixed host. All machines run GNU/Linux with kernels 2.6.35 or 2.6.39, recompiled with all the needed modules to support mobility. Node mobility through Mobile IPv4 is guaranteed by the home

host and foreign hosts running the *Dynamics Mobile IP* daemon. In order to support Mobile IPv6 and route optimization, the home host, the mobile node and the correspondent node run the *mip6d* daemon, and the home host and foreign host run the *radvd* daemon. We replay all the described attack scenarios and verify the possibility of evading the NIDSs as reported in Section 3.

To evaluate the prototype performance we select one environment, Mobile IPv6, and analyze two main features: timings of the state migration procedure and impact of migrations on NIDS performance. We measure the time required by state migration and the execution time of `state.export` and `state.import`. We replay the mobility-based evasion several times under various network conditions. In particular, we generate synthetic network traces characterized by a different number of simultaneously active TCP connections. This is a factor with a direct impact on the resource consumption of the two NIDSs because each NIDS has to create and maintain a dedicated session for each monitored TCP connection. The measures related to these tests are reported in Figure 8a, where the X-axis is the number of concurrent TCP connections in the network traces monitored by Snort, and the Y-axis reports the time in ms. The triangles and circles denote the duration of state export and state import operations under different numbers of concurrent connections, respectively. Each point in the graph represents the average computed over five different experiments. Figure 8a shows that the time required to import the state of the mobile node in the Foreign NIDS is independent of the number of concurrent connections and is lower than 14 ms. On the other hand, the time required to export the state of the mo-

bile node increases linearly with the number of active TCP connections until Snort reaches the limit of concurrent connections tracked by the `Stream5` pre-processor, which is 8192 by default. After that, the state export time remains constant. In our tests the time required to export the state information is less than 32 ms, even in the worst case.

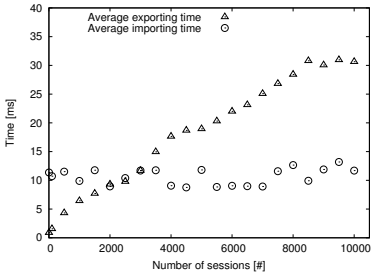
We also measure the time required to complete the state migration process, defined as the time elapsed between the `state.export` request sent to the home external agent by the foreign external agent, and the receipt of the result of the `state.import` operation from the foreign NIDS. The related experimental results are summarized in Table 1. The average state migration time (409 ms) is dominated by network delays. This value is one order of magnitude lower than the time required by the mobile node to complete the roaming from the home to the foreign network (on average $8.835s$ in our experimental testbed). These results and the ability of our prototype to handle out-of-order network packets[1] show that it is compatible with real time analysis of live network traffic.

To estimate the impact of state migration on Snort performance, we run several experiments for evaluating the maximum NIDS bandwidth for a given packet loss rate. The choice of the traffic trace is not simple for a fair NIDS evaluation because there is no standard consensus, there is no benchmark and the widely adopted IDEVAL set [18] is now deprecated because it refers to a scenario that is not at all representative of modern traffic. In our experiments we use a trace of the Capture-the-Flag Hacking Contest held in 2010, which

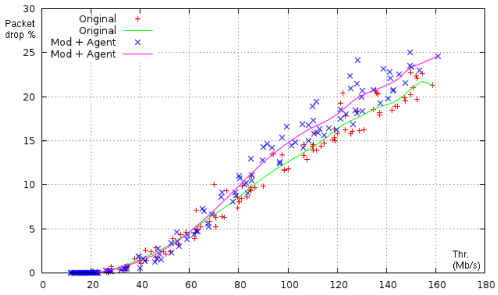
[1]The ability to handle out-of-order network packets is inherited from Snort and improves tolerance to network delays.

Table 1: Times required by state migration activities

	Average [ms]	σ [ms]	Peak [ms]
State import	12	1	13
State export (worst case)	30	1	31
Complete state migration	409	176	765
Network roaming	8835	3495	13209



(a) State importing and exporting times



(b) Snort: original vs. patched

is reasonably recent, publicly available and contains several attacks.

In the first experiment we compare the performance of the original Snort against our framework, which includes a modified version of Snort and the external agent that we run on the same machines. We use the default Snort configuration, and we replay the registered network traces at different speeds ranging from few Mb/s up to 160Mb/s using TCPreplay. We measure the average bit rate generated by TCPreplay and the number of dropped packets registered by Snort. The results are reported in Figure 8b. On our hardware, in both the scenarios Snort is able to process 100% of packets up to 20-25 Mb/s, then it starts dropping packets. The behavior of the original Snort and the patched Snort is the same up to 70-80 Mb/s with a 10% margin.

By increasing the bit rate of the network traces up to 160Mb/s, the patched Snort performs slightly worse at a 25% of dropped packets against 22% of the original Snort. However, both values are too high to be acceptable in real contexts. This means that in the bandwidth range manageable by Snort, our framework offers performance similar to the unmodified version. It is worth remarking that our results provide a lower bound to the real performance of the prototype, because we run both Snort and the EA on the same host.

The goal of the second set of experiments is to determine the impact of state migration on framework performance. In addition to the traffic traces used in the previous experiment, we prepare several traces simulating the migration of different mobile nodes. These traces contain some TCP segments exchanged between the mobile node and the correspondent node, the binding update packets used by Mobile IPv6, and the *state.export* requests sent by the foreign external agent to the home external agent. We focus on the *state.export* mechanism due to its heavier impact on Snort than *state.import*.

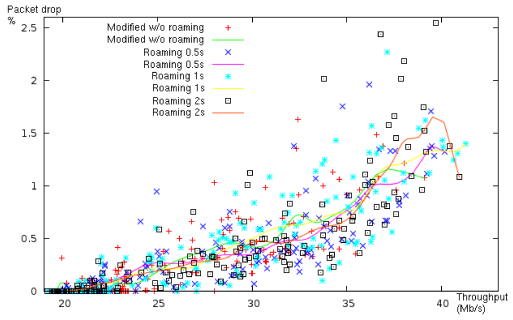


Figure 9: Patched Snort performance with and without roaming for both traces

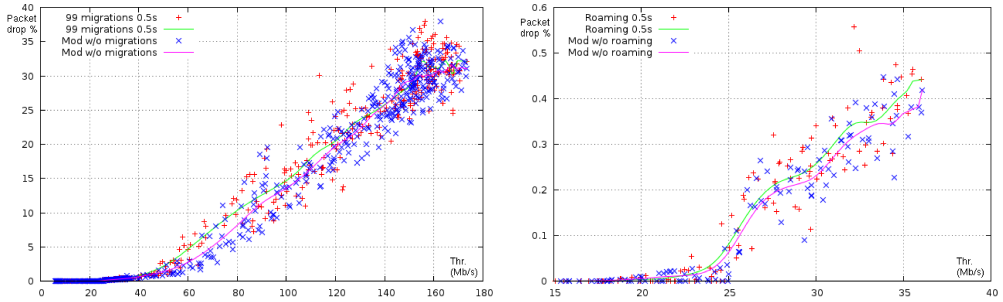
We replay the registered network traffic at different speeds ranging from about 20Mb/s to 40Mb/s, a small range around which Snort starts to drop

packets. Then, we replay the synthetic network traces by simulating mobile nodes roaming at regular intervals: 0.5s, 1s and 2s. We measure the number of packets dropped by Snort and the actual average bit rate generated by *Tcpreplay*. The results are plotted in Figure 9, where we can appreciate that there is no clear difference in performance for different roaming intervals. This reflects the low impact of the state migration on Snort performance. In particular, the *state.export* operation keeps Snort busy for 10-30 ms, that is a limited time period during which the received packets fill the buffer. As soon the operation ends, Snort processes events and it frees the buffer prior to the next roaming.

The last set of experiments aims to verify whether the impact of the state migration process could become more distinguishable at higher bit rates. We compare the performance of our framework in the case of two migrations per second and in the case of no migration at all, and consider a range from about 10 Mb/s up to 170 Mb/s. The results are shown in Figure 10a and in Figure 10b that is a zoom of the range 15-35 Mb/s. While the average results obtained in the two scenarios are really close to each other, the impact of the state migration process is evidenced by the dispersion of measures than is larger than that related to the absence of state migration. These results confirm the effectiveness and the validity of the proposed solutions and prototype.

9. Conclusions

We describe a new attack, called mobility-based evasion, that can be used to perform stealth network intrusions and that is undetectable by state-of-



(a) Patched Snort: w/o roaming vs. roaming every 0.5 seconds

(b) Patched Snort: zoom in the 15-40Mbps range

the-art NIDS architectures. This attack is not due to design or implementation flaws, but it is a strategy combining fragmentation of a malicious payload and node mobility. We also propose an original solution to mobility-based evasion and implement it in a prototype as an extension of the popular Snort software. Several experimental results confirm the efficacy and the efficiency of the proposed solution for several protocols offering network mobility.

[1] Albin, E., Rowe, N., March 2012. A realistic experimental comparison of the Suricata and Snort intrusion-detection systems. In: Enokido, T. (Ed.), Proc. 26th Int. Conf. Advanced Information Networking and Applications, WAINA'12. IEEE, Los Alamitos, CA, pp. 122–127.

[2] Alpcan, T., Bauckhage, C., Schmidt, A. D., 2010. A probabilistic diffusion scheme for anomaly detection on smartphones. In: Samarati, P., Tunstall, M., Posegga, J., Markantonakis, K., Sauveron, D. (Eds.), Information Security Theory and Practices. Security and Privacy of Pervasive Systems and Smart Devices. Springer, Berlin, DE, pp. 31–46.

[3] Andreolini, M., Casolari, S., Colajanni, M., Marchetti, M., July 2007.

Dynamic load balancing for network intrusion detection systems based on distributed architectures. In: Wolf, M., Quaglia, F., Avresky, D. (Eds.), Proc. 6th Int. Symp. Network Computing and Applications, NCA'07. IEEE, Los Alamitos, CA, pp. 153–160.

[4] Becher, M., Freiling, F., Hoffmann, J., Holz, T., Uellenbeck, S., Wolf, C., May 2011. Mobile security catching up? Revealing the nuts and bolts of the security of mobile devices. In: Frincke, D. (Ed.), Proc. Int. Symp. Security and Privacy, SP'11. IEEE, Los Alamitos, CA, pp. 96–111.

[5] Butun, I., Morgera, S. D., Sankar, R., 2014. A survey of intrusion detection systems in wireless sensor networks. Communications Surveys & Tutorials 16 (1), 266–282.

[6] Carli, L. D., Sommer, R., Jha, S., November 2014. Beyond pattern matching: a concurrency model for stateful deep packet inspection. In: Proc. 21st Conf. Computer and Communications Security, SIGSAC'14. ACM, New York City, NY, pp. 1378–1390.

[7] Colajanni, M., Gozzi, D., Marchetti, M., December 2007. Enhancing interoperability and stateful analysis of cooperative network intrusion detection systems. In: Yavatkar, R., Grunwald, D., Ramakrishnan, K. (Eds.), Proc. 3rd Int. Symp. Architectures for Networking and Communication Systems, ANCS'07. ACM, New York City, NY, pp. 165–174.

[8] Colajanni, M., Marchetti, M., September 2006. A parallel architecture for stateful intrusion detection in high traffic networks. In: Carle, G.

(Ed.), Proc. 1st Workshop on Monitoring, Attack Detection and Mitigation, MonAM'06. IEEE, Los Alamitos, CA, pp. 9–16.

[9] Colajanni, M., Zotto, L. D., Marchetti, M., Messori, M., June 2011. Defeating NIDS evasion in mobile ipv6 networks. In: Bononi, L., Banchs, A. (Eds.), Proc. 1st Int. Symp. World of Wireless Mobile and Multimedia Networks, WoWMoM'11. IEEE, Los Alamitos, CA, pp. 1–9.

[10] Colajanni, M., Zotto, L. D., Marchetti, M., Messori, M., February 2011. The problem of NIDS evasion in mobile networks. In: Ghazawi, T. E., Fratta, L. (Eds.), Proc. 4th Int. Conf. New Technologies, Mobility and Security, NTMS'11. IEEE, Los Alamitos, CA, pp. 1–6.

[11] Curti, M., Merlo, A., Migliardi, M., Schiappacasse, S., July 2013. Towards energy-aware intrusion detection systems on mobile devices. In: Proc. 1st Int. Conf. High Performance Computing and Simulation, HPCS'13. IEEE, Los Alamitos, CA, pp. 289–296.

[12] Garcia-Teodoro, P., Diaz-Verdejo, J. E., Macia-Fernandez, G., Vazquez, E., 2009. Anomaly-based network intrusion detection: techniques, systems and challenges. Computers & Security 28 (1), 18–28.

[13] Gobbo, N., Palmieri, F., Castiglione, A., Migliardi, M., Merlo, A., 2014. A denial of service attack to UMTS networks using SIM-less devices. IEEE Trans. Dependable and Secure Computing 11 (3), 280–291.

[14] Golmie, N., 2009. Seamless mobility: are we there yet? IEEE Wireless Communications 16 (4), 12–13.

[15] Ho, Y. L., Heng, S.-H., December 2009. Mobile and ubiquitous malware. In: Proc. 7th Int. Conf. Advances in Mobile Computing and Multimedia, MoMM'09. ACM, New York City, NY, pp. 559–563.

[16] Khan, M., Ahmed, A., Cheema, A. R., August 2008. Vulnerabilities of UMTS access domain security architecture. In: Proc. 9th Int. Conf. Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD'08. IEEE, Los Alamitos, CA, pp. 350–355.

[17] Kholidy, A. H., Abdelkarim, E., Abdelwahed, S., Baiardi, F., June 2013. Ha-cids: A hierarchical and autonomous IDS for cloud systems. In: Proc. 5th Int. Conf. Computational Intelligence, Communication Systems and Networks, CICSyN'13. IEEE, Los Alamitos, CA, pp. 179–184.

[18] Lippmann, R., Fried, D., Graf, I., Haines, J., Kendall, K., McClung, D., Weber, D., Webster, S., Wyschogrod, D., Cunningham, R., Zissman, M., January 2000. Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. In: Proc. Conf. Exp. DARPA Information Survivability, DISCEX'00. IEEE, Los Alamitos, CA, pp. 12–26.

[19] Locasto, M., Parekh, J., Keromytis, A., Stolfo, S., June 2005. Towards collaborative security and p2p intrusion detection. In: Proc. 6th Ann. Workshop Information Assurance, IAW'05. IEEE, Los Alamitos, CA, pp. 333–339.

[20] Loiola, C. D., Wagner, E., Lopes, D., Abdelouahab, Z., Froz, B., 2013.

Network intrusion detection system based on SOA (NIDS-SOA): enhancing interoperability between IDS. In: Elleithy, K., Sobh, T. (Eds.), Innovations and Advances in Computer, Information, Systems Sciences, and Engineering. Springer, Berlin, DE, pp. 935–948.

[21] Marchetti, M., Messori, M., Colajanni, M., 2009. Peer-to-Peer architecture for collaborative intrusion and malware detection at a large scale. Lecture Notes in Computer Science 5735 (1), 475–490.

[22] Nadeem, A., Howarth, M., 2014. An intrusion detection & adaptive response mechanism for MANETs. Ad Hoc Networks 13 (B), 368–380.

[23] Polla, M. L., Martinelli, F., Sgandurra, D., 2013. A survey on security for mobile devices. IEEE Communications Surveys & Tutorials 15 (1), 446–471.

[24] Porras, P., Neumann, P., October 1997. EMERALD: Event monitoring enabling responses to anomalous live disturbances. In: Proc. 20th Conf. Information Systems Security, ISS'97. National Institute of Standards and Technology, Gaithersburg, MA, pp. 353–365.

[25] Qi, Y., Cao, M., Zhang, C., Wu, R., 2014. A design of network behavior-based malware detection system for Android. In: Xiang, Y., Cuzzocrea, A., Hobbs, M., Zhou, W. (Eds.), Algorithms and Architectures for Parallel Processing. Springer, Berlin, DE, pp. 590–600.

[26] Shuaifu, D., Yaxin, L., Tielei, W., Tao, W., Wei, Z., September 2010. Behavior-based malware detection on mobile phone. In: Proc. 6th Int.

Conf. Wireless Communications Networking and Mobile Computing, WiCOM'10. IEEE, Los Alamitos, CA, pp. 1–4.

[27] Sommer, R., Paxson, V., December 2005. Exploiting independent state for network intrusion detection. In: Proc. 21st Ann. Conf. Computer Security Applications, CSA'21. IEEE, Los Alamitos, CA, pp. 13–20.

[28] Thamilarasu, G., Balasubramanian, A., Mishra, S., Sridhar, R., November 2005. A cross-layer based intrusion detection approach for wireless ad hoc networks. In: Proc. 2nd Int. Conf. Mobile Adhoc and Sensor Systems, MASS'05. IEEE, Los Alamitos, CA, pp. 855–861.

[29] Vallentin, M., Sommer, R., Lee, J., Leres, C., Paxson, V., Tierney, B., 2007. The NIDS Cluster: scalable, stateful network intrusion detection on commodity hardware. Lecture Notes in Computer Science 4637 (1), 107–126.

[30] Vasilomanolakis, E., Karuppayah, S., Fischer, M., Muehlhauser, M., Plasoianu, M., Pandikow, L., Pfeiffer, W., November 2013. This network is infected: HosTaGe-a low-interaction honeypot for mobile devices. In: Proc. 3rd Workshop Security and Privacy in Smartphones and Mobile Devices, SPSM'13. ACM, New York City, NY, pp. 43–48.

[31] Vigna, G., Kemmerer, R., December 1998. Netstat: A model-based real-time network intrusion detection system. In: Proc. 14th Ann. Conf. Computer Security Applications, CSA'98. IEEE, Los Alamitos, CA, pp. 25–34.

[32] Wu, Y.-S., Foo, B., Mei, Y., Bagchi, S., December 2003. Collaborative intrusion detection system (CIDS): a framework for accurate and efficient IDS. In: Proc. 19th Ann. Conf. Computer Security Applications, CSA'19. IEEE, Los Alamitos, CA, pp. 234–244.

[33] Xie, L., Zhang, X., Seifert, J.-P., Zhu, S., March 2010. pBMDS: a behavior-based malware detection system for cellphone devices. In: Proc. 3rd Conf. Wireless Network Security, WiSec'10. ACM, New York City, NY, pp. 37–48.

[34] Yazji, S., Scheuermann, P., Dick, R., Trajcevski, G., Jin, R., 2014. Efficient location aware intrusion detection to protect mobile devices. Personal and Ubiquitous Computing 18 (1), 143–162.

[35] Zhang, M., Raghunathan, A., Jha, N., 2014. A defense framework against malware and vulnerability exploits. Int. Journal of Information Security 13 (5), 439–452.

[36] Zhang, Y., Lee, W., AnHuang, Y., 2003. Intrusion detection techniques for mobile wireless networks. Wireless Networks 9 (5), 545–556.

[37] Zhou, C. V., Karunasekera, S., Leckie, C., November 2005. A peer-to-peer collaborative intrusion detection system. In: Proc. 13th Int. Conf. Networks, ICON'05S. IEEE, Los Alamitos, CA, pp. 6–14.