Communication Papers of the Federated Conference on
Computer Science and Information Systems pp. 81–88

DOI: 10.15439/2019F329
ISSN 2300-5963 ACSIS, Vol. 20

Design of a Distributed HIDS for IoT Backbone Components

Guilherme de O. Kfouri[*], Daniel G. V. Gonçalves[†], Bruno V. Dutra[‡], João F. de Alencastro[§],
Francisco L. de Caldas Filho[¶], Lucas M. C. e Martins[‖], Bruno J. G. Praciano[**],
Robson de O. Albuquerque[††]and Rafael T. de Sousa Jr.[‡‡]
National Science and Technology Institute on Cyber Security, Electrical Engineering Department,
University of Brasília (UnB), P.O. Box 4466, Brasília–DF, Brazil, CEP 70910-900
Email: kfouri@ieee.org[*], { daniel.goncalves[†], bruno.dutra[‡], joao.alencastro[§], francisco.lopes[¶],
lucas.martins[‖], bruno.praciano[**], robson[††], rafael.desousa[‡‡]} @redes.unb.br

***Abstract*—Nowadays DDoS attacks using devices from IoT networks are frequent and extensive. Given that IoT network instances are distributed and deployed on the conventional Internet structure, DDoS countermeasures in IoT need to be fully distributed and coordinated all over the components that form each IoT instance. This paper presents a host-based intrusion detection system (HIDS) that was designed and prototyped to protect the components of IoT network backbones comprising conventional switches and routers, not IoT devices. In our design, a set of the proposed HIDS executes conventional security verification, like default username and password, known attacks signatures, usage of resources, processes, ports and open connections, while also interacting with a Controller of the HIDS set to allow the coordination of intrusion detection actions relative to DDoS attacks distributed all over the IoT instance. The designed distributed HIDS is evaluated in a controlled environment that, although being a local and isolated network, realistically represents IoT network instances.**

***Keywords*—Internet of Things (IoT), IoT Security, Distributed Denial of Service (DDoS), Host-based Intrusion Detection System (HIDS), Distributed HIDS, Mirai botnet.**

I. Introduction

Nowadays a significant increase has been observed in both the total number and duration of distributed denial-of-service (DDoS) attacks [1], one of the most known cases being the Mirai botnet [2], which on October 21, 2016, performed a massive attack that left much of the US East Coast without internet access. This attack broke the record of generated traffic in that year reaching 1.2 Tbps using a worldwide network of security cameras and leaving websites such as Netflix, Twitter, Amazon and PayPal unavailable. This record was latter surpassed by a 1.35 Tbps attack against GitHub [3].

This kind of malware is particularly noted for taking advantage of IoT devices which are plagued by basic security vulnerabilities, such as default user identifiers and passwords. This allows the malware to establish access and then to be installed, thus turning the device into a bot victim or zombie, i.e., a malware-infected device that is somehow controlled by an attacker [4]. Upon reaching the desired amount of bots, the botnet controller commands the attack that is performed from potentially all compromised bots. Then, the DDoS attack consists of bots sending requests and traffic to a particular server to exhaust its computational resources and to exhaust the limit of connections supported by the server, disabling it to function properly.

These attacks, including the Mirai botnet and its variants, as well as other different botnets, call for greater security in IoT devices and IoT network components, given the risk of exposing the Internet infrastructure to increasingly larger DDoS attacks [4]. Popular cybersecurity reports, such as [5], discuss the reasons for such security events highlighting that organizations and users are implementing low-cost IoT devices as quickly as possible with little or no security concern. There is also the possibility of new vulnerabilities being discovered even if the known ones are repaired. In this situation, permanent and evolving security measures must be integrated into the components of IoT network instances.

Given the need for this type of security measure for IoT networks, and considering that current IoT network infrastructures mostly use conventional routers and switches, this paper proposes a distributed host-based intrusion detection system (HIDS) for IoT backbone components, including conventional switches and routers. IDS are security tools that are becoming increasingly necessary as firewalls and other security measures are not sufficient to guarantee the integrity of the network [6]. In our design, a set of local HIDS is distributed into the IoT backbone and these local HIDS interact with a set Controller that coordinates the distributed HIDS.

The proposed design also considers that a basic action to identify security vulnerabilities is the proactive monitoring of network elements. Several protocols can be used for this purpose, such as the traditional monitoring via the Simple Network Management Protocol (SNMP) that allows a network manager system to perform periodic requests from local Agents to network devices. This protocol is leveraged in this paper proposal. Another protocol used to support the detection of security vulnerabilities is Syslog. It is used to convey event notification messages [7] that usually come from logs generated by the operational systems, firewalls or other network elements and contain information about events in each host. The logs can be accessed locally or remotely or can be exported via Syslog to a centralized station where they can be analyzed and stored for longer periods.

Our design uses proactive monitoring allied to intrusion detection to allow the identification of malicious attacks and reduce significantly the possibility of invasions. The proposed HIDS is designed to be integrated with the IoT middleware and its supporting components as a fully distributed security service, given the need to have coordinated protection covering the whole set of components present in each IoT instance, as discussed in [8].

Besides this introduction, this paper is organized as follows: Section II presents a brief review of literature about IDS and Section III presents related works on security issues in IoT backbones. Section IV is devoted to the proposed Distributed HIDS for IoT Backbone Components and Section V describes respective validation tests and their results. Finally, Section VI presents general conclusions and suggestions for future work.

II. Related Works

As this paper focuses on security issues in IoT network backbones, this section summarizes literature views of some concepts that are useful for understating the paper content.

A. IDS

Authors in [9] define intrusion detection systems as a combination of software and/or hardware components that monitors computer systems and raises an alarm when an intrusion happens. Other references, like [10], include in their definition the detection of policy violation and the logging of events.

Intrusion detection systems are a very common tool in the defense against multiple kinds of threats, being widely utilized because of their benefits of flexibility, effectiveness, and interoperability, although these systems cannot be used as the sole and complete solution for security problems, as discussed in [6].

Existing IDS proposals use different structures and processes, regarding the location of the IDS, detection methods, responses, timing, architecture and criteria for monitoring targets. Still referring to [6], a Network-based IDS (NIDS) is the most common form of intrusion detection system whose basic idea is to monitor and analyze packets mirrored from a switch, firewall or any other network active device, and the alert about possible malicious activity. Alternatively, a Host-based IDS (HIDS) analyzes data internal to a computer system, such as audit trails, system logs, and critical system files.

B. HIDS

According to [11], an HIDS monitors and collects the state of hosts or server systems that are running in public services containing sensitive information, and the related suspicious activities. Reference [6] shows that HIDSs have a high precision rate on determining users and processes which are involved in an attack. But, paper [10] points out a potential disadvantage, since HIDS rules are predefined based on the current operational systems architecture and its behavior, thus future upgrades and drastic changes could cause problems.

C. Remote IDS

A remote IDS works similarly to a normal IDS, apart from the information being transmitted over the network to a resourceful server. Paper [12] proposes a system built using distributed intelligent data mining agents. Those agents would exchange the gathered data, logs and activity with each other and detect malicious activity at different levels.

III. Security Issues in IoT Backbones

Also in this section discusses other works about security systems for network infrastructure components, e.g., switches and routers, and also points out the difference between their approach and our proposal.

The security of network gear has been an interesting research and development issue for academy and industry given the permanent concurrency regarding ever evolving security requirements for network devices. As a result, for instance, devices may come with the capability of port security [13], which is a router software configuration that enables the filtering of packets via a white list of previously configured middle access control (MAC) addresses that are allowed to communicate with a given interface. In this case, when a packet from a not-allowed MAC source arrives in an interface with port security configured, the router software will apply a preconfigured action like port shutdown. Port security, however, is not the only security configuration for network devices, Direct Host Configuration Protocol (DHCP) snooping [14] is another method for implementing security in the network infrastructure that prevents a fake DHCP server from distributing IPs on a target network. This type of protection is of great importance in the network because the false DHCP server can distribute IPs inside a valid network range which will eventually generate duplicity of addresses, and thus unavailability of services. These solutions are efficient ways to protect the network devices against a small set of attacks. However, for other attack types, such security settings are useless, and there are still operating devices that do not have such security capabilities. Our solution differs from the discussed ones since its purpose is to complement these security policies by employing the analysis of collected metrics from the network devices operating system, regarding known security flaws for specific devices, malformed passwords, leaked passwords, SNMP communities etc, offering the functionalities that are discussed in Section IV.

Some authors have addressed secure network device leveraging an IDS in different ways. For instance, in [15], the authors propose an IDS called JiNao that analyzes OSPF traffic arriving in a network router to detect deviations from the protocol expected behavior. This detection system is divided into four main entities:

1) Rule based prevention: Module that implements a set of rules/policies used to filter packets related to a previously known violation. The application of these rules serves to avoid attackers to cause low response time from the IDS due to unnecessary processing;

2) Detection module: this module is responsible for inspecting OSPF packets to find deviations in the protocol behavior, which is performed according to two main methods: protocol and statistical analysis. The protocol analysis is implemented with an OSPF state machine that describes the normal OSPF operation and allows to detect whether an OSPF entity goes to a state that doesn't match the expected one. In this case, an alert is generated. The statistical analysis is used to define a behavioral pattern for the OSPF protocol so that, if a packet deviates from the pattern, an attack is identified. These methods are complementary because they address different situations;
3) Decision module: this module is the responsible for decision making in the JiNao IDS, a process that takes the output from protocol and statistical analysis to determine if an attack is happening and then, if misbehavior is identified, an alert is generated destined to responsible entities;
4) JiNao MIB: the IDS maintains an SNMP Management Information Base (MIB) to provide network management services with the information about the security analysis regarding the OSPF protocol. This MIB is used to feed information generated by JiNao into a network management system like Zabbix.

This organization of JiNao provides a security system for the network infrastructure that acts directly in the OSPF routing protocol, verifying if the protocol execution is as expected. This implementation however only focuses on the routing protocol, staying limited to the network layer operation, while our proposal acts directly in the network infrastructure system to detect vulnerabilities.

Also regarding network infrastructure security, the paper [16] proposes a NIDS for active routers using the data mining algorithm random forest for intrusion detection. First, this NIDS captures packets from the network and these packets are submitted to the random forest algorithm aiming at the creation of a dataset for the network existing services. This dataset is then used as the default network behavior to be used for the detection of intrusions which are characterized as anomalous behavior when compared to the created dataset. This cited paper argues that, due to the level of complexity of the misbehavior detection algorithm, the traffic analysis can not be made as the packets arrive, so the detection happens sometime after the packets arrive in the security system. In this same paper a three-layer architecture is proposed to enhance the security of the network routers, comprising a protocol decoder, a misuse detector and an anomaly detector. This anomaly detector is based on network behavior analysis using the previously generated dataset. The misuse detection proceeds with serial packet payload analysis involving each of the network routers, based on the fact that the first-hop router produces a message digest of the packet payload using the Message-Digest algorithm (MD5). Thus this digest is analyzed in each network router to see if the packets have been tempered with. Although this approach detects misbehavior by comparison with the standard network services pattern created using random forests, it is, however, unable to detect problems occurring in the router host as our proposal which acts directly in the network device to detect its security flaws.

Regarding host-based IDS proposals, the work [17] introduces the idea of a Distributed Intrusion Detection System (DIDS) which gathers information from remote hosts on a local area network (LAN), being directly related to the work being done in this paper. In the cited paper, a central role is given to the director entity, which represents the system where the information converges to. Similarly, in our proposal a centralized station, the Controller, collects and processes information gathered from the hosts of the network infrastructure, but while the LAN communications in [17] uses the CMIP protocol, which is an application layer protocol that lacks internal security, our proposed architecture leverages the SSH protocol and its safeness since it assures message authenticity by distributing the public key of the centralized station to the cooperating hosts in the proposed configuration process.

A remote IDS is proposed in [18] by means of web service that provides data analysis to hosts that do not have a local IDS. In this configuration, the hosts and sensors collect preprocess data that is sent to the centralized data analysis web service, which has a role similar to the Controller in our proposal. It is worth to point out that our centralized station is able to processing the data from a set of hosts and uses a neural network to analyze and perform machine learning so it allows more precise analysis and countermeasures as new agents use the IDS.

IV. Distributed HIDS for IoT Backbone Components

This work proposes the implementation of a Distributed HIDS for IoT Backbone components. In view of the concern with the security of IoT networks, the HIDS proposed must be located in a very usual infrastructure nowadays, an IoT network that has conventional routers and/or switches to connect the specific IoT equipment. So this paper focuses on modules for these routers and switches, not IoT devices, considering that complementary protections for the IoT specific devices must be provided with countermeasures specific to these devices such as proposed in the previous and correlated paper [19]. This cited paper describes a local signature-based HIDS that runs in IoT smart devices. Once a Smart device is inserted in the network, it must download the HIDS application. Then, each device updates its set of local rules with the remote rules that are constantly revised, maintaining all the final devices up-to-date with the security requirements. Thus, this paper proposal must be considered as being part of a comprehensive security solution for IoT in which different IDS configurations are established among different layers of the IoT infrastructure.

The architecture of the HIDS proposed in the present paper is shown in Fig 1 with the backbone components, i.e., conventional switches and routers in an IoT network instance, and our proposed Controller present and available by means if the Internet infrastructure. The Controller maintains databases for the data coming from the backbone components, as well as for the rules that will be used by the distributed detection set of HIDS.

The HIDS will remotely perform a set of periodical security verifications in each of the network managed devices and generate a report on the security vulnerabilities that are found in routers or switches.

To perform this role remotely, a Controller uses the database containing the necessary information collected from the backbone components that must be checked and the other database containing the rules for the verification to be executed. The database for backbone components data stores the information according to the JSON format, as shown in Fig. 2, so it can be easily used as a Python dictionary. Passwords are saved as hashes in this database.

A. HIDS parameters

The HIDS development was intended for the network administrator usage, as it needs confidential information from the backbone components. The information needed is:

- device type
- IP address
- username
- password
- secret

The Controller allows the administrator to list the devices that are registered in the database and then the administrator is able to insert or remove a device. The Controller will connect to each listed device through an SSH channel and run authenticated verifications. The administrator also has the option of not revealing the passwords of the backbone devices, then the Controller will execute some unauthenticated verifications. In this case. as the Controller cannot establish the connection to the device, it will try a brute force access to discover the username and password. If it is successful then all the verifications will be executed, otherwise just CPU usage and port scan verifications will be done.

In the end, a report is generated containing the detected security vulnerabilities highlighted in red and good security practices highlighted in green. Each device will have its specific report so that the administrator is able to take actions to correct security vulnerabilities considering all of those reports.

B. Security Verifications

The Controller performs a series of analysis on the remote backbone component to verify the integrity in the device's system. These verifications are presented bellow and classified according to the related risk.

1) Default username and password: One of the most common vulnerabilities is the use of default or common user identifier and password in devices. These devices are extremely vulnerable to malware that uses brute force dictionary attacks to gain access to a device. For instance, the malware Mirai malware uses a simple dictionary for the default username and password [2].

As mentioned above, as our proposed verifications will also be done when the administrator does not reveal the device password, in these case the Controller will use one brute force dictionary and try to establish the connection to the device. If the connection is successful it indicates that the device is vulnerable and has a default or common username and password.

In the case the administrator provides the device password registered in the Controller database, the connection will be established using the given password. Therefore, for the verification, instead of trying to establish a connection to the device using a brute force dictionary attack, the Controller will access all usernames and passwords in the device. For instance, ss the passwords are encrypted in usually 2 types in Cisco switches, type 5 and type 7, the Controller will try to convert the hash password to plain text, and then compare it with a brute force dictionary. If the password is successfully converted to plain text and is in the dictionary, it indicates a default or common username and password vulnerability.

It is important to emphasize that the data acquired in the brute force attempt is only used by the local Controller to find security vulnerabilities and remains in it. In no way, the information gathered will be used against the owner of the device, as it is expected that the Controller computer is physically and logically protected.

Based on the fact that a potential attacker can use the default username and password vulnerability to fully compromise the confidentiality, integrity, or availability of a target system, particularly in IoT instances, it is classified as a high-risk vulnerability.

2) Leaked password: Some attacks on companies that have a database of user information often lead to a leak of passwords, credit card numbers among other confidential information. Given this type of leak, a device password check is performed on the "haveibeenpwned.com" site.

The site checks if a password has already been compromised in some previous type of data leak. To not expose the password by sending it to the site, a hash technique is used. In this technique the user sends a request to the site with the first 10 characters of the password after it has been encrypted, then the site returns a list with the compromised passwords corresponding to the 10 hash characters sent. Then a local comparison is made to know if the user password is within the list, if so it implies that the password has been compromised by some leak.

Again it is important to highlight that the information will be used only locally and made available only for the administrator of the network to take actions.

As described for the precedent vulnerability, the flaw now described also creates loopholes that can completely compromise the integrity of the system. Therefore, it is also considered a high-risk vulnerability.

3) Known vulnerabilities: The National Vulnerability Database (NVD) provides to the community a large list of

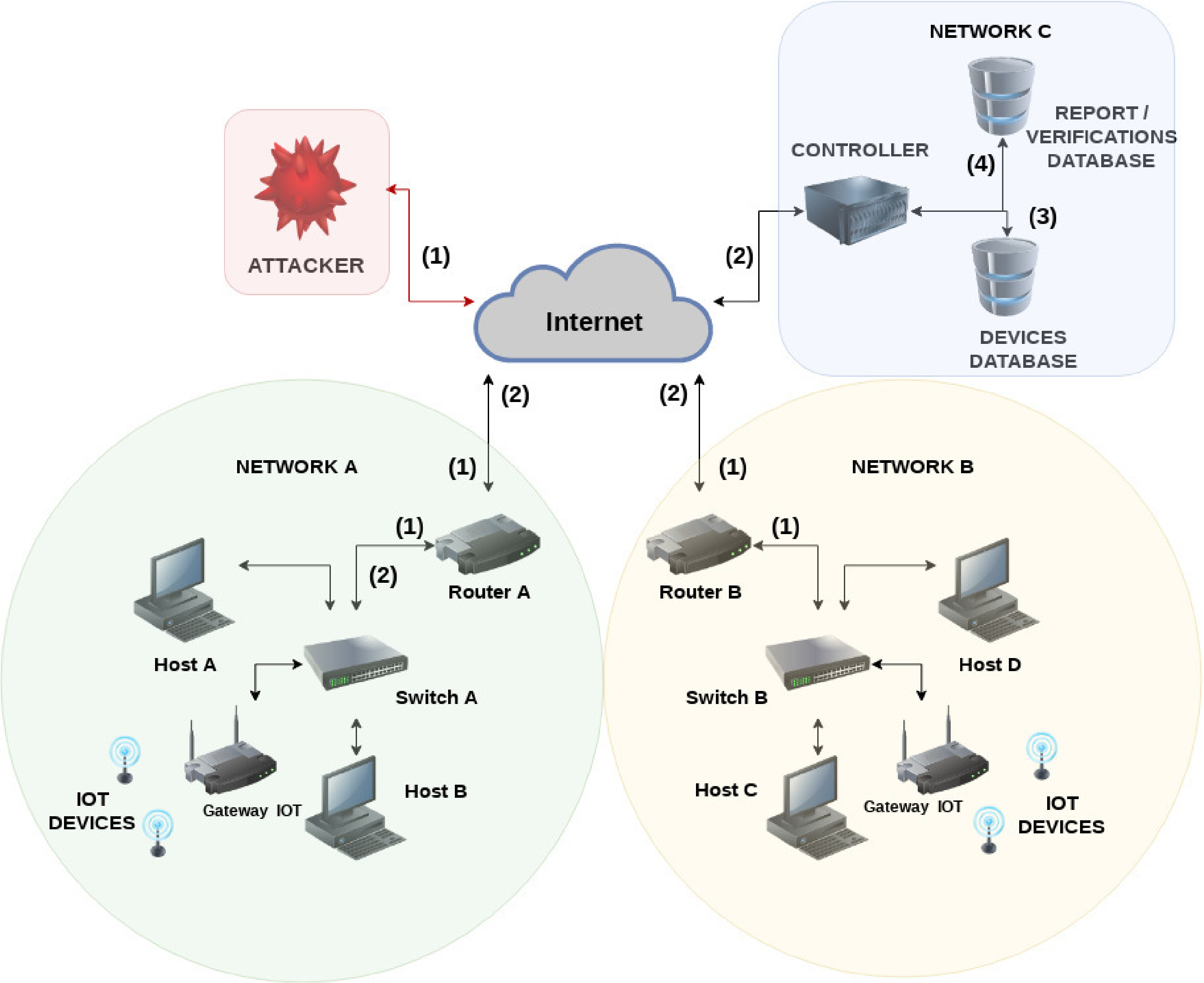
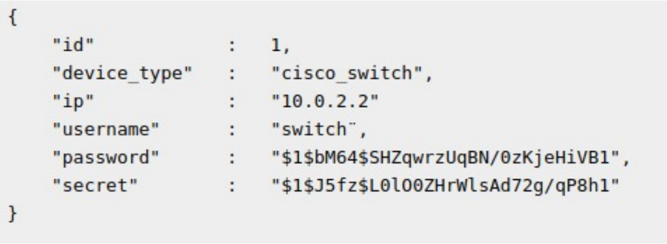


Fig. 1. The Logical architecture of our proposal. 1 represents the the attacker entry point. 2 represents the connection between the Controller and the network devices. 3 and 4 represent the data flow between the Controller and its databases



```
{
    "id"             :   1,
    "device_type"    :   "cisco_switch",
    "ip"             :   "10.0.2.2"
    "username"       :   "switch¨,
    "password"       :   "$1$bM64$SHZqwrzUqBN/0zKjeHiVB1",
    "secret"         :   "$1$J5fz$L0lO0ZHrWlsAd72g/qP8h1"
}
```

Fig. 2. JSON of a backbone component

security vulnerabilities, classified by the vendor, the model and the version of systems. For each vulnerability, a Common Vulnerabilities and Exposures (CVE) is created. CVEs are used to identify and catalog vulnerabilities in software or firmware into a free dictionary for organizations to improve their security.

The Controller, knowing the vendor, the model and the version of the device, as gathered by the SNMP protocol, is able to request all the CVEs for each specific device being verified. Then it gets a list of the CVEs containing the CVE-id, the publication date and a description of the vulnerability. This information with data obtained in the verifications allows to determine if there is a vulnerability identified by a CVE for each device. This information will be given to the administrator to take action.

As these vulnerabilities are extremely variable, their severity depends on the degree of risk they represent in a specific routing configuration and can not be previously classified.

4) Resources used by routing devices: Resources used by a router or a switch may indicate a problem with the device or some possibly malicious action that is trying to overload it.

Therefore, in our proposal a check is provided for the amount of CPU and memory being used by the devices.

Using the SNMP protocol, the percentage of CPU usage is checked and, if it is above 80 percent, this indicates that something is compromising the performance of the device. Then, a scan of the running processes is performed to list the processes that are consuming more CPU and possibly are related to the cause of the problem.

The amount of memory is also checked, including he total amount, the amount used and the amount of free memory. If the percentage of usage is higher than 80 percent of the total amount, the report will flag it as a potential vulnerability.

These vulnerabilities do not always represent an imminent risk to the integrity of the system as a whole and may sometimes only represent the normal functioning of the system. Therefore, it was considered a low-risk vulnerability.

5) Port scan: The device ports are scanned to check for possible security vulnerabilities in open ports. This information will be used in parallel with information already known from the model and specific version of the device.

Often port scanners are used in attack planning phases and can lead to serious security incidents. Having open ports is considered a medium risk vulnerability.

6) Current configuration: Once the Controller has established a connection to the backbone component, it will analyze the current configuration on the device. It will look specifically for configurations that can lead to misbehavior on the device.

The Controller starts by checking if the SNMP community string is common and whether it gives write permission to the MIB, which would allow an attacker to make changes in the device, what is considered an indeed important vulnerability. The Controller then looks if the HTTP protocol is enabled in place of HTTPS, as HTTPS is more secure than HTTP. Then it is checked if the IP CEF is enabled, as if not it can cause slowness in the device.

Finally, the Controler analyses if the telnet protocol is enabled, as using this information with the known vulnerabilities for the specific model of the device it is possible to classify it as a problem or not in the device.

Having the write permission to a SNMP protocol grants power enough to change specific configurations and to even turn off a device.Therefore, it was considered as a high-risk vulnerability.

Table I summarizes the risk rating of each vulnerability classified by the proposed IDS.

V. Tests and Results

To validate our Distributed HIDS for IoT Backbone Components we developed a corresponding prototype and performed a battery of tests in a controlled environment, i.e., a local and isolated network without compromising public or other private backbone components.

We executed the developed software to evaluate 3 backbone components. Fig. 3 shows the architecture used in the tests.

TABLE I
Vulnerability risk rating

Vulnerability	**Risk**
Brute force	high
Common password	high
Leaked password	high
Common secret	high
Memory utilization	low
CPU utilization over the last minute, CPU utilization over the last 5 seconds, CPU utilization over the last 5 minutes	low
Community string in SNMP protocol	high
IP CEF	low
HTTP is enabled without HTTPS	medium
Telnet enabled	medium

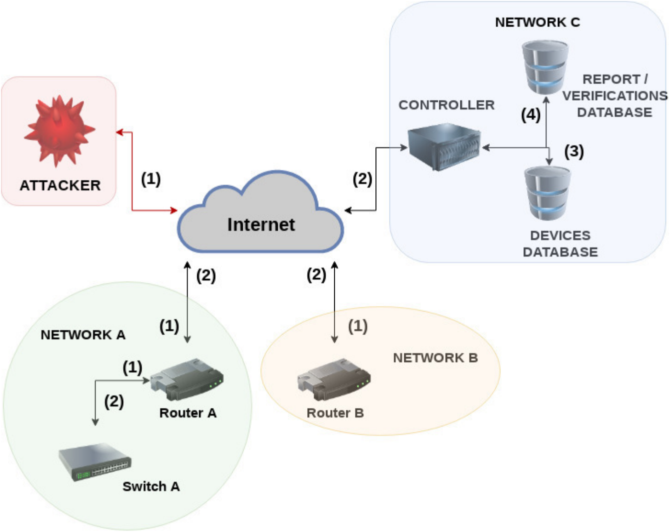


Fig. 3. Simplified architecture of the tests executed.

TABLE II
Expected format of the output in the report

Vulnerability	**Format**
Brute force	If it was used, if it was successful
Common password	Shows the common passwords found separated by comma
Leaked password	The password: amount of occurrences
Common secret	Default secret or no secret
Memory utilization	Memory utilization in percentage
CPU utilization over the last minute, CPU utilization over the last 5 seconds, CPU utilization over the last 5 minutes	CPU utilization separated by comma
Community string in SNMP protocol	The common strings followed by the configuration separated by comma
IP CEF	Configured or not
HTTP is enabled without HTTPS	Yes or no
Telnet	Open or not
Port scan	Open ports separated by comma
CVE Vulnerabilities	CVE-id, date, description

The first backbone component, called 'Device 1' is a Cisco Router where almost all the security vulnerabilities are present, except the exhaustion of resources which we could not simulate. The administrator has given the SSH device password and username to the IDS, so that the brute force technique is not utilized.

The second backbone component analyzed by our HIDS was the Cisco Switch named 'Device 2'. In this case, the administrator do not give the SSH username and password to the IDS, so it utilizes the brute force dictionary trying to guess the credentials. As it is successful messages about default username and password are printed in the report, as shown in Table III. Now, having the credentials of the device, the HIDS continue to execute the other verifications.

The third backbone component is a Cisco Router named 'Device 3' and considered by the authors a router with configurations close to those expected in the majority of real environments, which means that the vulnerabilities found in it are also the ones expected by the authors to be the most present when implementing the HIDS in a realistic situation.

Having the information needed from the 3 devices cited above in the devices database, the HIDS is able to establish an SSH connection to each device, one by one, and execute the verifications. In Table II we present the format expected of the outputs in the reports. A report is generated for each device and Table III summarizes the results.

The first interesting result regards the brute force technique that was used only against the Device 2, for which the Controller did not have the username and password. The proposed IDS discovered the username and password cisco and with this password the Controller continued to execute the verifications.

Then, as the common passwords of all users are checked, the IDS found in all 3 devices at least one user with a common password. The common passwords were also found to be leaked passwords, reinforcing the huge exposure that they represent.

The secret password was verified after that and again resulted in the default secret, which is cisco. Then the Controller looks for resources usage, analyzing the CPU and memory being used in our tested devices, concluding that there was no problem since none of them were using more than 80 percent of the total capacity.

A really important vulnerability that is surprisingly common in backbone components is the community string in the SNMP protocol being set as public for read and private for write operations in the MIB. An attacker with write permission access to a device can do massive damage, including turning off the device. In all 3 reports the warning to this vulnerability appeared, followed by the configuration of the SNMP protocol in each device.

Next, it was reported in the Device 1 that the IP CEF was not configured and the HTTP was enabled without HTTPS, representing a security vulnerability.

Finally, the open ports of the devices are presented in the reports. It is important to notice also that the HIDS found a CVE vulnerability for the device 1, as it is written in the description of the CVE that there is a possibility of DoS attack through Telnet. With that information and the information previously shown in the report that the Telnet is enabled and the port 23 is open, the administrator can take actions immediately.

In this validation process, the proposed HIDS set has proven to be capable of solving the problems presented in Section I related to common and default credentials. This distributed HIDS also provides key information relative to security breaches identified in generated reports for the administrator.

VI. Conclusion and Future Works

A distributed HIDS is proposed in this paper to execute a series of verifications and look for vulnerabilities in multiple devices in IoT backbones. A centralized station called Controller is used for storing and managing information regarding participant devices, being the main tool for the network administrator. The Controller interacts with a set of devices through SSH connections and for each device it runs verifications regarding vulnerabilities and security-related events.

Some basic vulnerabilities that are verified by the proposed system, like default or common usernames and passwords, are highly emphasized in this paper due to the large number of attacks that are taking advantage of them, especially by IoT botnets like Mirai.

By running tests in a local and controlled environment, it was possible to analyze three types of generated reports. The HIDS showed its ability to tackle the common username and password issue, either by executing tests directly in the devices for which the HIDS Controller had the necessary credentials, thus being able to find all weak passwords. Also, by using a brute force dictionary procedure on a device for which the Controller hadn't the credentials but managed to discover bad passwords, thus alerting the administrator of the possibility of brute force attacks to this device.

Based on other verifications regarding open ports, enabled protocols and information about the resources verified by the HIDS, the respective reports contained crucial information for the administrator of the network to take actions.

In future works, the distributed HIDS is going to be tested in a larger set of routers and switches from other different vendors. Also, considering the evolving relationship among vulnerabilities and attacks, a machine learning approach is under consideration for the analysis of the gathered information as a whole to allow the Controller to autonomously execute countermeasures against attacks.

Acknowledgment

The authors gratefully acknowledge the support from the IEEE VTS Centro-Norte Brasil Chapter, the Brazilian Research Councils CNPq (Grant 465741/2014-2 INCT on Cybersecurity), CAPES (Grant 23038.007604/2014-69 FORTE), and FAP-DF (Grants 0193.001366/2016 UIoT, 0193.001365/2016 SSDDC, and Call 01/2019), as well as the LATITUDE/UnB

TABLE III
RESULTS OF THE EXECUTED TESTS

Vulnerability	**Device 1**	**Device 2**	**Device 3**
Brute force	Not used	Used and successful. Username: cisco Password: cisco	Not used
Common password	1234, 1234	cisco	cisco, cisco
Leaked password	1234: 1256907	cisco: 5016	cisco: 5016
Common secret	Default secret	Default secret	Default secret
Memory utilization	0.13648%	0.09412%	0.28204%
CPU utilization over the last minute CPU utilization over the last 5 seconds CPU utilization over the last 5 minutes	21%, 54%, 9%	2%, 17%, 1%	2%, 2%, 1%
Community string in SNMP protocol	public RO, private RW	public RO, private RW	public RO
IP CEF	No	Yes	Yes
HTTP is enabled without HTTPS	Yes	No	No
Telnet	Open	Open	Closed
Port scan	22, 23, 80	22, 23	22
CVE Vulnerabilities	CVE-2000-0268. 2000-04-20T00:00:00. Cisco IOS 11.x and 12.x allows remote attackers to cause a denial of service by sending the ENVIRON option to the Telnet daemon before it is ready to accept it, which causes the system to reboot.	Not found	Not found

Laboratory (Grant 23106.099441/2016-43 SDN), the Ministry of the Economy (Grants 005/2016 DIPLA, 011/2016 SEST and 083/2016 ENAP), and the Institutional Security Office of the Presidency of the Republic of Brazil (Grant 002/2017).

REFERENCES

[1] Kaspersky. DDoS attacks in q1 2018. [Online]. Available: https://securelist.com/ddos-report-in-q1-2018/85373/

[2] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, and D. Menscher, "Understanding the Mirai Botnet," in *Proceedings of the 26th USENIX Security Symposium*. Vancouver, BC, Canada: USENIX, 2017, pp. 1093–1110.

[3] S. Hilton. Dyn Analysis Summary Of Friday October 21 Attack. [Online]. Available: https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/

[4] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[5] "Annual CyberSecurity Report," Cisco 2018, Tech. Rep., 2018. [Online]. Available: https://www.cisco.com/c/dam/m/digital/elq-cmcglobal/witb/acr2018/acr2018final.pdf

[6] P. N. Raju, "State of the Art Intrusion Detection: Technologies, Challenges, and Evaluation," Master's Thesis, Linköping University, Linköping, Sweden, 2005.

[7] R. Gerhards, "The Syslog Protocol - RFC 5424," Internet Engineering Task Force (IETF), Tech. Rep., March 2009.

[8] H. G. C. Ferreira and R. T. de Sousa Junior, "Security analysis of a proposed internet of things middleware," *Cluster Computing*, vol. 20, no. 1, pp. 651–660, Mar 2017.

[9] A. Lazarevic, V. Kumar, and J. Srivastava, *Intrusion Detection: A Survey*. Boston, MA, USA: Springer US, 2005, pp. 19–78.

[10] A. Patel, Q. Qassim, and C. Wills, "A survey of intrusion detection and prevention systems," *Information Management & Computer Security*, vol. 18, no. 4, pp. 277–290, 2010.

[11] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16 – 24, 2013.

[12] G. G. Helmer, J. S. K. Wong, V. Honavar, and L. Miller, "Intelligent Agents for Intrusion Detection," in *1998 IEEE Information Technology Conference, Information Environment for the Future (Cat. No.98EX228)*. Syracuse, NY, USA: IEEE, Sep. 1998, pp. 121–124.

[13] "Cisco Port Security," Cisco, Tech. Rep., 2018. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/routers/7600/ios/15S/configuration/guide/7600_15_0s_book/port_sec.html

[14] "DHCP Snooping," Cisco, Tech. Rep., 2018. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst6500/ios/12-2SX/configuration/guide/book/snoodhcp.html

[15] Y. F. Jou, F. Gong, C. Sargor, X. Wu, S. F. Wu, H. C. Chang, and F. Wang, "Design and implementation of scalable IDS for the protection of Network infraestructure," in *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, vol. 2. Hilton Head, SC, USA: IEEE, Jan 2000, pp. 69–83 vol.2.

[16] G. Prashanth, V. Prashanth, P. Jayashree, and N. Srinivasan, "Using Random Forests for Network-based Anomaly detection at Active routers," in *2008 International Conference on Signal Processing, Communications and Networking*, Chennai, India, Jan 2008, pp. 93–96.

[17] S. R. Snapp, J. Brentano, G. V. Dias, T. L. Goan, L. T. Heberlein, C.-L. Ho, K. N. Levitt, B. Mukherjee, S. E. Smaha, T. Grance, D. M. Teal, and D. Mansur, "DIDS (Distributed Intrusion Detection System)-Motivation, Architecture, and An Early Prototype," in *Proceedings of the 14th National Computer Security Conference*. Washington, DC, USA: NIST, 1991, pp. 167–176.

[18] M. Silva, D. Lopes, and Z. Abdelouahab, "A Remote IDS Based on Multi-Agent Systems, Web Services and MDA," in *2006 International Conference on Software Engineering Advances (ICSEA'06)*, Tahiti, Tahiti, Oct 2006, pp. 64–64.

[19] B. V. Dutra, J. F. de Alencastro, F. L. de Caldas Filho, L. M. C. e Martins, R. T. de Sousa Júnior, and R. de O. Albuquerque, "HIDS by signature for embedded devices in IoT networks," in *Actas de las V Jornadas Nacionales de Investigación en Ciberseguridad (JNIC 2019)*. Cáceres, Spain: Universidad de Extremadura, Jun 2019, pp. 53–61.