Improving SIP authentication

Lars Strand
Norwegian Computing Center / University of Oslo
Oslo, Norway
Email: lars.strand@nr.no

Wolfgang Leister
Norwegian Computing Center
Oslo, Norway
Email: wolfgang.leister@nr.no

***Abstract*—The digest access authentication method used in the voice over IP signaling protocol, SIP, is weak. This authentication method is the only method with mandatory support and widespread adoption in the industry. At the same time, this authentication method is vulnerable to a serious real-world attack. This poses a threat to VoIP industry installations and solutions. In this paper, we propose a solution that counters attacks on this wide-spread authentication method.**

***Index Terms*—SIP, authentication, Digest Access Authentication, security attack.**

I. Introduction

The most common protocol pair used for sending Voice over IP (VoIP) is the Session Initiation Protocol (SIP) [1] and Real-time Transport Protocol (RTP) [2]. RTP transfers the media content, while SIP handles the signaling, i.e., set up, modification and termination of sessions between two or more participants. VoIP is the emerging technology that will eventually take over from the traditional Public Switched Telephone Network (PSTN) [3] due to VoIP's improved flexibility and functionality, such as improved sound quality ("HD sound") using wideband codecs like G.722 [4], instant messaging (IM), presence, mobility support, and secure calls. VoIP reduces maintenance and administration costs since it brings convergence to voice, video and data traffic over the IP infrastructure.

SIP is an application layer protocol developed by the IETF. Its core functionality is specified in RFC3261 [1]. Additional functionality is specified in additional RFCs [5]. SIP sessions range from ordinary calls between two participants to advanced conference sessions between multiple participants communicating over video, voice, and IM.

However, SIP and RTP-based VoIP installations are rather difficult to secure [6]. VoIP inherits many security threats and Quality of Service (QoS) properties from the Internet, in addition to threats that come from the VoIP-specific technologies [7]. A clear and concise VoIP threat taxonomy is given by VOIPSA [8]. There are many obstacles in securing SIP, due to its use of intermediaries and the fact that functionality was the primary focus for the SIP designers, not security [1, page 232].

SIP supports several security services, and the RFC recommends their use. These security services can provide protection for authentication, confidentiality, and more. Yet, only one such security service is mandatory: the SIP Digest Access Authentication (DAA) method [1, page 193]. In our experience the other security services are neither implemented nor used. The only security service used is the mandatory authentication method.

DAA is primarily based on the HTTP Digest Access Authentication [9], and is considered to be weak and vulnerable to serious real-world attacks [10].

The main contribution of this paper is to present and analyze the seriousness of a vulnerability we presented in our earlier work – the registration attack [10]. We propose a solution to secure DAA that will counter this vulnerability.

The rest of the paper is organized as follows: We show our approach in Section II. We explain SIP authentication in Section III, and show the registration attack previously discovered in Section IV. In Section V, we show how to improve the authentication method to counter this attack. Related work is given in Section VI, before concluding in Section VII.

II. Method and case study

In Norway, both private companies and public authorities are migrating from PSTN to VoIP [11]. Our case study is taken from three companies in Norway; one medium sized company with 150 employees, and two larger companies with 3000 and 4700 employees. We have gathered several of these VoIP configurations and setups, and replicated the installations in our test lab [12]. In these companies, most of the employees have their own VoIP phone, called a User Agent (UA). All VoIP servers run the Linux operating system with the open source telephony platform *Asterisk* [13]. We found in these configurations that the digest authentication is the only authentication method for the UAs.

Our analysis follows the workflow shown in Fig. 1. In the following paragraphs, the numbers in parentheses refer to the numbers in Fig. 1.

In order to gain knowledge of the SIP protocol we use the specification documents (1), here the SIP standard. Then, we analyze VoIP network traffic going through the test lab (5). We have implemented two VoIP setups based on configurations from our industry partners ((2) and (3)). The network traffic is intercepted and saved to file using the network tool *tcpdump* (4). The network traffic is then analyzed off-line using the packet analyzer, *Wireshark* (5). An example of such an analysis is shown in Fig. 2.

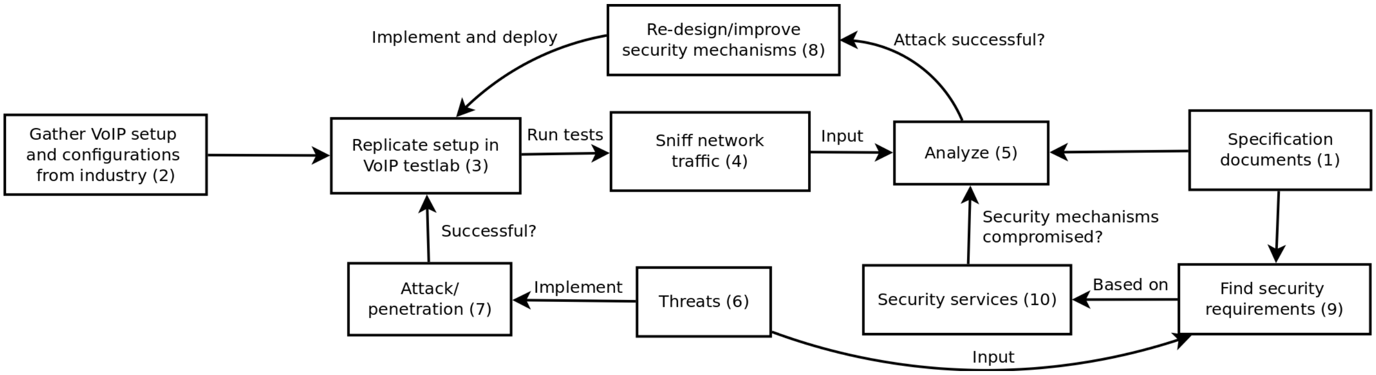
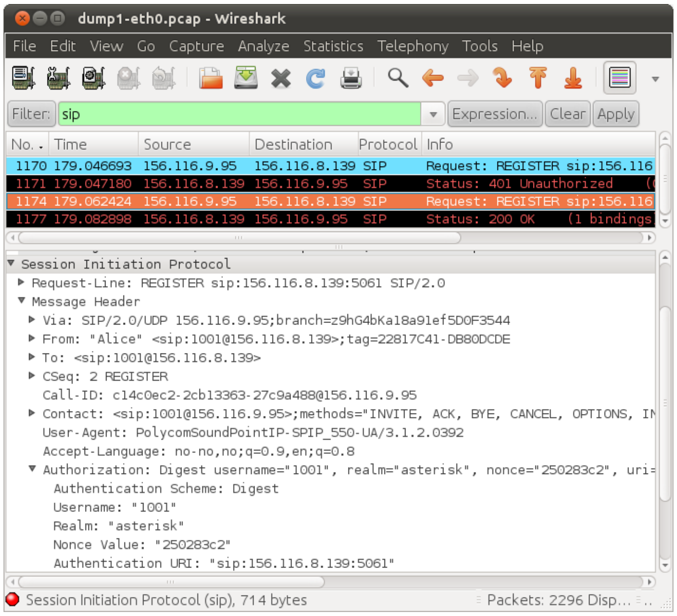

Implement and deploy
Re-design/improve security mechanisms (8)
Attack successful?
Gather VoIP setup and configurations from industry (2)
Replicate setup in VoIP testlab (3)
Run tests
Sniff network traffic (4)
Input
Analyze (5)
Specification documents (1)
Successful?
Security mechanisms compromised?
Attack/ penetration (7)
Implement
Threats (6)
Security services (10)
Based on
Find security requirements (9)
Input

Fig. 1: Workflow for analysis of the SIP authentication method.


dump1-eth0.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: sip
Expression... Clear Apply

```
No. .  Time        Source         Destination    Protocol Info
1170 179.046693 156.116.9.95   156.116.8.139 SIP      Request: REGISTER sip:156.116
1171 179.047180 156.116.8.139  156.116.9.95  SIP      Status: 401 Unauthorized    (
1174 179.062424 156.116.9.95   156.116.8.139 SIP      Request: REGISTER sip:156.116
1177 179.082898 156.116.8.139  156.116.9.95  SIP      Status: 200 OK      (1 bindings
```

```
▼ Session Initiation Protocol
  ▶ Request-Line: REGISTER sip:156.116.8.139:5061 SIP/2.0
  ▼ Message Header
    ▶ Via: SIP/2.0/UDP 156.116.9.95;branch=z9hG4bKa18a91ef5D0F3544
    ▶ From: "Alice" <sip:1001@156.116.8.139>;tag=22817C41-DB80DCDE
    ▶ To: <sip:1001@156.116.8.139>
    ▶ CSeq: 2 REGISTER
      Call-ID: c14c0ec2-2cb13363-27c9a488@156.116.9.95
    ▶ Contact: <sip:1001@156.116.9.95>;methods="INVITE, ACK, BYE, CANCEL, OPTIONS, IN
      User-Agent: PolycomSoundPointIP-SPIP_550-UA/3.1.2.0392
      Accept-Language: no-no,no;q=0.9,en;q=0.8
    ▼ Authorization: Digest username="1001", realm="asterisk", nonce="250283c2", uri=
        Authentication Scheme: Digest
        Username: "1001"
        Realm: "asterisk"
        Nonce Value: "250283c2"
        Authentication URI: "sip:156.116.8.139:5061"
```

```
Session Initiation Protocol (sip), 714 bytes          Packets: 2296 Disp...
```

Fig. 2: Network analysis using the network tool Wireshark.

As an additional input we consider threats deducted from formal analysis of the protocol, such as a SIP attack analyzed by Hagalisletto and Strand [10], using the protocol analyzer PROSA (6). We explain the attack in more detail in Section IV, and implement and execute the attack using the network tool *NetSED* (7) as shown in Fig. 6. Based on the security requirements (9) obtained from the SIP specification, we then checked if the authentication method (10) was compromised by the real-world attack. After careful analysis of the SIP headers we found that the SIP registration attack could be countered by a modification of the SIP authentication method (8).

III. Authentication in SIP

Authentication is the assurance that a communicating entity is the one that it claims to be [14]. Authentication consists of two basic steps: *a*) *Identification*, where an entity/client presents a value to the authentication system, and *b*) *Verification* where this value is validated against the authentication system [15]. When people that know each other are dialing or answering a phone call, they can often authenticate the other by just recognizing the other person's voice. However, when using new communications channels, such as instant messaging (IM), video, screencast and presence, determining the authenticity of the communicating partner is more difficult than for a voice call. To have established the identity of the caller is also important when, for instance, a physician need to communicate with a patient and discuss sensitive health information. For instance, someone else could masquerade as the patient and illegally obtain sensitive health information on the patient.

The SIP Digest Access Authentication (DAA) is currently the most common authentication scheme for SIP. Other authentication schemes have emerged, but DAA is the only mandatory authentication scheme [1, Section 22]. DAA uses a challenge-response pattern, and relies on a shared secret between client and server.

SIP is heavily influenced by the HTTP request-response model, where each transaction consists of a request that requires a particular response. The SIP messages are also similar in syntax and semantics to both HTTP and SMTP [16]. A SIP message consists of headers and a body. The SIP header fields are textual, always in the format `<header_name>: <header_value>`. The header value can contain one or more parameters. We show an example SIP header message in Fig. 4.

Any SIP request can be challenged for authentication. We show an example SIP DAA handshake in Fig. 3, and refer to the protocol clauses with a number in parentheses. The initial SIP `REGISTER` message (1) from Alice is not authorized and must be authenticated. The SIP server responds with a `401 Unauthorized` status message (3) which contains a `WWW-Authenticate` header with details of the challenge, including a *nonce* value. The client computes the required SIP digest that is embedded in (4) as an `Authorization` header. The SIP server, upon receiving the `Authorization` header, must perform the same digest operation, and compare the result. If the results are identical, the client is authenticated,

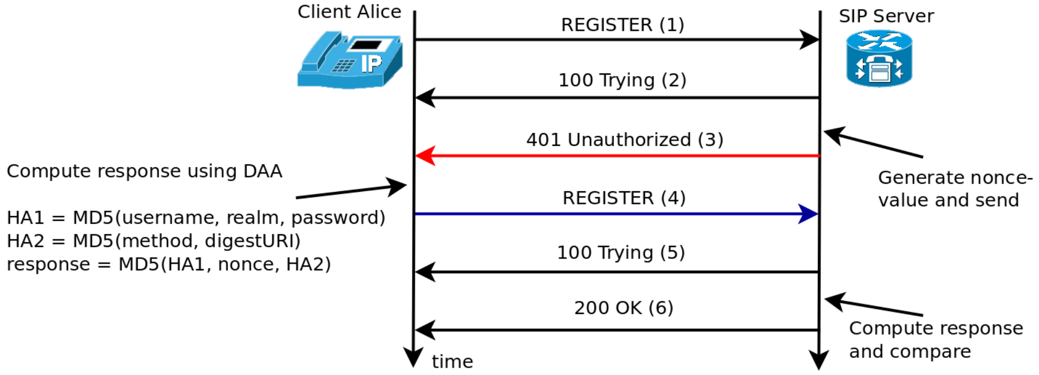
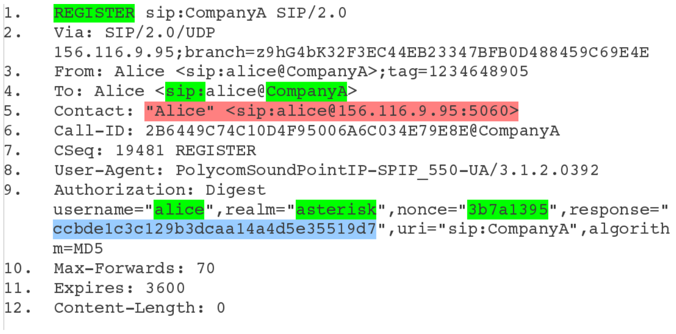

Client Alice
SIP Server
REGISTER (1)
100 Trying (2)
401 Unauthorized (3)
REGISTER (4)
100 Trying (5)
200 OK (6)
Compute response using DAA
HA1 = MD5(username, realm, password)
HA2 = MD5(method, digestURI)
response = MD5(HA1, nonce, HA2)
Generate nonce-value and send
Compute response and compare
time

Fig. 3: The SIP Digest Access Authentication method during a SIP REGISTER transaction.



```
REGISTER sip:CompanyA SIP/2.0
Via: SIP/2.0/UDP
156.116.9.95;branch=z9hG4bK32F3EC44EB23347BFB0D488459C69E4E
From: Alice <sip:alice@CompanyA>;tag=1234648905
To: Alice <sip:alice@CompanyA>
Contact: "Alice" <sip:alice@156.116.9.95:5060>
Call-ID: 2B6449C74C10D4F95006A6C034E79E8E@CompanyA
CSeq: 19481 REGISTER
User-Agent: PolycomSoundPointIP-SPIP_550-UA/3.1.2.0392
Authorization: Digest
username="alice",realm="asterisk",nonce="3b7a1395",response="
ccbde1c3c129b3dcaa14a4d5e35519d7",uri="sip:CompanyA",algorith
m=MD5
Max-Forwards: 70
Expires: 3600
Content-Length: 0
```

Fig. 4: The only attributes included in the digest response (blue) are depicted in green.

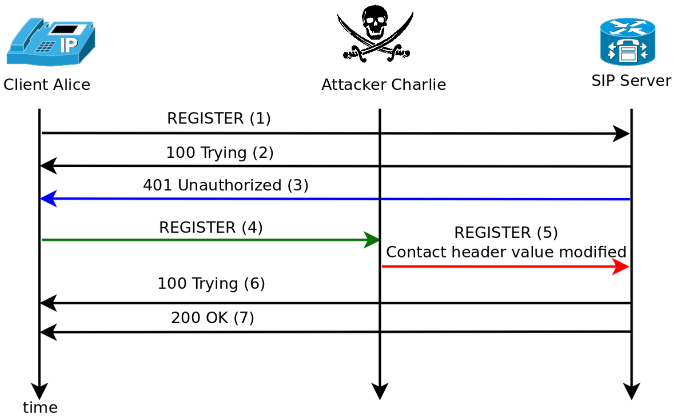


Fig. 5: The attacker Charlie can modify the Contact header value, and thereby have all Alice's calls redirected to him.

and a 200 OK message (6) is sent.

The SIP DAA is almost identical to the HTTP digest access authentication [9]. As we will show later, too few attributes are included in the digest computation, thus leaving some values unprotected. Formally, the DAA is expressed as follows:

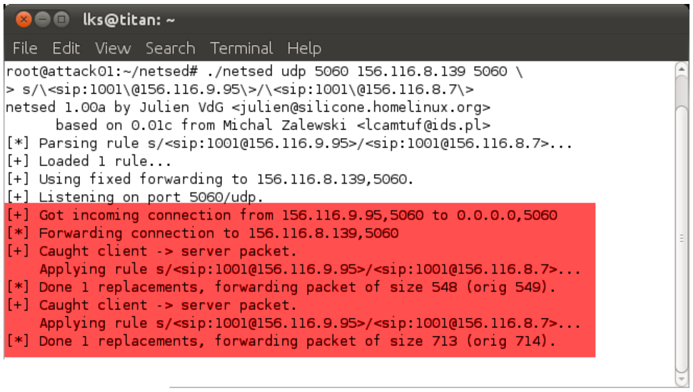
$$\begin{aligned} HA1 &= \mathrm{MD5}(A1) \\ &= \mathrm{MD5}(username : realm : password) \\ HA2 &= \mathrm{MD5}(A2) = \mathrm{MD5}(method : digestURI) \\ response &= \mathrm{MD5}(HA1 : nonce : HA2) \end{aligned}$$

In this context, $A1$ is the concatenated string of Alice's $username$, the $realm$ (usually a hostname or domain name) and the shared secret $password$ between Alice and the server. For $A2$, the $method$ is the SIP method used in the current transaction, in the above example that would be REGISTER. In a REGISTER transaction the $digestURI$ is set to the URI in the *To:*-field. The digest authentication $response$ is the hash of the concatenated values of $HA1$, the $nonce$ received from the server, and $HA2$. A SIP REGISTER message with a computed digest embedded in the Authorization header is shown in Fig. 4. DAA provides only reply protection due to the nonce value and one-way message authentication. There is no encryption of the content, nor confidentiality support, except the shared secret $password$ between client and server. All messages are sent in clear. DAA only works within a local domain so cross-domain authentication is not supported, which implies that end-to-end authentication is not supported. There is no provision in the DAA for the initial secure arrangement between a client and server to establish the shared secret. However, DAA has low computation overhead compared to other methods [17].

IV. Attack on DAA

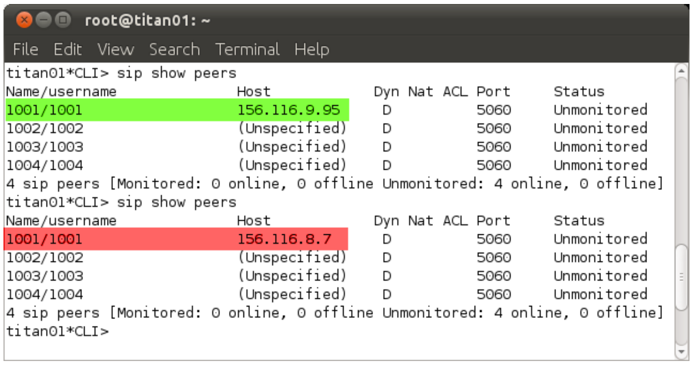
When a UA comes online it registers its contact point(s) to a *location service*. Contact points are the preferred methods a user can be contacted by, for example using SIP, mail, or IM. Usually, only a SIP URI contact method is present. The location service is responsible to redirect SIP requests (for VoIP calls) to the correct SIP end-point. For example, an incoming SIP call destined to alice@CompanyA.org does not contain information about which hostname or IP-address Alice's phone can be reached. Therefore, a SIP proxy will query the location service to receive Alice's phone's hostname or IP-address, and then redirect the call to this address.

The binding of Alice's phone to a hostname or IP-address is done during the REGISTER transaction, as depicted in Fig. 3. Before the binding, or registration, the SIP server should ask the client to authenticate itself, as explained in the



```
root@attack01:~/netsed# ./netsed udp 5060 156.116.8.139 5060 \
> s/\<sip:1001\@156.116.9.95\>/\<sip:1001\@156.116.8.7\>
netsed 1.00a by Julien VdG <julien@silicone.homelinux.org>
      based on 0.01c from Michal Zalewski <lcamtuf@ids.pl>
[*] Parsing rule s/<sip:1001@156.116.9.95>/<sip:1001@156.116.8.7>...
[+] Loaded 1 rule...
[+] Using fixed forwarding to 156.116.8.139,5060.
[+] Listening on port 5060/udp.
[+] Got incoming connection from 156.116.9.95,5060 to 0.0.0.0,5060
[*] Forwarding connection to 156.116.8.139,5060
[+] Caught client -> server packet.
    Applying rule s/<sip:1001@156.116.9.95>/<sip:1001@156.116.8.7>...
[*] Done 1 replacements, forwarding packet of size 548 (orig 549).
[+] Caught client -> server packet.
    Applying rule s/<sip:1001@156.116.9.95>/<sip:1001@156.116.8.7>...
[*] Done 1 replacements, forwarding packet of size 713 (orig 714).
```

Fig. 6: The network packet stream editor NetSED modifies network packets in real time based on a regular expression (in red).



```
titan01*CLI> sip show peers
Name/username              Host             Dyn Nat ACL Port     Status
1001/1001                  156.116.9.95      D          5060     Unmonitored
1002/1002                  (Unspecified)     D          5060     Unmonitored
1003/1003                  (Unspecified)     D          5060     Unmonitored
1004/1004                  (Unspecified)     D          5060     Unmonitored
4 sip peers [Monitored: 0 online, 0 offline Unmonitored: 4 online, 0 offline]
titan01*CLI> sip show peers
Name/username              Host             Dyn Nat ACL Port     Status
1001/1001                  156.116.8.7       D          5060     Unmonitored
1002/1002                  (Unspecified)     D          5060     Unmonitored
1003/1003                  (Unspecified)     D          5060     Unmonitored
1004/1004                  (Unspecified)     D          5060     Unmonitored
4 sip peers [Monitored: 0 online, 0 offline Unmonitored: 4 online, 0 offline]
titan01*CLI>
```

Fig. 7: Host name before (green) and after a successful attack (red), which makes Asterisk believe that Alice's phone (with number 1001) is reachable at an IP-address of the attacker's choice.

previous section. After a successful authentication, the client's hostname or IP-address is registered. A re-registration is normally done at regular intervals. This registration is repeated usually every 3-10 minutes, depending on the configuration. The client's preferred contact methods, including hostname or IP-address, is carried in the SIP header `Contact`, as depicted in Line 5 in Fig. 4. However, this SIP header value is sent in clear, and is not protected by DAA. Thus, the registration is vulnerable to a man-in-the-middle attack [10].

If an attacker modifies the hostname or IP-address in the *contactURI* header value during a `REGISTER` phrase, as depicted in Fig. 5, all requests, and hence calls, to the client will be diverted to a hostname or IP-address controlled by an attacker. Here, Alice cannot perceive that she is unreachable. An attacker can modify Alice's `REGISTER` session in real-time using NetSED [18] as depicted in Fig. 6. The SIP server (Asterisk), will not detect nor suspect that anything is wrong, and register Alice's phone number with the attackers IP address, as seen on Asterisk's terminal in Fig. 7. When Asterisk receives a call to Alice, the call will be forwarded to the attackers registered IP address.

V. Improving DAA

The SIP digest authentication is weak, which is stated in both the SIP specification [1], and the digest specification [9]. Specifically, DAA only offers protection of the value in the `To` header called the `Request-URI` and the *method*, but no other SIP header values are protected. Other better and stronger authentication methods have been recommended [19]. Nonetheless, we suggest improving the DAA as well as possible, since DAA is the authentication method commonly used due to its simplicity and widespread support and adoption.

A minor modification of DAA can counter the registration hijack attack [10], which is caused by having too few SIP header parameters protected by the digest. Since an attacker can modify and redirect all requests, we protect the header by including the `Contact` header value in the digest. By including the `Contact` value, which we name *contactURIs* in the digest, we effectively counter the registration hijack attack.

We define *HA0* with *contactURIs*. The new digest computation algorithm is as follows:

$$
\begin{aligned}
\mathit{HA0} &= \mathrm{MD5}(A0) = \mathrm{MD5}(\mathit{contactURIs}) \\
\mathit{HA1} &= \mathrm{MD5}(A1) \\
&= MD5(\mathit{username} : \mathit{realm} : \mathit{password}) \\
\mathit{HA2} &= \mathrm{MD5}(A2) = \mathrm{MD5}(\mathit{method} : \mathit{digestURI}) \\
\mathit{response} &= \mathrm{MD5}(\mathit{HA0} : \mathit{HA1} : \mathit{nonce} : \mathit{HA2})
\end{aligned}
$$

Weaknesses in the MD5 hash have been found. In particular we mention collision attacks where two different input values produce the same MD5 hash [20]. This weakness is not known to be exploitable to reveal a user's password [21]. Nonetheless, a stronger hash function, like SHA1 [22], is recommend.

We implemented and tested our modified DAA by using the Python Twisted [23] networking engine, using both MD5 and SHA1. According to our test, the computation overhead by including *HA0* with the *ContactURIs* is minimal, as shown in Fig. 8. The difference between the original DAA and our modified DAA with MD5 for 100.000 authentication requests on a 2.2Ghz Intel CPU, is only 0.44 seconds, a negligible amount.

A modified DAA means a modification of the SIP standard. Since the SIP standard has seen widespread industry adoption, it can be difficult to re-deploy a non-standardized SIP DAA. To prevent a modification of the SIP standard, we can use the DAA parameter `auth-param` to store our modified digest response. The parameter `auth-param` is reserved "for future use" [9, page 12], and can be a part of the `Authorization` header.

SIP devices that do not support the updated and more secure digest, can and will ignore this value, and use the original DAA for authentication. However, we cannot recommend this approach, since an attacker could remove this value and force the usage of the original standardized DAA. We would prefer to modify the DAA digest computation to force an upgrade to

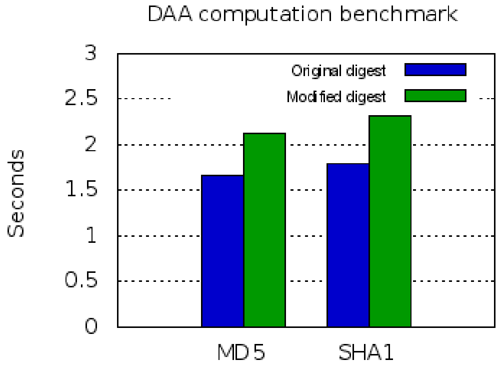


Fig. 8: The computation overhead for 100.000 iterations for original DAA and our modified DAA for both MD5 and SHA1.

the new improved DAA method, instead of compromising on security.

VI. Related work

Based on the DAA, Undrey [24] proposed a more flexible use of variables protected by the digest. His paper addresses the shortcomings of DAA and suggests to allow the server to decide which headers it requires to be included and protected by the digest computation. Unfortunately, his approach does not require specific headers fields to be included. Therefore, transactions that do not include `Contact` fields are still vulnerable to the registration attack.

Palmieri et al. [25], [26], dismiss DAA as a usable authentication method, and instead craft a new authentication schema with digital signatures based on public-key encryption. They rely on public key infrastructure (PKI), but admit that PKI is difficult and costly to implement.

Yang et al. [27] also conclude that DAA is weak. They argue that, since DAA is vulnerable to an off-line password guessing attacks, a more secure authentication method is required. They propose an authentication method based on Diffie-Hellman. Unfortunate, they do not discuss nor add any additional SIP headers in their new authentication scheme. So their solution is also vulnerable to the registration attack.

The H.323 recommendation for the VoIP protocol from the International Telecommunication Union (ITU) has failed to see widespread adoption by industry players, and is considered abandoned in favor of SIP/RTP [16]. The authentication methods in H.323, specified in H.235 [28], [29] uses well established security mechanism, like certificates, and Diffie-Hellman key exchange, to enforce authentication. Further analysis is needed to see whether the H.235 standard protects the signaling better than SIP.

The Inter-Asterisk eXchange (IAX) [30], also published by the IETF, establishes a competing protocol to SIP/RTP. IAX has several security properties that are better than SIP. By multiplexing channels over the same link and transporting both signaling and media over the same port, enforcing security mechanisms is easier. IAX supports two authentication methods: *1*) MD5 Message Digest authentication [31] computed over a pre-shared secret and a challenge (nonce), or *2*) using RSA public-key encryption on the challenge. In both methods, the nonce value is the only protocol parameter that is integrity protected by the authentication. Future work needs to investigate whether the IAX authentication method is adequately secure.

Other, more secure, authentication methods for SIP have been standardized, such as the support for public key encryption with S/MIME [32], the "Asserted Identity" extension [33], and the "Identity" header extension [34]. None of these authentication methods have seen any widespread deployment yet [19].

VII. Conclusion

We have seen that the widely deployed authentication method DAA in SIP is weak and vulnerable to attacks. Moreover, we have confirmed and verified that the attack analyzed earlier [10] can be performed on the SIP protocol in real-time. We have examined this authentication method, and proposed a solution to counter the serious registration attack. By including more SIP header parameters in the authentication digest this attack can be countered.

The original SIP designers focused on functionality and compliance at the cost of security. A more thorough investigation of the SIP DAA in the design phase would have revealed the vulnerability presented here, and the vulnerability could have been prevented early on.

Our remedy presented here solves an serious problem with the DAA. However, other weaknesses and shortcomings of DAA are too serious to be part of a strong and secure authentication scheme for SIP. Therefore, we intend to investigate other authentication methods for SIP, including support for Generic Security Service API (GSS-API) [35].

Acknowledgment

This research is funded by the EUX2010SEC project in the VERDIKT framework of the Norwegian Research Council (Norges Forskningsråd, project 180054). The authors would like to thank Trenton Schulz for discussions. We also thank Anders Moen Hagalisletto and the anonymous reviewers for comments on earlier drafts of this paper.

References

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261 (Proposed Standard), Internet Engineering Task Force, Jun. 2002, updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630, 5922, 5954, 6026. [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt [Accessed: 1. Nov 2011]

[2] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550 (Standard), Internet Engineering Task Force, Jul. 2003, updated by RFCs 5506, 5761. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt [Accessed: 1. Nov 2011]

[3] L. Strand and W. Leister, "A Survey of SIP Peering," in NATO ASI - Architects of secure Networks (ASIGE10), May 2010.

[4] International Telecommunication Union, "7 kHz Audio-Coding within 64 kbits/s," ITU-T Recommendation G.722, 1993.

[5] "IETF Session Initiation Protocol Core Charter." [Online]. Available: http://datatracker.ietf.org/wg/sipcore/charter/ [Accessed: 1. Nov 2011]

[6] D. York, Seven Deadliest Unified Communications Attacks. Syngress, Apr. 2010.

[7] H. Dwivedi, Hacking VoIP: Protocols, Attacks, and Countermeasures, 1st ed. No Starch Press, Mar. 2009.

[8] VoIPSA, "VoIP security and privacy threat taxonomy," Public Realease 1.0, Oct. 2005. [Online]. Available: http://voipsa.org/Activities/VOIPSA_Threat_Taxonomy_0.1.pdf [Accessed: 1. Nov 2011]

[9] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication," RFC 2617 (Draft Standard), Internet Engineering Task Force, Jun. 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2617.txt [Accessed: 1. Nov 2011]

[10] A. M. Hagalisletto and L. Strand, "Formal modeling of authentication in SIP registration," in Second International Conference on Emerging Security Information, Systems and Technologies SECURWARE '08. IEEE Computer Society, August 2008, pp. 16–21.

[11] L. Fritsch, A.-K. Groven, L. Strand, W. Leister, and A. M. Hagalisletto, "A Holistic Approach to Open Source VoIP Security: Results from the EUX2010SEC Project," International Journal on Advances in Security, no. 2&3, pp. 129–141, 2009.

[12] L. Strand, "VoIP lab as a research tool in the EUX2010SEC project," Norwegian Computing Center, Department of Applied Research in Information Technology, Tech. Rep. DART/08/10, April 2010.

[13] "Asterisk: The Open Source PBX & Telephony Platform." [Online]. Available: http://www.asterisk.org/ [Accessed: 1. Nov 2011]

[14] International Telecommunication Union (ITU), "Security Architecture For Open Systems Interconnection (OSI)," The International Telegraph and Telephone Consultative Comittee (CCITT), X.800 Standard, 1991.

[15] R. Shirey, "Internet Security Glossary, Version 2," RFC 4949 (Informational), Internet Engineering Task Force, Aug. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4949.txt [Accessed: 1. Nov 2011]

[16] H. Sinnreich and A. B. Johnston, Internet communications using SIP: Delivering VoIP and multimedia services with Session Initiation Protocol, 2nd ed. New York, NY, USA: John Wiley & Sons, Inc., August 2006.

[17] S. Salsano, L. Veltri, and D. Papalilo, "SIP security issues: The SIP authentication procedure and its processing load," Network, IEEE, vol. 16, pp. 38–44, 2002.

[18] "NetSED: The network packet stream editor." [Online]. Available: http://silicone.homelinux.org/projects/netsed/ [Accessed: 1. Nov 2011]

[19] D. Sisalem, J. Floroiu, J. Kuthan, U. Abend, and H. Schulzrinne, SIP Security. WileyBlackwell, Mar. 2009.

[20] X. Wang and H. Yu, "How to break MD5 and other hash functions," IN EUROCRYPT, vol. 3494, 2005.

[21] P. Hawkes, M. Paddon, and G. G. Rose, "Musings on the wang et al. md5 collision," Cryptology ePrint Archive, Report 2004/64, 2004.

[22] D. Eastlake 3rd and P. Jones, "US Secure Hash Algorithm 1 (SHA1)," RFC 3174 (Informational), Internet Engineering Task Force, Sep. 2001, updated by RFC 4634. [Online]. Available: http://www.ietf.org/rfc/rfc3174.txt [Accessed: 1. Nov 2011]

[23] "Twisted Matrix Labs." [Online]. Available: http://twistedmatrix.com [Accessed: 1. Nov 2011]

[24] J. Undery, "Ieft draft: SIP authentication: SIP digest access authentication," IETF, Tech. Rep., Jul. 2001.

[25] F. Palmieri, "Improving authentication in voice over IP infrastructures," in Advances in Computer, Information, and Systems Sciences, and Engineering, K. Elleithy, T. Sobh, A. Mahmood, M. Iskander, and M. Karim, Eds. Springer Netherlands, 2006, pp. 289 – 296. [Online]. Available: http://www.springerlink.com/content/pj11582775h177q0/ [Accessed: 1. Nov 2011]

[26] F. Palmieri and U. Fiore, "Providing true end-to-end security in converged voice over IP infrastructures," Computers & Security, vol. 28, no. 6, pp. 433–449, Sep. 2009.

[27] C. Yang, R. Wang, and W. Liu, "Secure authentication scheme for session initiation protocol," Computers & Security, vol. 24, no. 5, pp. 381–386, Aug. 2005.

[28] International Telecommunication Union, "H.323 security: Framework for security in H-series (H.323 and other H.245-based) multimedia systems," ITU-T Recommendation H.235.0, 2005.

[29] ——, "H.323 security: Framework for secure authentication in RAS using weak shared secrets," ITU-T Recommendation H.235.5, 2005.

[30] M. Spencer, B. Capouch, E. Guy, F. Miller, and K. Shumard, "IAX: Inter-Asterisk eXchange Version 2," RFC 5456 (Informational), Internet Engineering Task Force, Feb. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5456.txt [Accessed: 1. Nov 2011]

[31] R. Rivest, "The MD5 Message-Digest Algorithm," RFC 1321 (Informational), Internet Engineering Task Force, Apr. 1992. [Online]. Available: http://www.ietf.org/rfc/rfc1321.txt [Accessed: 1. Nov 2011]

[32] J. Peterson, "S/MIME Advanced Encryption Standard (AES) Requirement for the Session Initiation Protocol (SIP)," RFC 3853 (Proposed Standard), Internet Engineering Task Force, Jul. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3853.txt [Accessed: 1. Nov 2011]

[33] C. Jennings, J. Peterson, and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks," RFC 3325 (Informational), Internet Engineering Task Force, Nov. 2002, updated by RFC 5876. [Online]. Available: http://www.ietf.org/rfc/rfc3325.txt [Accessed: 1. Nov 2011]

[34] J. Peterson and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)," RFC 4474 (Proposed Standard), Internet Engineering Task Force, Aug. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4474.txt [Accessed: 1. Nov 2011]

[35] J. Linn, "Generic Security Service Application Program Interface Version 2, Update 1," RFC 2743 (Proposed Standard), Internet Engineering Task Force, Jan. 2000, updated by RFC 5554. [Online]. Available: http://www.ietf.org/rfc/rfc2743.txt [Accessed: 1. Nov 2011]