A less attack-prone, Internet deployment of iLanga

Courage Samuel Radu
Rhodes University
Computer Science Department
Grahamstown, 6140
+2774 561 9424
g11r3764@campus.ru.ac.za

Mosioua Tsietsi
Rhodes University
Computer Science Department
Grahamstown, 6140
+2746 603 8787
m.tisetsi@ru.ac.za

Alfredo Terzoli
Rhodes University
Computer Science Department
Grahamstown, 6140
+2746 603 8602
a.terzoli@ru.ac.za

ABSTRACT

Communication has become very important in the 21st century in a sense that people require communication services to be always available whenever they need them. As such, telephony has developed into a ubiquitous service. iLanga is a Voice over IP (VoIP) telephony system built using open source software with Asterisk Private Branch Exchange (PBX), Kamailio proxy server and MySQL database as main components. It provides its services over the Internet where there is a mixture of both legitimate users and potential attackers. This makes the system prone to security attacks. This paper discusses the attacks that were identified on iLanga system and the ways to mitigate them. We also discuss a simple mechanism to quarantine attackers from legitimate users, allowing users to effectively access services without causing denial of service. The paper concludes with a description of a secure version of iLanga that was implemented after considering the previous attacks and envisioned threats.

ACM Computing Classification System Classification

Thesis classification under the ACM Computing Classification System (1998 version, valid through 2011):"
D.4.6 [Security and Protection]: Authentication
K.6.5 [Security and Protection]: Unauthorized access (e.g., hacking, phreaking)

General Terms

Security

Keywords

VoIP, Asterisk, Security, iLanga

1. INTRODUCTION

Researchers at Rhodes University developed a VoIP telephony system named iLanga [6]. This system is essentially a complete, cost effective, computer-based Private Branch Exchange (PBX) that enables registered users, both on campus and on the open Internet, to communicate. As a VoIP system, iLanga has a rich set of features that includes interactive voice response, call conferencing, music on hold, support for multiple devices for a single user and many more.

As an open source system designed to facilitate service delivery over the Internet, iLanga is subject to a mixture of both legitimate users and attackers. This makes security a very important issue to consider. We will carefully look into how security can be considered and be prioritised, in order to deploy a reliable communication system.

2. BACKGROUND

In the past, iLanga has been compromised and at the beginning of this investigation it was shut down to avoid further exploitation rendering it unavailable to its users. The problem was be largley attributed to malicious elements on the Internet. The most prevalent threats to VoIP deployments today are the same security threats that exist in traditional data networks [1,4]. Among these threats are brute force attacks, Denial of Service (DoS) and Distributed Denial of Service (DDoS). Brute force attack is a technique whereby an attacker simply guesses username and password until he finds a combination that works. DoS and DDoS attack are characterised by an attempt to prevent the legitimate use of a service. Additionally, in the telephony world, there is a new threat known as Spam over Internet Telephony (SPIT) which refers to unsolicited bulk calls or instant messages [10].

3. iLANGA SYSTEM

Figure 1 shows a high level overview architecture of iLanga. This is a generic overview of what the system looks like. The server is the main hardware component and has Asterisk installed on it. Asterisk is software that turns an ordinary computer into a communications server [2]. It receives requests from users and processes them accordingly. Asterisk server is connected to the public Internet via campus network. Moreover, the server is connected to Telkom (the South African fixed line operator) via Integrated Service Digital Network (ISDN) connection. The server acts as a gateway between the Public Switched Telephone Network (PSTN) and the IP based network. Two more servers are also installed and these are Kamailio proxy server and MySQL database server.

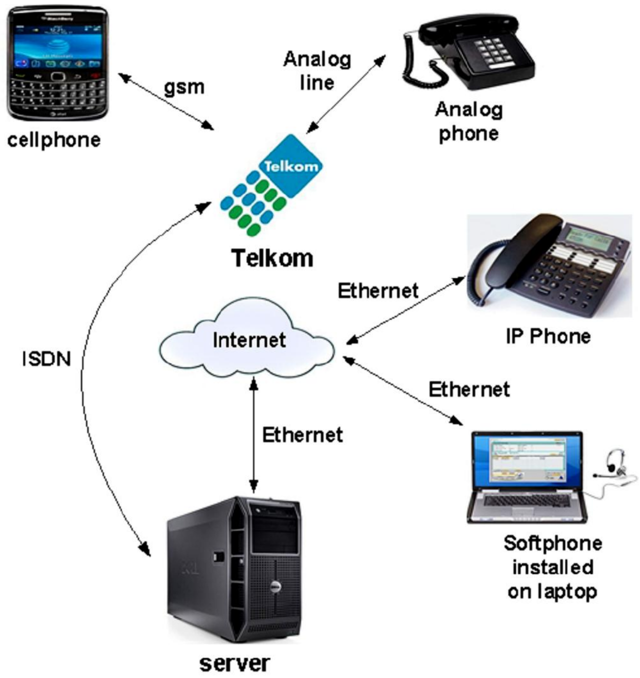


Figure 1: iLanga System overview

Kamilio proxy server is used for authentication and MySQL server is used to store user information.

4. SECURITY ANALYSIS AND IMPLEMENTION

4.1 A Secure Asterisk Installation

4.1.1 Run Asterisk as a non-root

When installing Asterisk, it is a good practice to run it as a non-root [7,13]. By default, Asterisk is configured to run as root, that is, as a super user. This is useful so that if the Asterisk is compromised, it cannot be used to take over the entire machine.

It is possible to run Asterisk with reduced privileges by adding a new user account on the system. Like any account, a username and password will be provided. When Asterisk is reconfigured to run as a normal user, it will then need permissions to be able to read and write certain files. Some files need to be writable so that Asterisk will be able to append lines for instance registration status, while other files have to be readable for instance the users.conf.

4.1.2 Changing default port

Port 5060 is the standard port used for SIP signaling. Changing the default port to another arbitrary port will add a smaller layer of protection. This is because the default port is well known and that can make the attacker's work easy. Changing the default port will thus add an extra layer of security and will force the attacker to work harder in order to determine the properties of the system. This intervention however will require users to alter the server settings on their phones.

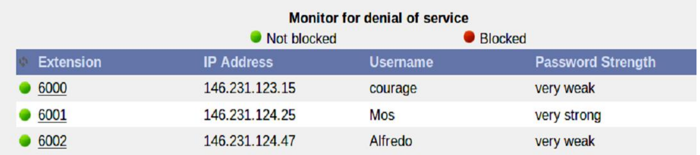
4.2 Public Key Authentication for SSH login

Secure Shell (SSH) is a program that allows a secure access to a remote machine. It allows secure data communication and command execution [8]. SSH uses public key authentication if necessary. Public key authentication is highly recommended since the server is visible over the Internet [9]. The goal is to prevent random username and password guesses before the attacker finds the correct combination. When SSH is enabled one can securely login remotely without having to provide a password. With public key authentication, one has to generate the private and public keys. The public key is then transferred to the server. The private key will remain on the client side. When logging in, the private key is used instead of providing the password. If the private key matches with the public key, after a mathematical computation, then it gets authenticated. The length of the private key is relatively large and thus makes it hard to use brute force methods on it.

4.3 Securing User Passwords

In order to make phone calls, one has to be a registered user. Registered users are uniquely identified by a username and a password. Asterisk is not programmed to detect weak passwords, thus users end up using or creating weak passwords. This weakness is within the system because there are no guidelines for creating strong passwords that users can follow. Additionally, when the administrator creates a new user account with a weak password, the system does not complain.

To mitigate against this, a perl script is used to check the strength of passwords for users and alerts the administrator of any weak passwords via the browser. Figure 2 shows the password strength for three registered users. This allows the administrator to always be alert of any weak passwords within the system.



Monitor for denial of service

🟢 Not blocked 🔴 Blocked

	Extension	IP Address	Username	Password Strength
🟢	6000	146.231.123.15	courage	very weak
🟢	6001	146.231.124.25	Mos	very strong
🟢	6002	146.231.124.47	Alfredo	very weak

Figure 2: Monitoring password strength

The perl script looks for the following features in a password:

1) Length (at least 12 characters)

2) Lowercase characters (at least two)

3) Uppercase characters (at least two)

4) Digits 0-9 (at least two)

5) Special character eg #,$,^, @, and & (at least two)

The system should always have strong passwords that are hard to guess. The features above enhance the strength of passwords [5].

4.4 Securing User Accounts

Since the Asterisk PBX is pointing to the public Internet, it is most likely to be scanned for valid user accounts. The intruder typically, checks for common usernames and then goes for numbered accounts, since it is common for administrators to name SIP accounts with the same name as the extensions on the

PBX. When enumerating usernames, the attacker tries to register a phone using different usernames. A server response will enable the attacker to determine whether the username exists or not.

Setting the variable alwaysauthreject=yes in the sip.conf file will resolve this problem. This will prevent the attacker from enumerating the usernames on the server. Additionally, the use of non-numeric usernames for VoIP accounts will make them harder to guess [7].

4.5 Dialplan Security

The Asterisk dialplan is another area where security should be considered important. In Asterisk, a dialplan is the most important part of the Asterisk system [7]. It defines how Asterisk handles incoming and outgoing calls. The configuration file extensions.conf contains the dialplan with all valid extension numbers. A new installation of Asterisk has the extensions.conf file installed by default. Modifying this default file by adding new extensions must be done carefully because some of the dialplan syntax have security risks. One can use the default extensions.conf file as a reference. A good approach is to create a new file and populate it with contents.

4.5.1 Default context

There is a context known as the default context. The default context should be secure. It should not have extensions that can cost the organisation money.

4.5.2 Dialplan injection

A dialplan should be built with great care in order to prevent one of the more recent dialplan vulnerabilities that have been discovered [3,7]. The channel variable `${EXTEN}` is commonly used in the dialplan. If this variable is used with wildcard pattern matches,it can lead to possible string injection vulnerability. The following example shows the use of a wildcard match in a dialplan.

```
exten => X.,1,Dial(SIP/${EXTEN})
```

It may be possible for an attacker to craft an INVITE which sends data such as

```
300&DAHDI/g1/4165551212
```

which would create an outgoing channel leg that was not originally intended. If evaluated the extension will become

```
exten=>X.,1,Dial(SIP/300&DAHDI/g1/4165551212)
```

If the system has an interface to the PSTN installed and configured, this will cause the call to go out on the number chosen by the attacker, even though the administrator did not grant access to that caller. This will probably cost the organisation.

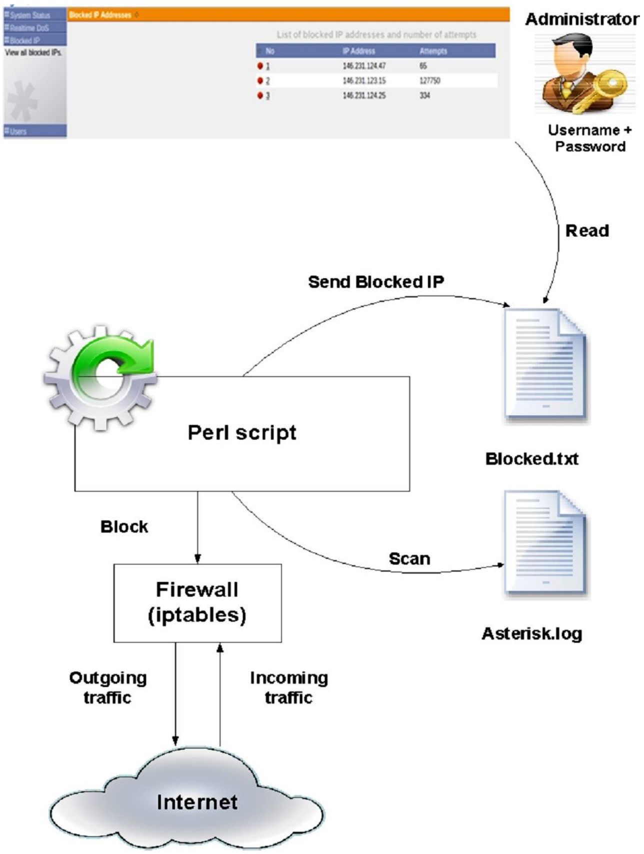
Administrators or developers should be careful on how foreign data will flow in the system when designing the dialplan. This problem can be solved by filtering string data from external sources. This can be done using the FILTER() dialplan function. All incoming context data should be filtered before it starts to flow in the system. Strict pattern matching can be used if the length of the extension is known before hand, for example

```
exten => _XXXX,1,Dial(SIP/${EXTEN})
```

4.6 Monitoring suspicious events

It is highly recommended that an intrusion prevention mechanism be used in order to protect VoIP systems like iLanga. Fail2ban [7] can be used as a basic countermeasure to quarantine offending IP addresses. Fail2ban bans an IP that makes too many failed login attempts. It then updates the firewall to reject that particular IP address.

It is useful to alert the system administrator about the status of blocked IP addresses. Fail2ban has a useful functionality that sends a mail everytime it bans an IP address. Nevertheless, this functionality requires a Mail Transfer Agent (MTA) like Postfix [7] to be installed in order to work. While it is beneficial to receive a blocked IP address this has some weaknesses. Firstly, the administrator might receive thousands of blocked IP addresses in a day which does not give extra information like the number of attempts made by each particular IP address. Secondly, there is no way the administrator can tell if the blocked IP is being used by a legitimate user. Thirdly, the option `ignoreip =` , where one can specify IP addresses that should never be blocked especially the server IP can be a risk if the attacker spoofs that server IP address.



List of blocked IP addresses and number of attempts

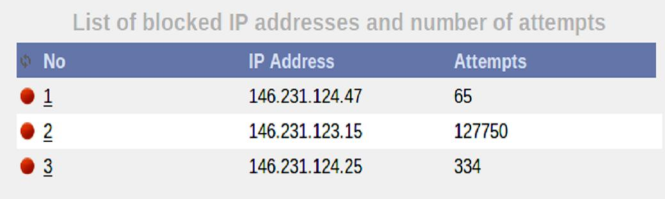
No	IP Address	Attempts
● 1	146.231.124.47	65
● 2	146.231.123.15	127750
● 3	146.231.124.25	334

Figure 3: Quarantining offending IP address

4.6.1 Banning offending IP addresses

We developed a perl script that behaves the same way as Fail2ban in order to suite our domain problem. This script is an extension of a perl script originally created by an Asterisk consulting company Teamforest [12]. Asterisk PBX comes with an inbuilt mini web server that is enabled so that the administrator will be able to view the blocked IP addresses via the browser. An administrator will have easy access since the browser is accessible everywhere. Figure 3 shows the perl script running in the background, blocking IP addresses that exceeds six attempts with wrong passwords.

An administrator can view this information anywhere via the browser. Figure 4 shows the web interface with three blocked IP addresses and the number of attempts made.



List of blocked IP addresses and number of attempts

No	IP Address	Attempts
● 1	146.231.124.47	65
● 2	146.231.123.15	127750
● 3	146.231.124.25	334

Figure 4: Summary of blocked IP addresses

4.7 The less attack-prone iLanga

Figure 5 shows the less attack-prone iLanga architecture after all implementation.

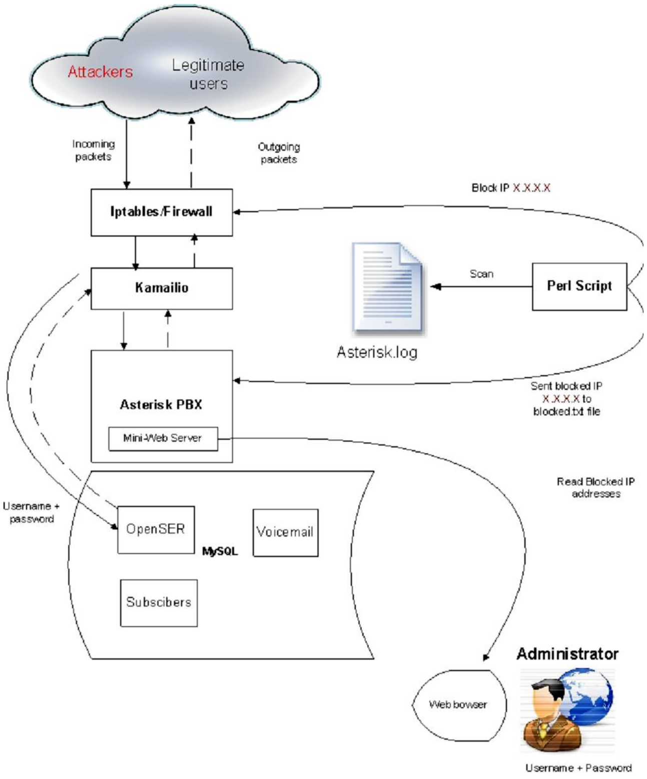


Figure 5: The less attack iLanga

The secure implementation shows the use of the inbuilt Linux firewall (iptables) working with the perl script blocking offending IP addresses, the Asterisk PBX server running as a non-root, variable `alwaysauthreject=yes`, the administrator can SSH login using private key, all user passwords are strong, a carefully designed diaplan and the administrator can monitor security related information via the browser.

5. EXPERIMENTS AND DISCUSSION

5.1 Enumerating usernames on Asterisk server

An experiment was carried out to demonstrate the importance of setting the variable alwaysauthreject=yes. It was a three step procees 1) host identification 2) enumerating valid usernames, and 3) cracking a password for a valid user. SIPvicious suite was used to do the three steps [11]. The results showed that when the variable `alwaysauthreject=no`, it is possible to enumerate valid users on Asterisk server. When the variable `alwaysauthreject=yes`, the server will respond with an error message.

5.2 Enumerating usernames on Kamailio server

Kamailio server responses will help an attacker to enumerate all valid extensions on the server [4]. Server response ñ401-Unauthorisedò, ñ403-Forbbidenò and ñ404-Not Foundò will tell if the username exists or not. If both username and password are wrong the server response is ñ401-Unauthorisedò. If the peer exist and has wrong password the response will be ñ403-Forbbidenò. If the peer does not exist the response will be ñ404-Not Foundò. If the attacker repeatedly sent requests, he can build a list of all valid usernames. Brute force methods can then be used to crack passwords for the valid usernames.

6. CONCLUSION

We managed to achieve our goal in securing the iLanga system. However, it is important to stress that security is an on-going process and as new threats emerge, they must be mitigated. Over the course of the investigation, we identified threats as well as vulnerabilities within the system. Among the threats was to brute force attacks, on both sip user accounts and the server root account. Vulnerabilities were also noted which include dialplan injection, ability to enumerate user accounts on Kamailio proxy server. Some weaknesses are within the Asterisk server itself, for instance, it allows users to create weak passwords. Toll fraud is another problem that we noted on the system whereby unauthourised users were making international calls.

There are several countermeasures we implemented on the system to make iLanga less attack-prone. We used the inbuilt Linux firewall (iptables) to filter incoming traffic. The firewall was used together with the perl script that we developed to quarantine offending IP addresses. To prevent denial of service to legitimate users, we developed a web UI that will alert the administrator when they got blocked. During the initial installation, we installed the Asterisk server as a non-root to prevent further compromise if the machine is hacked. There were basic configurations that we did, for instance, setting the variable `alwaysauthreject=yes` and changing the default port 5060 to another arbitrary port. In order to prevent toll fraud, we created a secure dialplan with a secure default context and all incoming contexts being filtered to prevent dialplan injection. Administrators always need to remotely login to the server machine so we configured public key authentication and this would make brute force methods hard to perform on the root account.

We also made sure that all default passwords were changed especially on Kamailio which comes with two default passwords `openserrw` and `openserro`. During the initial MySQL installation, we followed a guide provided by MySQL for securing the initial MySQL accounts. We immediately changed all default passwords in the grant tables and making sure all accounts have passwords. Finally, we made an update on all servers so that they use the latest server versions.

7. REFERENCES

[1] BLAKE, E. A. Network security: VoIP Security on data networkï A guide. In *Proceedings of the 4th annual conference on Information security curriculum development* (New York, NY, USA, 2007), InfoSecCD ô07, ACM, pp. 27:1ï 27:7.

[2] DIGIUM. Asterisk. [Online at http://www.asterisk.org/] last visited October 2011.

[3] DIGIUM. Security Advisories. [Online at http://www.asterisk.org/security] last visited October 2011.

[4] ENDLER, D., AND COLLIER, M. *Hack Exposed VoIP: Voice Over IP Security Secrets & Solutions*. McGraw-Hill, 2007.

[5] GARRISON, C. P. Encouraging good passwords. 1ï 4.

[6] J. PENTON, A. T. iLanga: A Next Generation VoIP-based, TDM-enabled PBX.

[7] MADSEN, L., MEGGELEN, J. V., AND BRYANT, R. *Asterisk : The Definitive Guide*. OôREILLY, 2011.

[8] OPENSSH. Security. http://www.openssh.com/.

[9] PLANELLA, D. SSHOpensshkeys. [Online at https://help.ubuntu.com/community/SSH/OpenSSH/Keys] last edited March 2011, last visited June 2011.

[10] S DRITSAS, J MALLIOS, M. T. G. F. M. D. G. Threat analysis of the session initiation protocol regarding spam, 2007.

[11] SECTECHNO. Hacking Exposed VoIP/SIP. [Online at http://www.sectechno.com/2011/05/23/hacking-exposed-voipsip/] last visited October 2011.

[12] TEAMFOREST. Automatically block failed SIP peer registrations. [Online at http://www.teamforrest.com/blog/171/asterisk-no-matching-peer-found-block/] last visited July 2011.

[13] VOIP-INFO. Asterisk slimming. [Online at http://www.voip-info.org/wiki/view/Asterisk+Slimming] last visited June 2011.