TOWARDS CENTRAL VULNERABILITY MANAGEMENT BY MOBILE PHONE OPERATORS

Thamsanqa Moyo, Barry Irwin, Madeleine Wright

Computer Science Department
Rhodes University
Grahamstown

g02m1612@campus.ru.ac.za, b.irwin@ru.ac.za,m.wright@ru.ac.za

ABSTRACT
The application of XML-based approaches in passing vulnerability information between vulnerability management devices or software residing on wired networks has been demonstrated. We propose a proof of concept framework for mobile operators that extends this use of XML into the area of vulnerability management on public land mobile networks. Our proposed framework allows for a pro-active central management of vulnerabilities found on mobile stations such as mobile phones. Despite the relatively limited number of reported vulnerabilities on mobile stations, such a pre-emptive approach from mobile operators is necessary to acquire the confidence of early adopters in Mobile Commerce. Given the diverse collection of devices and software that exist on a public land mobile network, XML-based approaches are best able to providing the inter-operability required for vulnerability management on such a network. Our proposed framework leverages web services by using the Open Vulnerability Assessment Language (OVAL) to provide vulnerability descriptions, and by securing these descriptions in SOAP messages conforming to the OASIS Web Services Security (WSS) standard. We contribute in three areas: firstly, through this framework we show that mobile operators can carry out centralized vulnerability management on their public land mobile networks comprising of a wide variety of devices and software. Secondly, the assurance of integrity, confidentiality and non-repudiation inherently lacking in OVAL vulnerability descriptions is achieved through their encapsulation in SOAP messages conforming to the OASIS WSS standard. Thirdly, SOAP-based web service implementations allow for integration with vulnerability management tools and devices that do not conform to OVAL.

KEY WORDS
Vulnerability Management, Mobile Computing Security, Web Services, OVAL

TOWARDS CENTRAL VULNERABILITY MANAGEMENT BY MOBILE PHONE OPERATORS

1 INTRODUCTION

Reported vulnerabilities on mobile stations[1](MS) found on public land mobile networks (PLMNs) are less common than those reported on devices typically found on wired networks. However, trends show that the exploitation of vulnerabilities on MSs is increasing [2, 3]. This increase may be attributed to the improving technical capabilities of typical MSs and the growth of m-commerce [4].

The exploitation of MS vulnerabilities may result in consequences which include increased billing, stolen subscriber data and denial of service attacks on the PLMN itself [5]. Such consequences may lead to losses in potential revenue for mobile operators from m-commerce activity, as consumers lose confidence in using their MSs. Therefore it is in a mobile operator's interest to assume a pro-active approach in managing vulnerabilities that may exist on MSs.

We propose a framework that allows mobile operators to assume management of vulnerabilities on subscriber MSs. Our framework utilises XML-based approaches to provide platform- and vendor-independent vulnerability management. In addition, we address challenges faced when implementing centralised vulnerability management on PLMNs.

Our first contribution is in showing that, through this framework, it is possible for a mobile operator to protect subscribers through platform-independent, centralized vulnerability management on its public network. In addition we contribute by showing that the challenges faced in attempting central vulnerability management on a public network are handled by our framework.

Our work is presented as follows: Section 2 gives a description of related work; Section 3 details the design of our proposed framework and a discussion is provided in Section 4 .

2 RELATED WORK

We split the work that relates to ours into two categories: the first includes current strategies employed by mobile operators in attempting vulnerability management on their PLMNs. The second category is the application of XML-based approaches in vulnerability management on LANs.

Tian et al describe the vulnerability management process as follows:

> Vulnerability management means discovery, disclosing, remediation, and publication of vulnerabilities in their whole life cycle [6].

Since our framework is based on this definition, we use it to evaluate the related work. We first review the current work carried out by mobile operators.

2.1 Mobile Operator Vulnerability Management Strategies

Current vulnerability management efforts in the mobile industry are immature, and mobile operator efforts are currently focused on the remediation of vulnerabilities through the deployment of anti-virus software [7]. The F-Secure corporation [8, 9] details in two case studies its partnerships with Telia-Sonera Finland and Telecom Italia Mobile in providing a centralised anti-virus management system

[1]In GSM terminology, mobile stations are devices connected to the GSM network via the air interface[1]

on the respective operator's PLMN. These efforts involve the deployment of anti-virus software on MSs and the propagation of over-the-air (OTA) virus definition updates. All this activity is managed from a central location.

2.1.1 Assessment

This strategy is similar to the one we present in this paper as it advocates that mobile operators should take a pro-active stance in securing subscribers MSs. However, the approach of centralised anti-virus management deals only with the remediation aspect of vulnerability management. Since our framework is concerned with a holistic approach to vulnerability management, we review other efforts that take this approach, in the area of vulnerability management on LANs.

2.2 Vulnerability Management on LANs using XML-based Approaches

The rapidly increasing number of vulnerabilities discovered on devices resident on LANs has led to the development of a large and diverse selection of security tools [10]. Since most tools communicate in a non-interoperable manner, a need has arisen for the automatic coordination of these tools. The use of XML-based approaches such as OVAL and web services promises a platform offering a vendor-independent communication mechanism. We review two efforts that highlight this promise.

Martin details the role that the OVAL language will play in the United States Department of Defense's move towards standardising and automating its vulnerability management [11]. OVAL allows a standardised representation of these vulnerabilities in a format that compliant vulnerability management tools will understand. This will enable the automatic collaboration of a heterogeneous collection of vulnerability management devices in all areas of the vulnerability management process.

Tian et al, propose another XML language, CVML, to carry out automated central vulnerability management [6]. CVML presents information on the evaluation of a vulnerability, the way a vulnerability is checked for, how to remediate the vulnerability and details of any exploits. Therefore, CVML presents more information for the vulnerability management process than does OVAL which represents only a description of the vulnerability or the configuration state of a system. This extra information provided by CVML makes the automation of the vulnerability management process easier.

2.2.1 Assessment

The work described above is similar to ours as it seeks to manage vulnerabilities centrally on a network employing a heterogeneous collection of vulnerability management tools. However each approach has limitations when applied to the PLMN scenario:

1. The OVAL schema and CVML do not provide a mechanism for securing themselves: there are no mechanisms in either language that assure the integrity of a message and the identity of its sender. This lack of built-in security has more significance in PLMNs where public access to the network is easier than that to an enterprise LAN.
2. The use of the OVAL schema or of CVML implies that vulnerability management tools are OVAL- or CVML-compliant respectively. [6] mention that the adoption of CVML remains a major challenge to their efforts. While vulnerability scanners are adopting OVAL as an output

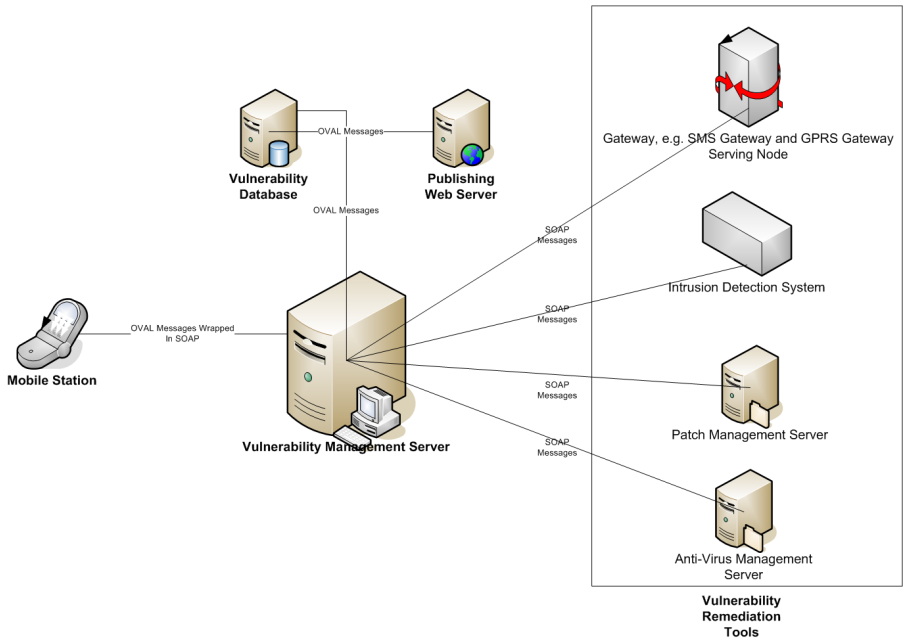


Figure 1: Logical Components of PLMN vulnerability management framework

data format, there is a lack of remediation tools that understand OVAL[12] . Therefore, the use of both languages in central vulnerability management is limited by the number of tools that support them.

The framework we present extends the use of XML-based approaches into the area of vulnerability management on PLMNs, addressing the challenges mentioned above. We detail the design of our framework next.

3 DESIGN

This section gives an overview of our entire design. We set out the objectives the framework is designed to meet and highlight the communication mechanisms and components utilized in it.

3.1 Design Objectives

The objectives to be met by our framework are as follows:

1. The framework must be independent of the underlying structure and protocols of the PLMN . For example, the mechanisms for securing a GPRS backbone are left up to the discretion of a mobile operator by the GPRS standard [13]. As a result our framework cannot assume that data it passes along this backbone will be secured by the underlying PLMN as operators may implement different mechanisms for securing their backbone, if any at all. Therefore this objective is necessary in the mobile network scenario to keep the framework as PLMN independent as possible, since operators may approach implementation issues of their PLMNs in different ways.

2. The framework must utilize a standard mechanism for representing vulnerabilities. This is important to cover the heterogenius collection of devices that may exist on a PLMN.
3. The framework must not rely on external entities to secure it and must provide its own security mechanisms. While the framework aims to secure a PLMN, it may be undermined by attacks on it. For example, an attacker may achieve a denial of service attack by inserting multiple vulnerability scan results and a result forcing a mobile operator to block services in order to prevent the exploitation of falsely reported vulnerabilities. Therefore, this objective ensures that the framework does not create new avenues of attack on the PLMN.
4. Related to the objective above, the messages sent by the framework must be secured to ensure their integrity, confidentiality and non-repudiation.
5. The framework must be able to integrate with existing vulnerability remediation tools. As discussed in Section 2.2.1, the lack of common language adoption by remediation tools is a major hindrance to centralised vulnerability management. Therefore, the framework must incorporate current technology used to remediate vulnerabilities.

Figure 1 shows the logical layout of the framework which will meet these objectives. The rest of this section is concerned with the details of this layout and the communication mechanisms utilized in it. We start by presenting the vulnerability description language chosen

3.2 Vulnerability Description Language

In order to meet our design objective 2, OVAL was selected as our vulnerability language of choice for the framework since it has a close mapping to the Common Vulnerabilities and Exposures list (CVE) [10]. Each CVE list entry relating to an OVAL-supported platform has a description in the OVAL format. Therefore, the OVAL schema covers a comprehensive number of vulnerabilities it can describe and is considered to be the most widely adopted XML language used for patch management. At the time of writing there were no official schemata [14] for platforms more commonly found on MSs. However, CVE provides us with a base from which we can easily develop additions to the OVAL schema that cover MS platforms.

The OVAL schema has the following shortcomings:

1. The OVAL schema provides no means of securing itself. Therefore there is no mechanism of determining the integrity of an OVAL message or verifying the identity its sender within the OVAL schema itself. This is particularly of concern in the PLMN situation where the network is publicly accessible.
2. The OVAL schema is only understood by remediation tools that support it and as a result mobile operators may need to replace existing tools with OVAL compliant ones.

To overcome these shortcomings our framework utilises the SOAP and this is detailed next.

3.3 Communication Protocol chosen

In order to mitigate the shortcomings of OVAL, our framework wraps OVAL messages in the body section of a SOAP message. The SOAP protocol is built on XML and is a standard protocol used in providing inter-operability between different platforms and devices [15]. Our framework utilises the

SOAP protocol to achieve two distinct goals: the security of OVAL messages and the integration of non-OVAL-compliant tools.

3.3.1 Securing OVAL messages using SOAP

The WS-Security standard [16] provides mechanisms for securing the integrity and confidentiality of SOAP messages. The standard uses the XML Encryption [17]and XML Signature[18] standards to secure these messages. Instead of prescribing specific security tokens, the standard details how to attach them to a SOAP message and non-repudiation can be achieved through attachment of tokens such as X.509 certificates [19]. These tokens are kept in a security header of the SOAP message to be secured.

In order to achieve our design objective 4 of providing integrity, confidentiality and non-repudiation for messages sent in our framework, we specify that OVAL messages must be encapsulated inside the message body of SOAP messages. In turn the SOAP message is secured in a manner compliant with the WS-Security standard. The second goal achieved by the use of the SOAP protocol is described next.

3.3.2 Integration of non-OVAL-Compliant devices

In order to meet our objective 5 of integrating non-OVAL compliant remediation tools, we specify the use of SOAP to facilitate communication with these devices . Two steps are required to enable this communication:

Message Transformation: A new SOAP message containing actions that a specific remediation tool must carry out is constructed by the vulnerability management server . These actions are derived from a OVAL results message and the corresponding security policy associated with the vulnerability described by the OVAL message. The actions are formatted as SOAP elements, secured using mechanisms specified by the WS-Security standard [16] and sent to the relevant remediation tool.

Message Interpretation: The new SOAP message containing actions to be carried out is translated into instructions that a remediation tool can understand by an intermediary SOAP gateway . If any feedback is provided by the tool, this feedback is translated into a SOAP message by the SOAP gateway and relayed back to the vulnerability management server.

The components that are connected using the mechanisms mentioned above are detailed next.

3.4 Components

Some components may physically reside in locations dependent on where a mobile operator finds it convenient to place them on its PLMN. Therefore, in order to meet objective 1, we specify the components in a logical manner (Figure 1) that will allow the mobile operator some flexibility in its deployment of the components. The role that each component plays in the framework is described next:

3.4.1 Mobile Station:

This is the point where vulnerabilities exist. Two types of software found on the MS are vulnerability scanning tools and a description propagation agent.

Given the heterogeneous collection of MSs that exist on a PLMN and the wide range of vulnerabilities that may exist on a single type of MS, our framework is structured to accommodate a diverse group of scanning tools. Therefore, we do not specify specific scanning tools that must be used. However,we do specify two criteria for tools that will be compatible with our framework.

1. The scanning tools must support OVAL, which provides a common vulnerability description format, in line with meeting our objective 2.
2. The scanning tools must be able to receive OTA updates . This provides a mobile operator an easy means of updating the schema that a scanning tool uses.

The description propagation agent is responsible for transporting the OVAL messages to the vulnerability management server in a secure manner. Therefore, the agent reads in the OVAL message output by a scanning tool and encapsulates it in a SOAP message. The SOAP message is then secured by the agent in a manner compliant with the WS-Security standard and transported to the vulnerability management server. The vulnerability management server is described next.

3.4.2 Vulnerability Management Server

The vulnerability management server is responsible for coordinating the entire vulnerability management system and this is where the majority of the processing logic resides. Its roles are defined in more detail as follows:

Policy Management: Security policies are kept on this server. The policies are related to each known vulnerability to be handled by the system. For example, each CVE entry relevant to a MS will have a corresponding policy on how to remediate and whether to publish the vulnerability. Therefore, the server is responsible for determining the action to be taken on the discovery of a vulnerability.

Remediation Tool Interface: The server is responsible for encoding the actions that non-OVAL-Compliant remediation tools must carry out in SOAP. If OVAL-compliant remediation tools exist, the server may simply encapsulate the OVAL messages in a SOAP message without transforming the messages.

Schema Update As new vulnerabilities are found and OVAL schemata expand, a need will arise for the upgrade of the OVAL schema and of the scanning tools . This distributed upgrade is the responsibility of the server.

Message Security Management: The server is responsible for verifying the security claims that are made by an incoming message from a MS. The server will reject any message whose sender cannot be verified or whose integrity is in question.

External Tool Interface: Figure 1 shows a vulnerability database and a web server publishing information concerning the operation of the system or the vulnerabilities found on the system. Such tools are beyond the scope of the framework. However, they are included to demonstrate that our framework may also interface with systems complementing its efforts.

The final component described concerns the vulnerability remediation tools.

3.4.3 Vulnerability Remediation Tools

The tools found in this component are responsible for the remediation of vulnerabilities found on the system. Such tools may include patch management systems, firewalls and anti-virus management systems (Figure1) . Given the diversity of the tools that may exist, each tool has a SOAP gateway with which it interfaces. When a new remediation tool is added, a SOAP gateway for that device is developed. The gateway in turn provides a standardised interface through which the vulnerability management server can communicate with the tools. Therefore, the main role played by the SOAP gateway is to translate SOAP messages received from the vulnerability management server into commands understood by the remediation tools.

The role of the gateway changes slightly if the tool it interfaces with is OVAL compliant. If this is the case, the vulnerability management server will wrap and secure the OVAL message in a SOAP as discussed in 3.3.1. The SOAP gateway is responsible for verifying the security claims of the SOAP message it is sent . Once this has been done, the SOAP gateway simply passes on the OVAL message to the device.

We provide a discussion of this framework next. In order to aid the discussion we provide a suggested implementation of the framework.

4 DISCUSSION

We provide a brief discussion of the design of our framework in this section. In order to produce a meaningful discussion, we detail a suggested implementation of a our framework. This implementation is not concrete and only provides suggested detail of how the framework may be implemented. We assume that the central vulnerability management service is administered as a value added service. Therefore a subscriber must sign up for the service and each MS they own is assigned a unique X.509 certificate. This certificate will reside on the MS's storage device.

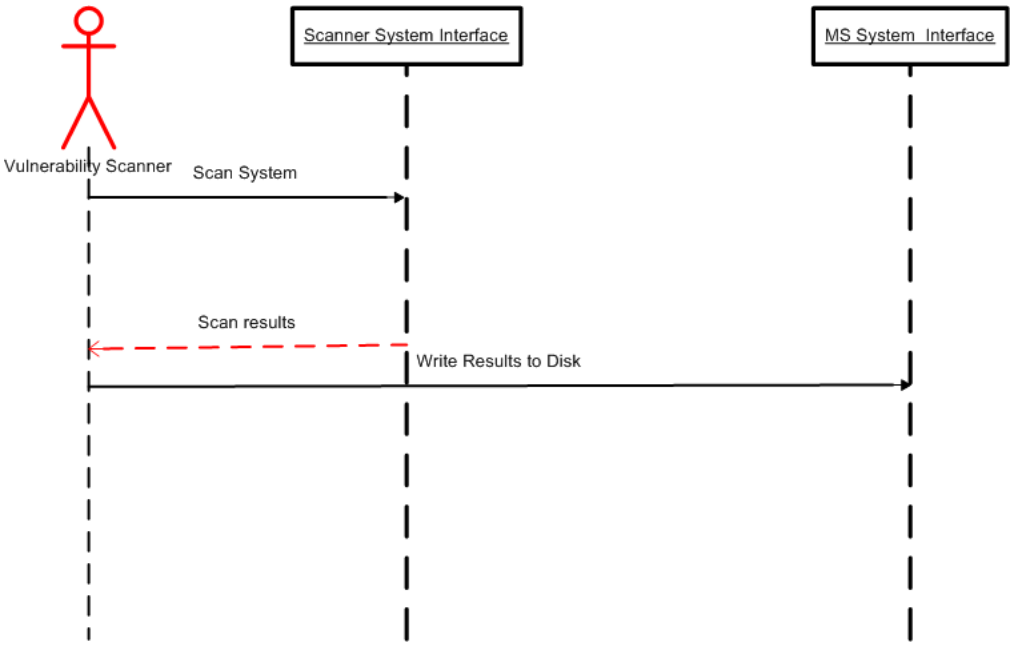
The rest of our discussion is split into 2 areas: Vulnerability Discovery and Vulnerability Remediation. We do not discuss the disclosure or the publication of vulnerabilities. While our framework is designed to handle these issues, as highlighted in 3.4.2, they fall outside our discussion as they are strongly influenced by the manner which the mobile operator decides to manage them.

4.1 Vulnerability Discovery

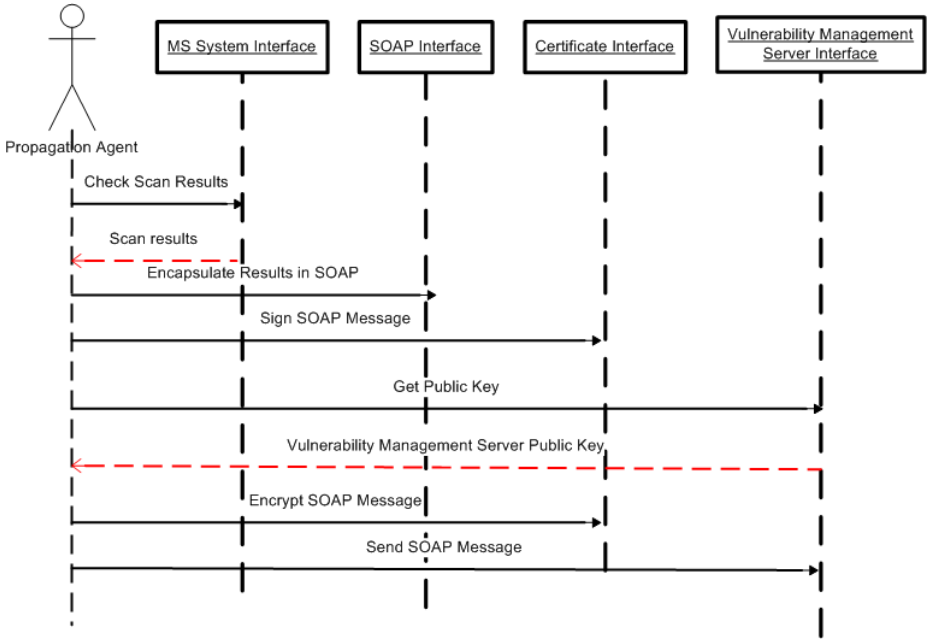
Figure 2 shows the interactions that will take place during the discovery of a vulnerability. Figure 2 (a) details the work carried out by an OVAL compliant scanner, which simply dumps its OVAL formatted results onto the MS storage device. The OVAL scan results are picked up by the description propagation agent, encapsulated into a SOAP message and signed with MS's certificate (Figure 2 (b)). The description propagation agent will encrypt the SOAP message with the vulnerability management server's public key and send the SOAP message to the vulnerability management server.

4.1.1 Advantages

By wrapping OVAL messages in SOAP we overcome the shortcoming of OVAL that it cannot secure itself. We also keep the processing carried out on MSs to a minimum as these devices have relatively low powered CPUs. Only three resource intensive operations are carried out on the MS:



(a) Scanner Interactions



(b) Propagation Agent Interaction

Figure 2: Interactions in Vulnerability Discovery

scanning, signing of the SOAP message and encryption of the SOAP message. The rest of the logic and processing is deferred to the vulnerability management server.

4.1.2 Open issues

It may be possible for an attacker to insert malicious vulnerability messages into the system by forging the output of a scanner. While it is possible to identify the attacker through their certificate if they send such messages from their own MS,they can insert messages onto another individuals MS via removable storage media.

In addition, the scanning, signing and encryption processes may be too resource intensive for some lower powered MSs.

Messages sent as a consequence of the discovery of a vulnerability are handled in our next area of discussion, vulnerability remediation.

4.2 Design: Remediation

Once a SOAP message is received from a MS and verified by the vulnerability management server, the server identifies the policy associated with the vulnerability described and formulates an action for remediation. The action is sent to the SOAP gateway of the appropriate remediation tool. Examples of such action include the propagation of patches to the appropriate MSs, service blocking and informing the owner of their vulnerable MS.

4.2.1 Advantages

Using SOAP to provide integration with non-OVAL-Compliant devices, plugs the current shortcoming of the slow adoption of OVAL by remediation tools. Our approach also provides modularity in the management of remediation tools used within the framework. When a new device is added a new SOAP gateway is written and when a device is removed, the SOAP gateway is removed. Therefore, tools can be easily added and removed from the framework with minimal changes to the vulnerability management server and no change to scanning tools.

4.2.2 Open Issues

It is questionable whether SOAP is needed to provide interoperability since each tool has a gateway that transforms incoming messages from the vulnerability management server into a format understood by the tool. Therefore it is plausible to pass OVAL messages to the gateway and have them transformed into tool-specific commands. However, the advantages of modularity are lost if this approach is taken as changes will need to be made to every client gateway when the changes are made to the schema. Our proposed approach requires that only the vulnerability management server and scanning tools need be adapted to changes in the OVAL schema.

5 CONCLUSION

We present a centralised vulnerability management framework which enables mobile operators to manage the vulnerabilities on its subscribers MSs. Our framework still needs to be tested through

an implementation. Issues such as the creation of OVAL schema for common MS platforms can be resolved through implementation. In addition, the extent to which the scanning, encryption and signing processes burden the resources of various MSs needs to be ascertained in order to determine the framework's current feasibility.

Our framework contributes in three ways: Firstly we show that it is conceptually possible for mobile operators to proactively manage vulnerabilities on their subscriber's MSs. Secondly our framework overcomes the shortfall of OVAL's inability to secure itself by encapsulating OVAL messages in a SOAP message body. Thirdly, our framework, through web services, integrates non-OVAL compliant tools in vulnerability management system utilizing OVAL compliant scanners

References

[1] B. Ghribi and L. Logrippo, "Understanding GPRS: the GSM packet radio service," *Computer Networks (Amsterdam, Netherlands: 1999)*, vol. 34, no. 5, pp. 763–779, 2000.

[2] Trend Micro Incorporated, "Q1 2005 virus roundup," quarterly report, Trend Micro Incorporated, 2005.

[3] Trend Micro Incorporated, "2005 Annual Roundup and 2006 Forecast," annual report, Trend Micro Incorporated, 2006.

[4] N. Leavitt, "Will Proposed Standard Make Mobile Phones More Secure?," *IEEE Computer*, vol. 38, no. 12, pp. 20–22, 2005.

[5] N. Leavitt, "Mobile Phones: The Next Frontier for Hackers?," *IEEE Computer*, vol. 38, no. 4, pp. 20–23, 2005.

[6] H. Tian, L. Huang, J. Shan, and G. Chen, "Automated Vulnerability Management through Web Services.," in *GCC (1)*, pp. 1067–1070, 2003.

[7] M. Laakso, "Vulnerabilities Go Mobile," in *AusCERT2002*, 2002.

[8] Fsecure Corporation, "Case Study: Fsecure protects the smart phones of Telecom Italia Mobile," 2004. Available:http://www.f-secure.com/marketing/materials/case-studies/cs-tim_2005-11-23.pdf. Last accessed 24/05/2006.

[9] Fsecure Corporation, " Case Study: Solutions for Mobile Operators," 2004. Available:http://www.f-secure.com/marketing/materials/case-studies/teliasonera-2005-10-21.pdf. Last accessed 24/05/2006.

[10] P. Michalek, "Dissecting application security xml schemas," *Information Security Technical Report*, no. 3, pp. 99–109, 2004.

[11] R. A. Martin, "Transformational Vulnerability: Management Through Standards," *CrossTalk - The Journal of Defense Software Engineering*, may 2005.

[12] L. E. Aguirre and A. Benavides, "A Common Model for Vulnerability Advisories," Master's thesis, Stockholm University/Royal Institute of Technology, 2005.

[13] G. S. Bjaen and E. Kaasin, "Security in GPRS: Master Thesis in Information and Communication Technology," Master's thesis, Adger University College, 2001. [Online]. Available

WWW: http://student.grm.hia.no/master/ikt01/ikt6400/ekaasin/Master%20Thesis%20Web.pdf (Accessed 31 May 2006).

[14] Mitre Corporation, "Official Oval Schemas," 2005. Available:http://oval.mitre.org/oval/download/schema/index.html. Last accessed 24/05/2006.

[15] W3C, "Soap Version 1.2 Part 1: Messaging Framework," tech. rep., 2003. Available:http://www.w3.org/TR/soap12-part1/. Last accessed 24/05/2006.

[16] OASIS Open, "Web Services Security:SOAP Message Security 1.1(WS-Security 2004)," tech. rep., OASIS Open, 2006. Available:http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf. Last accessed 24/05/2006.

[17] XML Encryption Working Group, "XML Encryption WG," 2005. [Online].Available WWW :http://www.w3.org/Encryption/2001 (Accessed 31 May 2006).

[18] XML Signature Working Group, "XML Signature WG," 2005. [Online]. Available WWW :http://www.w3.org/Signature (Accessed 31 May 2006).

[19] OASIS Open, "Web Services Security X.509 Certificate Token Profile 1.1," tech. rep., OASIS Open, 2006. Available:http://www.oasis-open.org/committees/download.php/16785/wss-v1.1-spec-os-x509TokenProfile.pdf. Last accessed 24/05/2006.

ACKNOWLEDGMENTS

This work was undertaken in the Distributed Multimedia Centre of Excellence at Rhodes University, with financial support from Telkom SA, Business Connexion, Comverse, Verso Technologies, Tellabs, StorTech, THRIP and the Andrew Mellon Foundation.