Universität Zürich[UZH]

VERTIEFUNGSARBEIT — Communication Systems Group, Prof. Dr. Burkhard Stiller

Security evaluation of the Accounting and Monitoring for AAI Services (AMAAIS) platform with the use of ethical hacking instruments

Samuel Liniger
Zürich, Switzerland
Student ID: 08-913-832

Supervisor: Christos Tsiaras, Andri Lareida
Date of Submission: April 15, 2013

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH—8050 Zürich, Switzerland

ifi

Vertiefungsarbeit
Communication Systems Group
Department of Informatics (IFI)
University of Zurich
Binzmühlestrasse 14, CH—8050 Zürich, Switzerland
URL: http://www.csg.uzh.ch/

Table of Content

Table of Figures

1 Summary

1.1 Einführung

Diese Vertiefungsarbeit befasst sich mit einer Sicherheits-Evaluation des AMAAIS- Projekts (Accounting and Monitoring for AAI Services).

1.2 Ziele

Das Ziel der Arbeit ist das Aufzeigen potentieller Sicherheitslücken und das Erarbeiten möglicher Gegenmassnahmen mithilfe von Ethical-Hacking-Instrumenten [2] .

Dies beinhaltete das Durchführen einer Black-Box-Attacke, bei der keine Informationen über AMAAIS verfügbar waren, sowie einer White-Box-Attacke [3], bei der mehr Informationen über das System bereitgestellt wurden.

1.3 Resultate

Der Black-Box-Test ergab, dass aufgrund der wenigen öffentlich verfügbaren Dokumente momentan keine signifikante Bedrohung besteht. Die White-Box-Tests ergaben, dass möglicherweise eine Gefahr von DoS-Attacken [4] besteht, da der AMAAIS-Server bei den Tests zum Teil nicht mehr verfügbar sein. Des Weiteren wurde ein möglicher Fragekatalog aufgestellt um ein weiteres strukturiertes Sicherheits-Assessment durchzuführen.

1.4 Aussicht

Zum Schluss der Arbeit wird beschrieben was in Zukunft für Tests durchgeführt werden können, um weitere Evaluationen über die Sicherheit des AMAAIS-Projekts durchzuführen.

2 Problem to be solved

The goal of this thesis was to perform black-box and white-box hacking techniques on an operational AMAAIS process in order to reveal system vulnerabilities. Thus a student in the role of an ethical hacker was engaged to test the system. First of all, ethical hacking techniques had to be explored. Thus, a variety of practitioner literature had to be studied [1][2][3][4][9][10][12][14][15][16][17]. After this adjustment to the job the ethical hacker had no information about the system in a first step. The goal of this step was to find out, if a potential malicious user, who has no insight to the server, could attack the system and crash it. For the evaluated weaknesses possible countermeasures were proposed. In a second step then, information about the architecture and help how to use the AMAAIS server were provided. The goal was to reveal potential weaknesses of the AMAAIS system, that users with an insight to the infrastructure and architecture might get to abuse.

2.1 Ethical Hacking

Nowadays, the term hacking is usually used in a negative sense. In the early beginning of the digital age the term hacker had two different meanings. A first definition said, that hacker is a person who enjoys learning the details of computer system. And the second definition stated, that a hacker is a person who programs enthusiastically or who enjoys programming rather than just theorizing about programming [15][14]. Today the term ethical hacker is usually used to distinguish hackers with a criminal intention of hackers that are officially allowed to attack a system, to reveal vulnerabilities. Palmer at [14] describes three main questions an ethical hacker wants to answer:

1 What can an intruder see on the target systems?

2 What can an intruder do with that information?

3 Does anyone at the target notice the intruder‘s attempts or successes?

2.1.1 Black Box Hacking

This approach assumes that the ethical hacker has approximately the same knowledge like a potential criminal hacker [9]. Hafele writes, that one has to differentiate between the different kind of hackers [9]. He categorizes the competencies into script kiddies or novices, technical astute hackers, sophisticated "Ueberhackers", and disgruntled insider attacks.

2.1.2 White Box Hacking

In contrary to the black box hacking, with the white box technique the hacking team has much more information about the system. One big advantage of this techniques is time and money [9]. Given this fact, the team can the focus on how to intrude a system instead of use a lot of time for an initial reconnaissance.

3 Discussion of the design choices

In this section the design choices of the black box and white box hacking approach is presented.

3.1 Black Box Tests

To start the black box testing phase in a structured manner, it was chosen to be done according to the five phases described by Hafele. This approach starts with the phase of initial reconnaissance. This includes the investigation, if there are any readily available public information about the target system. The second phase is called service determination or scanning phase. Hafele refers in this context to Namji [10] who writes that the activities of this phase are identification of listening services and ports that are operational on the system to evaluate. The third phase is the enumeration phase. The goal is to identify "open network services for possible exploit". The next phase is gaining access. This includes many different attack techniques such as password cracking and denial of service (DoS) attacks [17]. There are many possible forms of DoS attacks, but the goal is always to prevent or limit the availableness of a service by use limited or non-renewable resource, delete data or configurations or destroy computer- or communications facilities [1].The fifth phase is called privilege escalation. This stage assumes, that the hacker has gained access. Now the goal is to gain administrative or root level, to get complete control of the network.

3.2 White Box Tests

The structure of the white box hacking process was chosen to be similar to the black box testing phase. Since the initial reconnaissance was already done, the process should start with phase three immediately.

Additional to this technical tests also a social based security assessment was planned to take place. To do this part in a structured manner as well, the NIST Information Technology Security Assessment Framework should be used [12]. This is divided into five security levels. Level 1 states that a security policy has been documented. At level 2 also the procedures are documented and the asset has controls to implement the policy. If an asset has Level 3, it means that procedures and controls have been implemented. At Level 4 the procedures and controls are tested and reviewed. The last level, states that procedures and controls fully have been integrated into a program. NIST already provides a sample questionnaire to assess an organization. Due to the big coverage, only a selection of questions were considered in Section 4.2.

4 Solved Issues

This chapter illustrates how process of the black box and the white box approach took place. It highlights the revealed vulnerabilities and possible countermeasures.

4.1 Black Box Approach

For the initial reconnaissance as described in Section 3, the web was searched for information about the AMAAIS project. As result the official website `amaais.switch.ch` was found. Currently the documents that are public available are not helpful for a criminal hacker, because only the deliverables of phase one unto phase three are public accessible. Another reason why the documents are not very helpful yet, is because some architecture has already been changed, but the documented changes are untraceable for outsiders. Despite this, it is better to protect such documents, because it still contains sensitive data, since a potential criminal hacker could find helpful information about parties involved to the project. With this information the hacker could test the organization‘s staff wether they would provide sensitive information. Palmer describes this as social engineering [14]. As an example he mentions that an attacker could call an organizations computer help line and asking for help.

In AMAAIS deliverable D3 [6], one may come to the conclusion that an accounting database of AMAAIS is running on the server `amaais2.ethz.ch`. So the next step was to find out what services are running on this server. The first procedure was a portscan [8] on `amaais2.ethz.ch`. In deliverable three `amaais1.ethz.ch` was also mentioned. On this server a portscan was performed, too.The findings are documented in the following table:

Table 1: Portscan

Servername	IP	Open port	Possible service
amaais1.ethz.ch	129.132.65.31	22	ssh
amaais1.ethz.ch	129.132.65.31	13722	bpjava-msvc (BP Java MSVC Protocol
amaais1.ethz.ch	129.132.65.31	13724	vnetd (Veritas Network Utility)
amaais1.ethz.ch	129.132.65.31	13782	bpcd (VERITAS NetBackup)
amaais1.ethz.ch	129.132.65.31	13783	vopied (VOPIED Protocol)
amaais1.ethz.ch	129.132.65.31	53306	
amaais2.ethz.ch	129.132.65.32	22	ssh
amaais2.ethz.ch	129.132.65.32	80	http
amaais2.ethz.ch	129.132.65.32	8009	
amaais2.ethz.ch	129.132.65.32	13722	bpjava-msvvc
amaais2.ethz.ch	129.132.65.32	13724	vnetd
amaais2.ethz.ch	129.132.65.32	13782	bpcd
amaais2.ethz.ch	129.132.65.32	13783	vopied
amaais2.ethz.ch	129.132.65.32	53306	

According to AMAAIS deliverable D2 [5], the accounting server is running as java servlet web application in a java-webserver, like apache tomcat. To find out, if any such a service is running on the servers located before, a vulnerability test with a tool called nmap [13] took place. According to this test the server `amaais2.ethz.ch` is really an apache tomcat server. Although apache tomcat servers are quite common, the assumption that a service for AMAAIS might be running there could be done.

It turned out that no AMAAIS service is running on this server. That is the reason why the server has been tested locally. The enumeration phase, gaining access phase and finally the privilege escalation phase have been skipped. Instead of this, the next goal was just to bring the server in a denial of service state.

Thus, the next step was to let the server run locally on a MacBook Pro. The server worked, but with the gathered information no successful accounting process could have been tested. The reason was that with the provided information no running accounting client could be found. For this reason the server was tested with a tool called netcat [11]. The first goal was to establish a successful tcp connection. The next step was a test where only a single character was sent, but the server has rejected this, as expected Fig. 1.

```
t
HTTP/1.1 400 Bad Request
Connection: close
Server: Jetty(7.0.x)
```

Figure 1: Stream content

As second test a modified "syn flooding" [1] was performed, where a big amount of TCP SYNs are sent. According to Alexander [1] this includes only the first phase of the TCP-3-way-handshake. He also mentions that todays operating systems are adapted for this case. Thus this method was adapted and the whole 3-way-handshake performed. Such that only a tcp connection, was accomplished without sending anything afterwards Fig. 2.

```
public static void main(String[] args) {
 String ip = "127.0.0.1";
int port = 8080;
while (true) {
try {
Socket socket = new Socket(ip, port);
if (socket.isConnected()) {
System.out.printf("Connected to %s:%d\n", ip, port);
}

} catch (UnknownHostException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}
}
}
```

Figure 2: Syn Flooding java code

The local server was still working, after this test has been running five minutes. But meanwhile the server sometimes couldn‘t be reached with netcat anymore. The Java program shown in Fig. 2, threw the same exception as shown below in Fig. 7.

Except this, no other evidence for vulnerabilities with the gathered data was found. It might be possible, that the syn flooding blocked the server due to hardware limitations of the equipment that has been used to run the server. At the moment, where no public AMAAIS server is running, there is no significant threat of potential malicious users with no insight to the systems architecture.

4.2 White Box Approach

The first step of the white box testing included the setup of the updated AMAAIS server and an example client. Thereby it was possible to capture an example process Fig. 3, where a client is sending data to the server. The first goal was to re-inject such captured packets with original or modified data. Alexander calls this a replay-attack. He defines it as a

```
POST /amaais-acct-server HTTP/1.1
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 127.0.0.1:8080
Content-Length: 14465
Content-Type: text/xml; charset=UTF-8

<?xml  version="1.0"  encoding="UTF-8"?><acct:PublishEven-
tRequest       xmlns:acct="urn:mace:switch.ch:doc:account-
ing:profiles:1.0"       xmlns:xsd="http://www.w3.org/2001/
XMLSchema"   IssueInstant="-190184542-11-06T02:51:29.373Z"
Version="2.0"><ds:Signature   xmlns:ds="http://www.w3.org/
2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod   Algorithm="http://www.w3.org/
2001/10/xml-exc-c14n#"/>
<ds:SignatureMethod  Algorithm="http://www.w3.org/2000/09/
xmldsig#rsa-sha1"/>
```

Figure 3: Section of an event captured with

passive eavesdropping where the attacker is duplicating the message he has captured. This attack was chosen, because authentication protocols for applications like bank transactions are endangered.

In this case the captured data was re-injected with netcat. The AMAAIS server rejected this packet successfully.The result of this is shown in Fig. 4.

```
nc 127.0.0.1 8080 < original.txt
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
AMAAIS_STATUS: REJECTED
AMAAIS_REASON: Replayed
Content-Length: 0
Date: Tue, 02 Apr 2013 16:22:03 GMT
```

Figure 4: Re-injected packet

The first modification involved "ds:SignatureValue" of the element "ds:Signature" of the "acct:PublishEventRequest". The first character of the element value was replaced by another character. Afterwards this modified content was sent again with netcat. This time the server accepted it. The next modification included all "ds:SignatureValue"-elements.

Again, just the first characters of all these values were replaced. Now the server did not accept the packet with the reason "Untrusted Assertion" as it is shown in Fig. 5.

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
AMAAIS_STATUS: REJECTED
AMAAIS_REASON: Untrusted Assertion
Content-Length: 0
Date: Fri, 22 Mar 2013 13:07:31 GMT
```

Figure 5: Rejected packet

The same modification was done with the element "ds:DigestValue". First, only the first character of the element belonging to "ds:Signature" was replaced. Afterwards, all other "ds:DigestValue"-elements were modified. The results were the same as above. If only an element of "ds:Signature" was modified, the server accepted it. Otherwise it rejected the packet with the reason "Untrusted Assertion".

In a next step, the value of the first occurrence of "ds:SignatureValue" was changed and the "ds:SignatureValue" of the first "saml2:Assertion"-element. This led to the same answer of the server as above.

The next modification included the element "saml2:NameID". This value was renamed to "DefaultEvents" instead of "DefaultEvent". The resulted AMAAIS status was "ERROR" with the reason "Unable to parse input stream, it contained invalid XML".

In every test done so far, the connection to the client was closed by the server afterwards. But there was also a case where the server stayed connected. After the http request was sent with netcat Fig. 6, just random data was sent.

```
POST /amaais-acct-server HTTP/1.1
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 127.0.0.1:8080
Content-Length: 14465
Content-Type: text/xml; charset=UTF-8
```

Figure 6: HTTP Request

First, the server did not react. After a while it sent the response "AMAAIS_Statuts: ERROR" with the reason, that the input stream could not have been parsed. But the client remained connected.

In the white box testing phase also a syn flooding test (DoS attack) took place. After 300 connections have been established, the server was no more reachable. One possible

reason might be that the server ran locally and the syn flooding also started locally. But in contrary the CPU was not fully used.

```
300
java.net.ConnectException: Operation timed out
at java.net.PlainSocketImpl.socketConnect(Native Method)
at     java.net.AbstractPlainSocketImpl.doConnect(Abstract-
PlainSocketImpl.java:339)
at    java.net.AbstractPlainSocketImpl.connectToAddress(Ab-
stractPlainSocketImpl.java:200)
at  java.net.AbstractPlainSocketImpl.connect(AbstractPlain-
SocketImpl.java:182)
at          java.net.SocksSocketImpl.connect(SocksSocketIm-
pl.java:391)
at java.net.Socket.connect(Socket.java:579)
at java.net.Socket.connect(Socket.java:528)
at java.net.Socket.<init>(Socket.java:425)
at java.net.Socket.<init>(Socket.java:208)
at SynFlooding.main(SynFlooding.java:16)
```

Figure 7: Syn flooding exception

After ten seconds, the server was reachable again for 200 more syn‘s. Then it was blocked again.

As enhancement of the test, this syn flooding was also performed by an external workstation within the same wireless network Fig. 8.

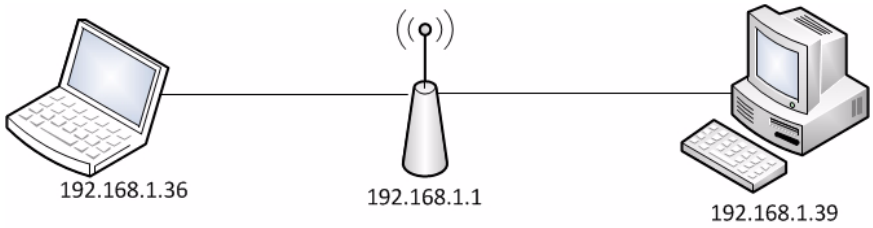

Figure 8: network topology

This test took place for five minutes. Meanwhile some connections attempts with telnet [8] were done, by the same workstation. The server was always reachable, but after the connection was established and a single character sent, it took some time (approximately five seconds) until the response of the server arrived. In a next step the syn flooding was started by the external workstation and the MacBook Pro with the running server. The first test took place for two minutes. The server reacted the same way as mentioned in the last test. This test was repeated for three minutes. After 445 connections the same exception as in Fig. 7 occurred on the workstation. Meanwhile a connection attempt with telnet was done by the workstation too. The server was not reachable anymore and the message shown in Fig. 9 resulted.

```
C:\Users\Kinsam>telnet 192.168.1.36 8080

Verbindungsaufbau   zu   192.168.1.36...Es   konnte   keine
Verbindung mit dem Host hergestellt werden, auf Port 8080:
Verbindungsfehler
```

Figure 9: Telnet connection error

A new connection attempt was taken immediately, but the same error occurred. But the server was not totally blocked. It recovered after some seconds and a new connection was established.

The next approach was also kind of a DoS attack. A more sophisticated program was written which reads an xml file containing the data to be sent. This xml file based on previously captured packets. This program modified the first occurrence of "ds:SignatureValue" as in the manual test mentioned above. This data was sent in a loop, like the syn flooding. During the test the server was in most cases still reachable by netcat. To be sure that the captured data was the data injected by netcat, the data sent with netcat was compared with the captured stream by a program called diffMerge [7]. The response time was not significant longer than in the case the server was not attacked. This is also apparent if the different wireshark [18] captures , which were captured on the server side, were compared Fig. 11. But there happened also the case, where the server sent the response "Internal Server Error" Fig. 10. By the next connection attempt the server had already recovered.

```
C:\Users\Kinsam\Downloads\nc111nt>nc  192.168.1.36  8080  <
clientEventRequest.txt

HTTP/1.1 500 Internal Server Error
Server: Apache-Coyote/1.1
Content-Type: text/html;charset=utf-8
Content-Length: 7046
Date: Sun, 07 Apr 2013 19:46:58 GMT
Connection: close

<html><head><title>Apache Tomcat/7.0.37 - Error report</ti-
tle><style><!--H1 {fon
t-family:Tahoma,Arial,sans-serif;color:white;background-
color:#525D76;font-size:
22px;}       H2       {font-family:Tahoma,Arial,sans-ser-
if;color:white;background-color:#525
D76;font-size:16px;}   H3   {font-family:Tahoma,Arial,sans-
serif;color:white;backgro
und-color:#525D76;font-size:14px;} BODY {font-family:Taho-
ma,Arial,sans-serif;col
or:black;background-color:white;}    B    {font-family:Taho-
ma,Arial,sans-serif;color:w
hite;background-color:#525D76;} P {font-family:Tahoma,Ari-
al,sans-serif;backgroun
d:white;color:black;font-size:12px;}A       {color       :
black;}A.name {color : black;}HR
```

Figure 10: Section of Internal Server Error

Figure 11: Comparison of connections (top normal case, bottom syn flooding case)

A totally different approach to the technical tests done before was the NIST security assessment. NIST provides already a sample questionnaire [12]. There is also a self assessment guide, where more questions are provided [16]. Out of these questions the most important for AMAAIS were selected and listed. The next clause is related to this NIST

document. First of all, the category of sensitivity of the assessment should be clarified. If the confidentiality is high, this means that it must be protected from unauthorized disclosure. Integrity means that the information must be protected from unauthorized, unanticipated or unintentional modification. This includes authenticity, non-repudiation and accountability. The first term means, that a third party must be able to verify that the content has not changed in transit. Non-repudiation means that the origin or the receipt of a message must be verifiable by a third party. With accountability it is meant that a security goal that generates the requirement for actions of an entity, has to be traced uniquely to that entity. The last content of the sensitivity assessment is the availability. Thus the information resource must be available on a timely basis. The different levels, high, medium and low are distinguished as follows. A high level means that it could cause loss of life, imprisonment, major financial loss, or require legal action for correction. A medium level could cause significant financial loss or require legal action for correction. A low level would cause only minor financial loss or require only administrative action for correction.

After the assessment of the sensitivity the selected questions should be answered. For each question the level as described in Section 3.2 should be assessed. For each question there is also a field called "risk based decision made". This means that a decision could be done to either provide more rigid controls than they are addressed by the questionnaire or the decision not to implement the control. The questions begin with the section personnel security that includes questions related to the members of the team working in the organization. The next section is about authentication which includes questions about passwords and the transmission of passwords. This section is followed by the section risk management. The next section contains question about security control reviews. The last section includes questions about physical and environmental protection. This questionnaire is attached in APPENDIX A (Adapted NIST Questionnaire).

4.2.1 Countermeasures

To avoid attacks like replay attacks the connection between server and client should be encrypted. An attacker could still sniff packets and re-inject them, but he does not have insight to the structure of the packets. Therefore, it is more complicated to modify a packet that still would be accepted. Alexander describes three other countermeausures for replay attacks. First approach is the assignment of unique random numbers. The next approach are timestamps for the packets. And the last approach are unique tokens, that a server sends to a client, before it can start the communication [1].

Although the consideration that the server was blocked during the performed DoS attacks due to hardware limitations, some countermeasures are described according Alexander in the following. A first short-term approach is the increase of resources such as CPU power or memory. This could be sufficient to keep acceptable response times while other mechanisms like filter rules has taken place. A very effective procedure is the use of a CDN (content distribution network) for the distribution of a service among different servers. The disadvantage of this are high costs for commercial CDNs.

5 Open Issues

Until now, the AMAAIS server has only been running on a local MacBook Pro. Thus, the server was already limited by the hardware. For a future testing, the server should be running on a better system that can handle more connections. To bring the server to a limit, a DDoS (distributed denial of service) attack should be performed. Thus multiple machines should generate a huge amount of traffic. These tests might happen over a big time period. Because the tests performed until now only took place within a few minutes. For security issues it might be better to test the whole system in a bigger manner. That means with a testing team within a bigger time frame. The reason for this propositions is that the AMAAIS system has to be very secure, because it works with sensitive data. There are also many other possible attacks that should be performed too, if the server is running in a real environment. This could include packet crafting, IP spoofing or Smurf/Fraggle attacks. As packet crafting Alexander defines the attack where tcp packets with the same destination address as the source address. When the attacker sends a packet with a faked IP address, the author calls it IP spoofing. Smurf/Fraggle attacks are Reflected DoS attacks. Alexander describes two main goals of Smurf/Fraggle attacks. The first one is to overstress a destination IP and the second one to reach a high network load. This is done by broadcast storms or broadcast floodings [1].

Within this assignment the focus was on DoS attacks such as the adapted syn flooding and a combination of a DoS and replay attack, by injecting a big amount of duplicated and modified messages.

Acknowledgement

Many thanks to all who supported me during this work. Special thanks to Christos Tsiaras for supervising me and to Andri Lareida for the great help to get the AMAAIS server running.

References

[1] M. Alexander, "Netzwerke und Netzwerksicherheit - Das Lehrbuch", Hüthig, Wien, 2006, (Chapter 2.4 Attacken und Gegenmassnahmen).

[2] I. Arce, G. McGraw, "Guest Editors' Introduction: Why Attacking Systems Is a Good Idea", Secu- rity & Privacy, IEEE , vol.2, no.4, pp. 17- 19, 2004. URL: http://ieeexplore.ieee.org/stamp/ stamp.jsp?tp=&arnumber=1324593&isnumber=29316. Visited in February 2013.

[3] T. Caldwell, "Ethical hackers: putting on the white hat", Network Security, Volume 2011, Issue 7, Pages 10-13, 2011. URL: http://www.sciencedirect.com/science/article/pii/ S1353485811700757). Visited in February 2013.

[4] G. Carl, G. Kesidis, R.R. Brooks, R. Suresh, "Denial-of-service attack-detection techniques", Internet Computing, IEEE , vol.10, no.1, pp. 82- 89, 2006. URL: http://ieeexplore.ieee.org/stamp/ stamp.jsp?tp=&arnumber=1580418&isnumber=33376. Visited in February 2013.

[5] Deliverable D2, "AMAAIS Phase 2: Architecture Design and Implementation". URL:http://www.csg.uzh.ch/research/amaais.html. Visited in March 2013.

[6] Deliverable D3, "AMAAIS: Workflow, Accounting Model and API". URL:http://www.csg.uzh.ch/research/amaais.html. Visited in March 2013.

[7] DiffMerge. URL: http://download-us.sourcegear.com/DiffMerge/3.3.2/DiffMergeManual.pdf. Visited in April 2013.

[8] G. Gruman, "OS X Mountain Lion Bible", John Wiley & Sons, 2012.

[9] D. M. Hafele, "Three Different Shades of Ethical Hacking: Black, White and Gray", SANS Institute, (2004). URL: http://www.sans.org/reading_room/whitepapers/hackers/shades-ethical-hacking-black-white-gray_1390. Visited in March 2013.

[10] Najmi, "How Hackers/Crackers Break Into Your System?", 2002. URL: http://www.techiwarehouse.com/engine/17249a96/How-Hackers/Crackers-Break-Into-Your-System. Visited in March 2013.

[11] Netcat. URL: http://netcat.sourceforge.net. Visited in April 2013.

[12] NIST: Federal Information Technology Security Assessment Framework, Nov 28, 2000, URL: http:// csrc.nist.gov/drivers/documents/Federal-IT-Security-Assessment-Framework.pdf. Visited in April 2013.

[13] Nmap. URL: http://nmap.org. Visited in April 2013.

[14] C. Palmer, "Ethical hacking", IBM Systems Journal , vol.40, no.3, pp.769-780, 2001.

[15] E. S. Raymond, "The New Hacker’s Dictionary", MIT Press, Cambridge, MA (1991).

[16] M. Swanson, "Security Self-Assessment Guide for Information Technology Systems", NIST, (2001). URL: http://infohost.nmt.edu/~sfs/Regs/sp800-26.pdf. Visited in April 2013.

[17] Z. Wilson and M. Poulin, "Hacking: The Basics", (2004). URL: http://www.sans.org/reading_room/whitepapers/hackers/hacking-basics_955. Visited in February 2013.

[18] Wireshark. URL: http://www.wireshark.org/about.html. Visited in April 2013.

APPENDIX A. Adapted NIST

Table 2: Adapted NIST Questionnaire

Category of Sensitivity	**Confidentiality**		**Integrity**		**Availability**	
High						
Medium						
Low						
Specific Control Objectives	**L.1 Policy**	**L.2 Procedures**	**L.3 Implemented**	**L.4 Tested**	**L.5 Integrated**	**Risk Based Decision Made**
Personnel Security						
Are all positions reviewed for sensitivity level?						
Are mechanisms in place for holding users responsible for their actions?						
Authentication						
Are passwords, tokens or biometrics used?						
Do passwords contain alpha numeric, upper/lower case and special characters?						
Are passwords changed at least every ninety days or earlier if needed?						
Is there guidance for handling lost and compromised passwords?						
Are passwords transmitted and stored with one-way encryption?						
Is there a limit to the number of invalid access attempts that may occur for a given user?						
Risk Management						
Are risk assessments performed and documented on a regular basis or whenever the system, facilities, or other conditions change?						

Table 2: Adapted NIST Questionnaire

Has data sensitivity and integrity of the data been considered?						
Have threat sources, both natural and man-made, been identified?						
Has a list of known system vulnerabilities, system flaws, or weaknesses that could be exploited by the threat sources been developed and maintained current?						
Has a mission/business impact analysis been conducted?						
Review of Security						
Has the system and all network boundaries been subjected to periodic reviews?						
Has an independent review been performed when a significant change occurred?						
Are routine self- assessments conducted?						
Are tests and examinations of key controls routinely made, i.e., network scans, analyses of router and switch settings, penetration testing?						
Physical and Environmental Protection						
Is access to facilities controlled through the use of guards, identification badges, or entry devices such as key cards or biometrics?						
Does management regularly review the list of persons with physical access to sensitive facilities?						
Are unused keys or other entry devices secured?						
Is suspicious access activity investigated and appropriate action taken?						
Are sensitive data files encrypted on all portable systems?						