

PART FOUR

**E-TERRORISM AND INTRUSION
DETECTION**

INFORMATION WARFARE

Hartmut Pohl

Hartmut.Pohl@sang.net

University of Applied Sciences, Bonn-Rhein-Sieg and ISIS – InStitute for Information Security, Cologne, Germany.

Keywords: Attack types, Critical infrastructure protection, Information warfare.

1. INTRODUCTION

The changes in technology in the last 10 years and the future will definitely result into a convergence of communications and computing in the fields of communications (telephone mobile also, internet, satellite), electric power, gas, oil, water supply, banking, stock exchanges, insurances, (air) traffic control, emergency services and disease management, information processing facilities of governments, governmental activities etc. in all countries especially in the first world. These and others are the so called critical infrastructures [Clinton 1996]. They are highly vulnerable because of their dependability of computers (hardware and especially software). Attacking one of these fields or infrastructures may result in a total disaster of the whole state.

The critical infrastructures depend on each other – for example the traffic depends on the telephone/fax and internet. The internet depends on electric power, electric power distribution depends on the internet. There is no only one independent critical infrastructure.

It is possible to connect to the internet all over the world with cost about less than 50 \$ US per month. Computers cost about 500 \$ US. Therefore the attack costs are low – especially compared with the possible damage.

2.DEFINITIONS

2.1 Information Warfare

I will discuss information warfare as a warfare attacking information systems by using information systems to destroy information processing of a town, region or country with the aim to damage or destroy one or more critical infrastructures.

I will not use the word information warfare as the classical psychological warfare and the distribution of information by mail, files, papers, radio or TV. And I definitely will not use the word information warfare as information operations conducted during time of crisis or conflict to achieve information superiority or promote specific objectives over a specific adversary or adversaries. [DoD]

2.2 Strategy to Secure Cyberspace

National strategies to secure cyberspace are part of our overall effort to protect our nations. It is an implemented component of the national strategy for homeland security and is complemented by a national strategy for the physical protection of critical infrastructures and key assets. [Bush 2003].

2.3 Other Aspects of Information Warfare

Information Warfare can be divided into:

- Offensive information warfare.
- Defensive information warfare with all the security measures like access control, encryption, filtering (firewalls), monitoring, detection and prevention of intrusions, management of information security in companies, agencies and states.

In this paper especially the first aspect offensive information warfare is discussed.

3.CASES OF INFORMATION WARFARE

There are only few cases of information warfare seriously published – especially cases of business information warfare.

4. SEVEN THESES FOR THE FUTURE OF INFORMATION WARFARE.

4.1. Aims

The aim of an information warfare is to destabilize a state, a region or a government by shutting down one of the critical infrastructures – especially first of all the communication infrastructure, which serves the other infrastructures.

4.2. Attack Types

The attack types are very well known like viruses, worms, buffer overflows, Trojan horses, etc. Some are only a little bit hypothetical like the Warhol–worm or Flash–worm. The attacks of the future will be very quick in attacking most servers of the internet in minutes and will act for a long time covert.

4.3. Perpetrators and Motives

Perpetrators are of the level of computer criminals: High grade experts are specialised in the fields of operating systems, communications, database systems, and standard software like SAP, People soft etc. You can find those potential perpetrators, specialised experts all over the world. Most countries offer studies in computer science and studies in computer security and information security.

4.4. National Activities versus Coordinated Global Planning

The mentioned critical infrastructure do not end at the borders of national states but exceed continents like the communication infrastructure (internet), telephone system, electric power, gas, oil, water supply, banking, stock exchanges, insurances, and air traffic control. National activities are not useless but not equivalent to the global risks; one nation alone is not able to secure the links to the continental or global infrastructures.

4.5. Critical Infrastructure Protection

The way to secure critical infrastructures is to identify the most vulnerable infrastructures first and secure them. But because of the linkage between the infrastructures it is very necessary to secure them all.

4.6. Legal, Political and Technical Security Measures

It is necessary to adopt security legislation in all states and initiate security programs. These legal and political measures are also necessary on a supranational level.

The mentioned measures of the defensive information warfare have to be installed in total depending on the value of the processed data.

4.7. Arms Control

It is necessary to control the development of the mentioned attacks and new ones all over the world – for example by monitoring the internet.

5. CONCLUSIONS

Information warfare will be the war of the future between high tech states; and as asymmetric warfare between low and high tech states. The same attack types will be used by terrorist – small groups of men (one or more) attacking high tech companies and states and also bigger groups of men against (international) companies or states.

6. REFERENCES

Alberts, D.S.; Garstka, J.J.; Stein, F.P.(2000), Network Centric Warfare. Developing and Leveraging Information Superiority. 2nd Edition Washington.

Arquilla, J.J.(1993), Cyberwar is coming.
<http://gopher.well.sf.ca.us:70/0/Military/cyberwar>

Arquilla, J.J.; Ronfeldt, D.F.(1995), Cyberwar and Netwar: New Modes, Old Concepts, of Conflict.
<http://www.rand.org/publications/RRR/RRR.fall95.cyber/cyberwar.html>

Bush, G. (2003), The National Strategy to Secure Cyberspace. Washington.

Clinton, W.J.(1996), Critical Infrastructure Protection. Executive Order 13010. The White House <http://www.pccip.gov/eo13010.html>

Denning, D.(1999), Information Warfare and Security. Reading.

DoD (Ed.): DoD Dictionary of Military Terms. O.J.
<http://www.dtic.mil/doctrine/jel/doddict/data/i/03097.html>

Geiger, G.(1997), Verteidigung im "Cyberspace". Internationale Probleme, nationale Aufgaben. Ebenhausen.

Johnson, L. S., Toward a Functional Model of Information Warfare
http://www.infowar.com/mil_c4i_101497a.html-ssi o.J.

Karresand, M.(2002), A Proposed Taxonomy of Software Weapons. Linköping.

Molander, R.C (1996), Riddile, A.; Wilson, P.A.: Strategic Information Warfare. Santa Monica.

PCCIP (1997), President's Commission on Critical Infrastructure Protection: Survey Form. Washington.

PCCIP, President's Commission on Critical Infrastructure Protection: Critical Foundations. Protecting America's Infrastructures. The Report of the President's Commission on Critical Infrastructure Protection. Washington o. J.
<http://www.dis.anl.gov/survey>

Pohl, H.(1996), Informationssicherheit der Global Information Infrastructure (GII) - Einige Bemerkungen zu Problemen und Entwicklungen. In: Tauss, J. et al. (Hrsg.): Deutschlands Weg in die Informationsgesellschaft. Herausforderungen und Perspektiven für Wirtschaft, Wissenschaft, Recht und Politik. S. 358 - 390. Baden Baden.

Pohl, H. (1998), Information Warfare – Information Survivability. Datenschutz und Datensicherung, 2.

Pohl, H.(1998), Information Warfare: Der Krieg im Frieden. Zusammen mit Cerny, D. In: Bauknecht, K.; Büllsbach, A.; Pohl, H.; Teufel, S. (Hrsg.): Sicherheit in Informationssystemen SIS '98. Zürich .

Pohl, H.(2000), Business Information Warfare. In Geiger, G. (Hrsg.): Sicherheit der Informationsgesellschaft. Gefährdung und Schutz informationsabhängiger Infrastrukturen. Baden Baden.

Pohl, H.(2000),Information Warfare. In: Reinermann, H. (Hrsg.): Regieren und Verwalten im Informationszeitalter. Heidelberg.

Pohl, H., Civil War in Cyberspace. Ziviler Ungehorsam, innere Unruhen und Bürgerkrieg in der Informationsgesellschaft.

Schubert, S. et al. (Hrsg.)(2002), Informatik bewegt. Proceedings der 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI). Bonn.

Rathmell, A.; Overill, R.; Valeri, L.; Gearson, J.(1997) The IW Threat from Sub-State Groups: An Interdisciplinary Approach. <http://kcl.ac.uk/orgs/icsa/terrori.htm>

Szafranski, R.(Preparing for 2020), A Theory of Information Warfare. o.J. <http://www.cdsar.af.mil/apj/szfran.html>

Szafranski, R.,Parallel War and Hyperwar: Is every eant a Weakness? <http://www.cdsar.af.mil/battle/chp5.html> o.J.

Vatis, M.(2001), Cyber Attacks during the War on Terrorism: A Predictive Analysis. Dartmouth http://www.ists.dartmouth.edu/ISTS/counterterrorism/cyber_a1.pdf.

Wenger, A.; Metzger,JDunn, M.(2002): The International CIIP Handbook. An Inventory of Protection Policies in Eight Countries. Zürich.

STATE OF THE ART VULNERABILITY DETECTION AND SUGGESTIONS FOR IMPROVEMENT

H.S. Venter

hventer@cs.up.ac.za

J.H.P. Eloff

eloff@cs.up.ac.za

Department of Computer Science, University of Pretoria, 0002, Pretoria, South Africa.

Abstract: The focus of this paper is to give an overview of current vulnerability scanner (VS) products and to provide ideas for future improvements. Since each VS product available on the software market today is developed by a separate vendor, there are significant differences in these VS products. VS products differ extensively from each other. The main differences between state of the art VS products are the types of vulnerabilities that are detected as well as the number of vulnerabilities that can be detected. This paper suggests the concept of a common set of vulnerability categories, referred to as harmonised vulnerability categories, to be used by different VS products. Furthermore it introduces the concept of vulnerability forecasting.

Keywords: Harmonised vulnerability categories, Vulnerability, Vulnerability scanner (VS), Vulnerability mapping, Vulnerability assessment, Vulnerability forecasting.

1. INTRODUCTION

Due to the increasing awareness of the public of security issues on the Internet, the number of security products available on the software market today is myriad and still increases. This is why you face a dilemma when choosing the right security product for your organisation's security needs.

There are many ways in which information can be secured by using various information security technologies [VEE1 03]. Computer security in an organisation can generally be addressed in two ways: before a security incident can take place, or after a security incident has taken place. Security that is addressed before a security incident takes place is referred to as proactive security. Proactive security is implemented by using vulnerability scanner (VS) products. Security addressed after a security incident has taken place, or when the security incident is still taking place, is referred to as reactive security. Reactive security is implemented by intrusion detection systems [BACE 00].

The focus for this paper, however, is to develop a better understanding of current state of the art in VS products. Vulnerability scanning means having an automated scanning program, referred to as a VS, that scans a computer or a network of computers for a list of known weaknesses, referred to as vulnerabilities [SCHN 00]. In other words, vulnerability scanning refers to the application of state-of-the-art information security technology to secure information on the Internet [VEE1 03].

There are many VS products available on the software market. They often refer to the same vulnerability in a different way and this makes it very difficult to see exactly which vulnerabilities are scanned for by the different VS products. This dilemma can be solved by using the framework of **harmonised vulnerability categories** [VEE2 03]. Other aspects of VS products are also considered in this paper, for example, the specific database structure of a VS. These aspects are discussed in an attempt to shed more light on the problems that the different VS products pose.

The sections that follow will discuss VS products in more detail. An overview of the current VS products is discussed. Some of these products are discussed in detail, with the emphasis on the databases that these VS products employ. Some issues on the future of VS products concludes this paper.

Table 1: The harmonised vulnerability categories

Harmonised vulnerability categories	
1	Password cracking and sniffing
2	Network and system information gathering
3	User enumeration and information gathering
4	Backdoors, Trojans and remote controlling
5	Unauthorised access to remote connections & services
6	Privilege and user escalation
7	Spoofing or masquerading
8	Misconfigurations
9	Denial-of-services (DoS) and buffer overflows
10	Viruses and worms
11	Hardware specific
12	Software specific and updates
13	Security policy violations

2. VS PRODUCTS

It is important to be aware of the different VS products available on the software market before studying some of them in more detail. There are freeware as well as commercial versions of VS products available and some of the products differ extensively from other products. The section that follows lists some of the major role players in VS technology available today and attempts to place the different aspects of the products in perspective to each other.

2.1. VS Product Overview

Table 2 shows a list of two well-known VS products available today in no particular order of preference.

Table 2: VS products

VS product	Commercial or freeware	Reference
bv-Control	Commercial	[BIND 03]
Internet Security Scanner (ISS) 6.2.1	Commercial	[ISSN 03]
Nessus Security Scanner	Freeware	[DERA 03]
Security Administrator's Integrated Network Tool (SAINT) 4.0	Commercial	[SAIN 03]
Security Analyzer 5.1	Commercial	[NETI 03]

The SAINT, the ISS, and the Nessus Security Scanner will be discussed in more detail in the following sections. The focus of the discussion of these products will not be to evaluate and compare them with each other, but rather to comment on the practical experience encountered by the authors while

working with the products. This is followed by elaborative discussions on each product's vulnerability database in terms of differences.

2.2. The SAINT

The Security Administrator's Integrated Network Tool (SAINT) [SAIN 03] is discussed in this paper because it was freely available until recently and supports the use of CVE. CVE is an acronym for "Central Vulnerabilities and Exposures" [MITR 03]. CVE is a list or dictionary that provides common names for publicly known information security vulnerabilities and exposures. The SAINT can run on UNIX and LINUX operating systems and also scans for vulnerabilities on multiple operating systems. The SAINT is also available in an online version.

2.2.1 Practical Experience with the SAINT

Because the SAINT incorporates CVE into its vulnerability database, standard vulnerability names are used. In addition, CVE's Web site also has more information available on how to fix the detected vulnerabilities. This is a major advantage of the SAINT. The disadvantage of the SAINT is that it categorises its vulnerabilities into 177 categories, which makes it difficult to work with. It is better to have fewer vulnerability categories that are more manageable as the harmonised vulnerability categories suggest.

2.2.2 The SAINT Vulnerability Database

Of the 13 harmonised vulnerability categories, Password cracking and sniffing, User enumeration and information gathering, Backdoors, Trojans and remote controlling, Spoofing or masquerading, Viruses and worms, Hardware specific, and Security policy violations are covered in very little detail, if at all, by the SAINT's vulnerability database.

2.3. The Internet Security Scanner (ISS)

The ISS version 6.2.1 is discussed in this paper because the ISS was one of the first VS products available on the software market with a graphical user interface. It is established and widely used in the industry today. There is an ISS version [ISSN 03] that can be downloaded from the Internet free of charge with full functionality, but it is limited to scan only the host on which it is installed.

The ISS supports the CVE standard to enable users to easily determine if issues with different names are the same, and to allow for efficient sharing of security information. A CVE reference, however, may not exist for every

vulnerability check used in the ISS and because of this CVE is only partially supported by the ISS.

2.3.1 Practical Experience with the ISS

The ISS was installed on a Windows workstation and then set up to scan workstations and servers connected to the network for the vulnerabilities as specified in its vulnerability database. The ISS runs on Windows and has a very good user interface, but it can also scan for vulnerabilities on other operating systems like UNIX. Depending on the size of the network and the specific scan policy that is set up before the scan can commence, the ISS scans the network for vulnerabilities and is relatively fast. A scan on a Windows workstation was completed in just over four minutes before a report was generated. Figure 1 shows an extract of one of the vulnerabilities in this report.

The advantages of the ISS report are that it contains good and detailed descriptions and remedy procedures. In addition, a reference to additional information for the specific vulnerability detected is provided as well as information on which operating system platforms the particular vulnerability can occur. Another big advantage is that the ISS classifies the particular vulnerability into a low, medium, or high risk factor so that the rectification of vulnerabilities can be prioritised. The disadvantage of this report is that it requires effort to work through because of its large size.

Modem detected and active (Active Modem)	
Risk Level:	Medium
Platforms:	Windows NT, Windows 95, Windows 98, Windows 2000, Windows ME
Description:	An active modem driver was detected. This situation only occurs when the modem is in use, or when the modem driver program is active. Modems can be a sign of an unauthorized channel around your firewall. Attackers could use a modem within the network to circumvent network security.
Remedy:	The modem must not be active while the computer is attached to the network. You may want to minimize the impact of a security breach caused by an unauthorized modem use by limiting which systems trust the computer using the modem. If using a modem on the network is required, configure all Remote Access Setup ports so that the port usage can dial-out only. Verify that your dial-out network configuration protocols match exactly the protocols you need to access the remote network. Review share permissions and account security to verify that the file system is not accessible from a remote location.
References:	ISS X-Force Modem detected and active http://xforce.iss.net/static/1292.php

Figure 1: An extract from the ISS report

2.3.2 The ISS Vulnerability Database

Of the 13 harmonised vulnerability categories, User enumeration and information gathering, Privilege and user escalation, Spoofing or masquerading, Misconfigurations, and Viruses and worms are covered in very little detail, if at all, by the ISS's vulnerability database.

2.4. The Nessus Security Scanner

The Nessus Security Scanner is discussed in this paper because it is a widely known freeware product [TALI 00]. The Nessus Security Scanner executes mainly on UNIX-based platforms, but it can scan for vulnerabilities on multiple operating system platforms. The Nessus Security Scanner is built upon client-server architecture where the server works on a UNIX-based platform. Different clients are available that can run on a UNIX or Windows

operating system platform. The Nessus Security Scanner also supports CVE references.

2.4.1 Practical Experience with the Nessus Security Scanner

The Nessus Security Scanner works on the concept of plug-in architecture. This means that there is a plug-in for each vulnerability that the Nessus Security Scanner can check for. This way, it is easy to add new vulnerability signatures as external plug-ins when they become available. These can simply be downloaded from the Nessus Security Scanner Web site [DERA 03] via FTP.

It is also possible to add customised vulnerability signatures. To be able to do this, the Nessus Security Scanner includes the Nessus Attack Scripting Language (NASL), which is a language designed to write customised vulnerability signatures easily and quickly. These plug-ins then also constitute the vulnerability database for the Nessus Security Scanner.

The biggest advantage of the Nessus Security Scanner is that it is very fast. The vulnerability tests performed by the Nessus Security Scanner co-operate so that nothing is done that is not necessary. For example, if an FTP server is found not to offer anonymous logins, then anonymous-related vulnerability checks will not be attempted or performed for anonymous FTP vulnerabilities, which saves time. Some VS products will attempt to scan for anonymous FTP vulnerabilities, if their scan policies were set up to do that, even if no anonymous FTP vulnerabilities are present. This causes those VS products to waste valuable time since it will not continue to scan for the next vulnerability, as defined by its scan policy, until scanning for anonymous FTP vulnerabilities has timed out. Another advantage of the Nessus Security Scanner is that it categorises the risk level of each vulnerability from low to very high in the report that it generates, enabling one to prioritise the urgency of fixing the vulnerabilities found. The disadvantage of this report, however, is that it requires effort to work through because of its large size.

2.4.2 The Nessus Security Scanner Vulnerability Database

Of the 13 harmonised vulnerability categories, *Password cracking and sniffing*, *User enumeration and information gathering*, *Spoofing or masquerading*, *Misconfigurations*, *Viruses and worms*, *Hardware specific*, and *Security policy violations* are covered in very little detail, if at all, by the Nessus Security Scanner's vulnerability database.

3. SUMMARY OF CURRENT VS PRODUCTS

In the previous sections different VS products were discussed and the reader should have a better understanding of how different the VS products operate. In essence all these products have one main goal: identifying vulnerabilities. But the way that these VS products go about in accomplishing this goal, often differ extensively from one VS product to another. What is more – these different VS products do not all scan for exactly the same type of vulnerabilities. Fortunately, by making use of harmonised vulnerability categories [VEE2 03], a measure is available to identify how the different VS products comply with harmonised vulnerability categories.

Figure 2 shows a mapping, compiled for this paper, of the vulnerabilities found for each of the five VS products discussed in the previous sections onto the harmonised vulnerability categories. The mapping process was done for each individual VS product. The vulnerability database of a specific VS product was carefully dissected by studying each vulnerability as defined in the vulnerability database. A particular vulnerability is then allocated to one of the 13 harmonised vulnerability categories.

From figure 2 it is clear that the different VS products comply differently with the 13 harmonised vulnerability categories. For example, the Nessus Security Scanner can detect far more network and system information gathering (category 2) vulnerabilities than all the other VS products. The Internet Security Scanner, on the other hand, outperforms all the other VS products when detecting Password cracking and sniffing (category 1), Backdoors, Trojans and remote controlling (category 4), Unauthorised access to remote connections & services (category 5), Spoofing or masquerading (category 7), Software specific and updates (category 12), and Security policy violations (category 13) vulnerabilities. In addition, only one VS product namely the Nessus Security Scanner scans for viruses and worms (category 10) and only for a very limited number of viruses and worms. The ISS, therefore, seems to be the VS product with the best amount of vulnerabilities that it can scan for across the harmonised vulnerability categories.

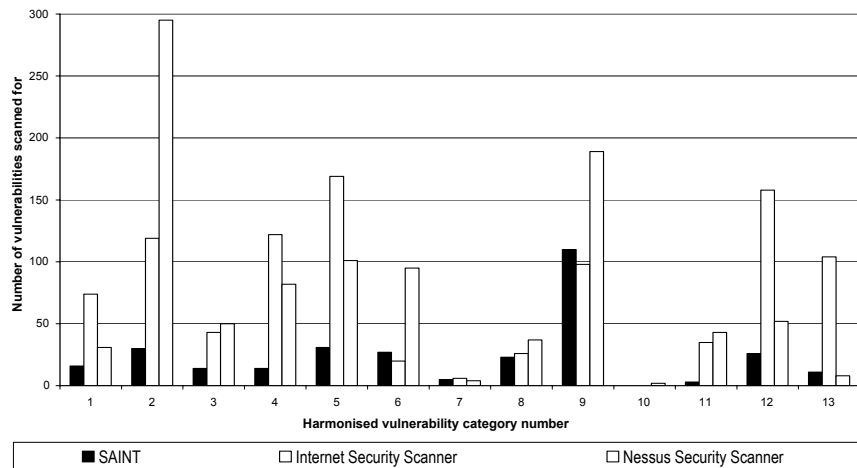


Figure 2: Vulnerability mapping of different VS products onto the harmonised vulnerability categories

4. THE FUTURE OF VS PRODUCTS

Although the proactive behaviour of VSs is a positive point, there are still many problems with state-of-the-art VSs. Problems such as the length and complexity of vulnerability reports produced by VS products as well as a complete absence regarding the ability of VS products to contribute to risk management on networks should be addressed. In a bid to address these and other types of problems, a conceptual model is introduced in this section.

4.1 Problems with State-of-the-Art VS Products

In summary, table 3 lists the problems identified with state-of-the-art VS products. In order to minimise the impact of these problems, the authors would like to introduce the concept of **vulnerability forecasting** as a future initiative to vulnerability scanning.

The term “vulnerability forecasting” (VF) can be defined as “that attempt to identify potential vulnerable areas on hosts across a network and to what extent such areas on hosts across a network will be vulnerable over a specific period in the near future”. The principal aim of VF is, therefore, to predict trends or patterns in which potential vulnerabilities could occur. Knowing what such a vulnerability forecast is means that proactive action can be taken in a bid to minimise the risks that such vulnerabilities may pose.

Table 3: Problems identified and addressed regarding state-of-the-art VS products

Problems identified
1. Conducting vulnerability scans is too time-consuming.
2. A VS product generally occupies a vast number of network and system resources, leading to the degradation of system performance while vulnerability scans are being conducted.
3. VS products lack intelligence because they are unable to learn about new vulnerabilities automatically.
4. The vulnerability database structure differs extensively from one VS product to another.
5. The types of vulnerabilities being scanned for differ extensively from one VS product to another.
6. Scans may not always be conducted at regular intervals due to unforeseen circumstances, for example when critical maintenance on servers and the network is carried out.
7. The vulnerability database should be updated before a scan is conducted, otherwise the scan result may not be accurate enough.
8. Most rectification procedures cannot be automated and still require the expertise of qualified personnel.
9. VS products do not provide adequate and sufficient information that would aid high-level risk management.

4.2 A Conceptual Model for VF

The high-level design of the conceptual VF model comprises three main components and is depicted in figure 3 below:

A brief description of the main components in figure 3 follows:

6. VS Technology (current)

This component constitutes one or more state-of-the-art VS products that are used for collecting the data needed for VF.

7. Vulnerability Harmonisation

This component serves as a coupler between the VS technology and the vulnerability forecasting components in a bid to “standardise” the VS product’s output into a harmonised form.

8. Vulnerability Forecasting

This component does the actual intelligent vulnerability forecast.

Each of the main components of the conceptual VF model, as introduced in the previous section, contains subcomponents.

4.2.1 The VS Technology (Current) Subcomponent

The reason for using current VS technology in the VF model enables the use of existing technology rather than attempting to design yet another module in the VF model. In addition, any current VS product can be used in the conceptual VF model, rendering the conceptual VF model more flexible. In summary, a VS product analyses the security state of a network of hosts on the basis of information collected, referred to as scans, at different intervals. After a scan is completed, the VS product generates scan results in the form of a report that states all the vulnerabilities found during the scan and leaves it up to a person to rectify these vulnerabilities.

4.2.2 The Vulnerability Harmonisation Subcomponent

The vulnerability harmonisation component is represented as the second subcomponent of the conceptual VF model. This component does not do the actual vulnerability forecasting yet, but serves as an in-between process where the data it received from component 1 of the conceptual VF model is transformed in such a way that it is “harmonised” and, thus, prepared to be “understood” by component 3 of the conceptual VF model. In summary, the output of the VS product in component 1 of the conceptual VF model, namely the scan result, serves as input to the vulnerability mapper in component 2 of the conceptual VF model. The vulnerability mapper maps the vulnerabilities found by the VS product onto the harmonised vulnerability categories and stores the result in the harmonised history database. This process is repeated each time a vulnerability scan is conducted.

4.2.3 The Vulnerability Forecasting Subcomponent

The vulnerability forecast component constitutes the third subcomponent of the conceptual VF model. This main component does the actual vulnerability forecasting. In summary, the output of the vulnerability mapper in component 2 of the conceptual VF model, namely the harmonised history database, serves as input to the vulnerability forecast engine in component 3 of the conceptual VF model. The vulnerability forecast engine attempts to predict trends or patterns, in terms of harmonised vulnerability categories, in which potential vulnerabilities could occur. The Vulnerability forecast engine

constitutes the heart of the conceptual VF model. Intelligent techniques are used in conjunction with history scan data and history forecast data to forecast which harmonised vulnerability category or categories would potentially pose vulnerability problems in the near future.

5. CONCLUSION

This paper discussed different VS products and looked at how each respective product differs in the way that they can scan for vulnerabilities and what the impact of vulnerability forecasting may be on VS products.

It was found that VS products differ extensively from each other in terms of the number of vulnerabilities that each different VS is able to detect. In the sections above it is clear that – most of the time – using the vulnerability count is a good way to determining what the differences are between different VS products.

Far from rendering existing VS products obsolete, VF is used proactively to co-ordinate output from existing VS products with that gleaned from intelligent techniques and history data.

The concept of vulnerability forecasting has many advantages. It saves considerable time, because instead of scans being conducted all the time to detect and rectify vulnerabilities; scans can now be conducted less frequently. Having vulnerability forecasts, vulnerability problem areas – in the form of harmonised vulnerability categories – can be attended to before they can erupt. In due course, system resources are occupied less often due to fewer scans that need to be conducted.

Using harmonized vulnerability categories along with vulnerability forecasting renders the process of doing vulnerability forecasting VS product independent. The difference in the types of vulnerability categories that various VS products scan for is therefore bridged by the use of harmonised vulnerability categories. In addition, adequate and sufficient risk management can be done due to the fact that, each time a vulnerability forecast is done, vulnerability forecasts are made available for each harmonised vulnerability category.

6. REFERENCES

[BACE 00] BACE, R. G.; 2000; Intrusion Detection; “Intrusion Detection Concepts”, pp. 37-43; Macmillan Technical Publishing; ISBN 1-57870-185-6.

[BIND 03] BINDVIEW CORPORATION; 2003; Proactive security management software and services; “bv-Control: the security solution to manage within and between organizations”; <http://www.bindview.com>.

[DERA 03] DERAISON, R.; 2003; Nessus Security Scanner; “What is Nessus Security Scanner?”; <http://www.NessusSecurityScanner.org/intro.html>.

[ISSN 03] INTERNET SECURITY SYSTEMS; 2003; Internet Security Systems; “ISS”; <http://www.iss.net>.

[MITR 03] THE MITRE CORPORATION; 2003; Common Vulnerabilities and Exposures (CVE); “CVE, The Key to Information Sharing”; <http://www.cve.mitre.org/introduction.html>.

[NETI 03] NETIQ; 2003; Products and Solutions; “Security Analyzer”; <http://www.netiq.com>.

[SAIN 03] SAINT CORPORATION; 2003; About SAINT; “SAINT 4 Vulnerability Assessment Tool”; <http://www.saintcorporation.com>.

[SCHN 00] SCHNEIER, B.; 2000; Secrets and Lies – Digital Security in a Networked World; “Intrusion Detection Systems”; pp. 194-197; John Wiley & Sons Inc.; ISBN 0-471-25311-1.

[TALI 00] TALISKER; 2000; Network Vulnerability Scanners; “Nessus”; http://www.networkintrusion.co.uk/N_scan.htm.

[VEE1 03] VENTER, H.S.; ELOFF; J.H.P.; 2003; Computers & Security; “A Taxonomy for Information Security Technologies”; Elsevier Science; ISSN 0167-4048.

[VEE2 03] VENTER, H.S.; ELOFF; J.H.P.; 2003; South African Computer Journal; “Harmonised Vulnerability Categories”; pp. 24-31; No. 29; Computer Society of South Africa South Africa; ISSN 1015-7999.

IP TRACEBACK OF DENIAL OF SERVICE (DOS) ATTACKS USING MOBILE AGENTS - TRACEABILITY IN E-SERVICES

Ghada El-Keissi

gh_elkeissi@hotmail.com

Sherif El-Kassas

sherif@aucegypt.edu

Computer Science Department, American University in Cairo, Egypt.

Abstract: A current important network threat is the launch of Denial of Service (DoS) attacks. The main problem behind such attacks is the ability of an attacker to spoof his IP address. Thus, it's very difficult to identify the actual attacker. Accordingly, tracing back an attacker to actual source became a very important step to respond to DoS attacks. This paper aims at introducing an improved technique in tracing back IP spoofed flooding attack by using mobile agent technology. The work presented is an improvement to work described in [1] and [2]. Attacker would be traced back through three different network topologies: LAN, Inter-connected Network and WAN. The agent system is made up of four different agents: Master, Manager, Sensor, and Tracer agents. Every LAN would have independent trace back system deployed on to it. Different trace back systems in different LANs would cooperate to trace back an attacker within inter-connected networks and WAN. Similar to work presented by [2], the trace back system is mainly based on the idea of using Data link-level identifier (MAC address) to identify the next hop in the trace path. This comes from the fact that it's common and easy to spoof IP address unlike MAC addresses. Experiment on real network topology has been conducted and result is described in the paper.

Keywords: IP traceback, Denial-of-service attacks, Mobile agents, IP spoofing.

1. INTRODUCTION

DoS attack is an attack that denies the target victim the ability to offer services to legitimate users. Many sites have been subject to such destructive attacks such as yahoo, cnn, and amazon. Most of the DoS attacks tend to be flooding attacks where the attacker uses spoofed IP addresses. Thus actual source of the attack couldn't be identified. Currently, there are many techniques used to respond to such destructive attacks. In this paper an improved technique based on work presented by [1] and [2] is presented. Its main aim is tracing back an attacker who launches a flooding attack using spoofed IP address. An experiment is described in details proving the ability of this improved method to identify actual source of the attack or at least the closest point to an attacker.

The rest of the paper will be organized as follows: Section 2 and Section 3 define DoS attacks and IP spoofing respectively. Section 4 describes the two categories of DoS attacks responsive mechanisms. Section 5 identifies an important responsive method which is IP trace back. It covers some common trace back methods used. Section 6 describes mobile agent technology. Section 7 outlines some related work in the field of IP trace back using mobile agent technology and data-link level identifiers in trace process. Section 8 presents in details the improved trace back technique implemented. Section 9 lists one of the experiments conducted and results achieved. Finally, section 10 summarizes findings and describes limitations and future work to be conducted.

2. DOS ATTACKS

As explained above, DoS is an attack that denies a target victim the ability to offer its services to legitimate users. It can achieve this in many ways e.g. flooding the network preventing legitimate traffic, disrupting connection between two machines preventing access to services, or disrupting service to system or user [3]. Thus the target victim resources could be fully consumed or even the system could crash making it unable to serve legitimate users. Example of the DoS attacks are UDP attacks [5], and TCP/SYN attacks [6].

3. IP SPOOFING

The widespread of DoS attack has emerged mainly due to weakness of security mechanisms implemented at different sites. Sites don't maintain and

update their security patches and anti-viruses, while attack tools have become highly advanced. Yet, the major reason for widespread of DoS attack is considered the ability of an attacker to send attacks to a victim using spoofed source IP address (i.e. fake IP address) hiding his true identity. This task has become a trivial one with the increase of the Internet size and number of IP addresses that could be easily forged by an attacker. With IP spoofing the victim finds it very difficult to take countermeasures against attacks. This comes from the fact that the actual source of an attack couldn't be determined, and it became difficult to decide whether the incoming packets are attack packets or legitimate ones. Accordingly the victim cannot decide whether to block an incoming traffic or not.

4. DOS RESPONSE TECHNIQUES

Many techniques have been used to mitigate the effect of DoS attacks. The techniques could be divided in to two categories: Proactive and Reactive. The proactive approach consists of techniques that deals and prevents an attack before they actually happen. Adding filters to routers, updating security patches at network hosts and using advanced intrusion detection tools makes good preventive methods. On the other hand, reactive approach tries to handle attack and mitigate its effect during and after the attack has taken place. IP traceback is one main reactive technique. This paper concentrates on reactive technique, mainly IP traceback, since it is considered more challenging and important solution to prevent DoS attacks compared to proactive techniques. This emerges from the fact that proactive methods can prevent an attack from happening at target host but don't stop an attack forever. Still attacker is there with the spoofed IP address and he is not discouraged to continue attacking other sites. Thus proactive techniques are not the best solutions, as they don't eliminate the problem as a whole.

5. IP TRACEBACK

IP traceback system is not easy to implement since it requires to traceback attacks from a LAN to another until the source of the attack is reached. This would require having a portable, extensible system that can spawn heterogeneous systems. Not only this, the IP traceback system implemented on target hosts could be an interesting target for DoS attack itself. An attacker could detect the components of traceback system and conduct flooding attack targeting it. In this case, the traceback system must be attack resistant. Many papers have described various techniques to

implement traceback systems. Some of the current IP Traceback systems described at [8] are as follows:

1. Link Testing: tracing starts at router closest to the victim and is repeated recursively on upstream routers. It stops when an attacker is identified, or trace leaves border of an ISP. The main disadvantage is that traceback process must occur during an attack, not after it.
2. Logging: logs packet information at key routers and then uses data mining techniques to determine the path the packet has traversed [8]. This is considered an expensive technique, due to high resource requirement.
3. Marking Algorithm: Burch and Cheswick have suggested a way to traceback an attack by marking packets probabilistically or deterministically with address of routers they traverse. Accordingly, the victim would use information in marked packets to traceback the source of the attack to its origin. The main disadvantage is that a packet might not have enough space for marked addresses.

Most of these developed techniques have many problems associated with them. In order to overcome the problems we need to build inexpensive, portable, extensible, attack resistant system that could spawn heterogeneous systems, and be able to trace an attack during or after an attack has been completed. Mobile agent technology was introduced to develop traceback systems.

6. MOBILE AGENTS

Mobile agents are mainly software components that can execute certain tasks. Agents reside at agent platforms that constitute their execution environment. The main agent's feature is its mobility and portability. The agent could move from one place to another and is portable. Its execution doesn't rely on the underlying OS and could spawn heterogeneous systems. Another important feature is that agents overcome network latency problem. In time of DoS attacks the network will already be congested. Tracing back attack might require transfer of data from one location to another. The use of mobile agents has overcome the problem of transferring bulky data at time of attack. Instead of transferring data to computation place, computation, in form of agent, is transferred to data. Of course agents would be much, much smaller in size compared to data. Example of mobile agent system is IBM Aglets [4].

Tracing intruders using mobile agents is considered a very efficient technique to solve DoS attacks. Midori, Shunji and Atsushi from Waseda University has tackled this problem and presented a solution [1]. Their work

was mainly to detect and traceback LAN attacks (i.e. attacks performed by users who have access to network machines) by using mobile agents. Based on their work, this paper will present an improved technique to traceback DoS attacks using mobile agents within Inter-connected network and WAN.

7. RELATED WORK

7.1. Tracing Intruders Using Mobile Agents

The Information-technology Promotion Agency (IPA) in Japan has developed an Intrusion Detection Agent (IDA) System [1] that could collect and gather information about intrusions and trace attackers with the help of mobile agents. The system is implemented fully to detect and respond to LAN attacks. The IDA system as described by [1] consists of manager, sensor, bulletin board, message boards, tracing agents, and information gathering agents.

When the sensor on a target host detects an MLSI (Marks Left by Suspected Intruder), the sensor would report that to the manager. Accordingly the manager would launch a tracing agent to that target. The tracing agent would launch an information-gathering agent on the target. The information-gathering agent would start to gather information about the MLSI on the target host. Tracing agent would try to identify the source origin of the attack using information about the network connections and processes running on the system. Independent of the tracing agent the information-gathering agent would report back information to the manager. The tracing agent would move to next hop in the tracing route until either it finds the attacker origin site or cannot move elsewhere. At this point it returns back to the manager. The tracing agents use message board to avoid the overlap of tracing routes by other tracing agents. Tracing agents use the message board to determine their path and destination.

7.2. Distributed IP TraceBack

A proposed architecture presented by [2] relies on hop-by-hop tracing where routers log packet information and keep this data to traceback attacks later on. The approach used goes beyond this where datalink-level identifiers such as Ethernet media access control (MAC) address, ATM virtual path channel identifier (VPI/VCI), and frame relay datalink connection identifier (DLCI) are used to identify the packet path [2]. The idea is mainly based on the fact that an attacker can easily spoof the source IP address but it will be extremely difficult for him to spoof the datalink-level identifier of the forwarding node. The forwarding node changes the packet

datalink-level identifier to match its interface identifier and by looking at the identifier in the packet the forwarding node can know the adjacent node through which the packet has passed through. The forwarding nodes, tracers, would keep information about the packet and its datalink-level identifier, in a buffer memory (also known as packet information area) [2]. Then it will identify the adjacent node by matching the datalink-level identifier of forwarded packet with that of the attack packet. The system architecture is composed of sensor, manager and tracer. It operates in a similar manner as [1] but by using distributed management approach [2] and not mobile agents.

8. IMPROVED IP TRACEBACK USING MOBILE AGENTS

Traceback system developed is an improvement to work presented by [1] and [2]. The traceback would cover LAN, Inter-connected networks, and WAN. The source of DoS attack would be traced back even if attacker uses spoofed IP address and creates flooding attacks. This is achieved by using datalink-level identifier (Mac address) and packet timestamps to identify next hop in trace path. Mobile agent technology has been used to implement the traceback system, mainly IBM Java Aglets agent system. For tracing to work successfully, agent platform must be deployed on all domains in WAN. This would enable agents to be dispatched and execute successfully on different LANs. An agent would fail to be dispatched to a LAN that doesn't have the agent platform installed on to it.

8.1 Architecture

The architecture is similar to that presented by [1]. The system is composed of Manager, Sensor, and Tracing agents. There is also an application implemented using Java Packet Capture (JPCap) library [7] designed to capture packets going in and out of a LAN. It works along with the sensor agent in detecting an attack. The system is based on having different domains. Each domain will have a machine acting as the manager of that domain. This manager machine is also called an Agent Station. The Agent station of each domain will have a sensor agent and Java Packet Capture tool running on it. They work together to detect an attack. All other machines in the domain will have sensor agents running and trying to detect attacks too. Only an Agent Station can create tracing agents and send messages to other Agent Stations in different domains. Each Agent Station has a file called Ether_IP.txt. This file contains information (IP address and Mac Addresses) of the following:

- Machines in the domain.
- Routers connecting the domain to other domains in Inter-connected networks.
- Agent Stations in different domains that are connected to those routers.

Based on such information, the tracing agent could identify the next hop. Obviously, it's not feasible to keep all information (MAC Address) of routers and other Agent Stations in a WAN at each individual Agent Station. Thus to extend search in a WAN, each Agent Station keeps information about a remote WAN "controller". This controller acts as a web server, and could be located at different ISPs. The controller would have a bigger list of IP addresses of routers and Agent Stations in a WAN. WAN would contain many controllers. Different LANs could be designed to connect to different controllers. This would require high coordination and management system to update the controllers with new routers and Agent Stations IP addresses. The paper doesn't focus on details of such coordination and management system as it's considered one thing to be investigated in the future work.

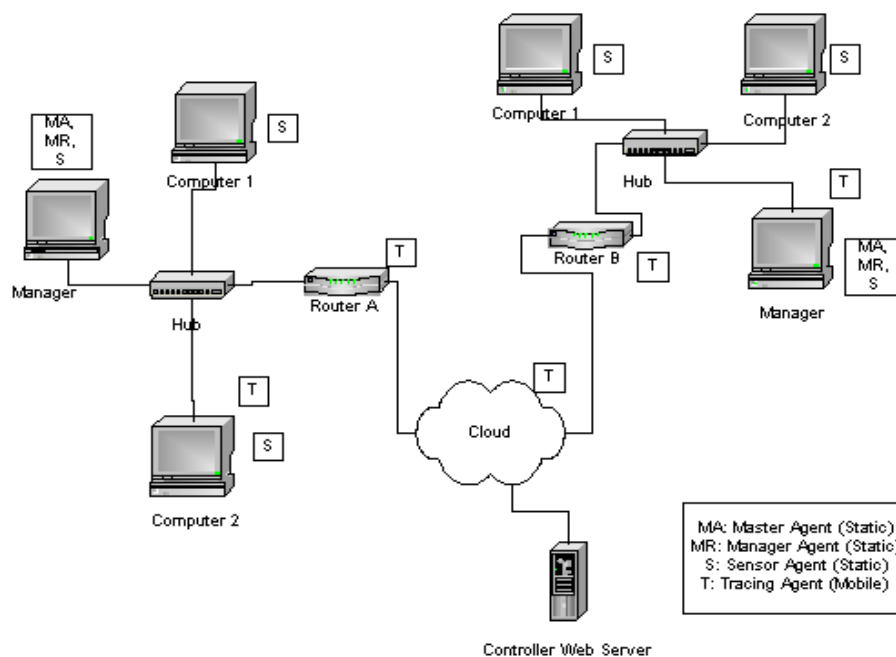


Figure 8.1 shows the traceback system architecture.

8.2 Tracing Technique

The Java Packet Capture tool is running on the Agent Station. It detects an attack if any of the following two situations happen:

- Both the source and destination ports of the attack packet are equal to 800. Port 800 was just chosen for testing purposes. It could have been set with any other port number.
- An attack packet is sent from same source to same victim at the Agent Station's domain for more than 20 times. The number is kept low (only 20) for testing. In real environment it should be kept higher than that.

Once an attack is detected, the Java Packet Capture tool sends a message to sensor agent running on the victim machine specifying that an attack has been detected. The sensor agent will receive the message and logs down the attack packet feature to a file. The sensor agent then sends the manager agent a message to start a new tracing process. It also sends the manager the attack packet features. The manager agent creates a new tracing agent and dispatches it to the victim. The tracing agent compares the source Mac of the attack packet against Mac addresses found at the Agent Station Ether_IP.txt file. The result of the Mac address search is one of the following:

- Mac address is found to be of a local machine to the domain.
- Mac address is found to be of a router in an Inter-connected network and has a list of Agent Stations connected to it.
- Mac address isn't found in the list and considered to be of a router in the WAN. In this case controller must be contacted.

Accordingly, the tracing agent action would be as follows:

- If the Mac address is a local one, then the tracing agent will report to the manager the attacker is found locally and the tracing process will stop.
- If the Mac address is that of a router within the inter-connected network, the tracing agent will retrieve all Agent Stations (at different domains) connected to that router. It will visit each Agent Station, to examine log files for traces of the attack packet. The tracing agent will create list of all Agent Stations it has visited and had traces of the packets in their logs. The list will be sorted in descending order based on timestamp of the attack packets logged. The tracing agent will send this list to the manager agent of its home domain, requesting it to start new tracing process at these different domains. The manager will send a message to first Agent Station in the list to start a new tracing process. The manager will send that remote Agent Station the attack packet features too. The manager

will wait for results from the remote Agent Station. The result could be one of the following:

- Attacker is found at that agent station domain.
 - Attacker isn't found but new suspected domains are identified with new list of Agent Stations.
 - Attacker isn't identified, and there aren't any new agent stations lists.
- If the attacker is found then the overall tracing process would be stopped. If the attacker isn't found and there are new agent stations identified, then this new Agent Station list will be appended to the front of the existing list at the manager. The manager will retrieve next Agent Station in the list and will send a request for it to start a new tracing process. If the attacker isn't identified and there are no any new suspected domains, then the next Agent Station in the existing list will be retrieved. The manager will send it a message to start a new tracing process and will wait for the result.
This tracing process is repeated until attacker is identified or the list becomes empty.
 - If the Mac address isn't found in the Ether_IP.txt file. Then it must be of a router that is considered to be linking the domain to WAN. Accordingly the tracing agent will contact main controller web server identified for that domain. The controller will have list of routers IP addresses and peer Agent Stations IP addresses. The tracing agent will send the list of all Agent Stations provided, to the manager requesting it to start new tracing process on all these remote Agent Stations domains. The manager will send tracing request to all Agent Stations along with the packet feature and will wait for the results.

9. EXPERIMENTS

Several tests were conducted using the American University in Cairo's network. The traceback covered different domains at the university. Tests have proved the ability of the implemented traceback system to identify actual attacker or at least closest point to the attacker. The attacker would create DoS flooding attack using spoofed IP address. Below is one of the conducted experiments and results. More experiments are described in thesis work.

9.1 Overview

The experiment is based on tracing back an attacker within three different domains. The attacker resides in one domain, while two victims reside on the two other domains. In this experiment, the attacker launches attack to two different domains. The two victims start a trace process until the actual attacker IP address is revealed. The experiment is divided in to two tests. The first test is conducted using an IP spoofing attack tool (check Appendix A.1). The second test is conducted using a flooding attack tool (check Appendix A.2). Time statistics is obtained for both tests. The tracing process executes while the attack is taking place.

9.2 Network Topology

The network topology implemented to conduct the test involves three domains.

9.2.1 First Domain – Attacker Domain

The domain is 172.25.3. It has two machines connected to a Hub.

- The attacker machine:

IP Address = 172.25.3.71

Mac Address = 00:08:74:e6:e1:a7

- The other machine is the Agent Station of the domain:

IP address = 172.25.3.70.

Mac Address = 00:03:47:a2:37:44

The Aglet Server is running on the Agent Station as well as the java packet capture application.

9.2.2 Second Domain – Victim Domain

The domain is 172.16.244. It has two machines.

- The victim machine:

IP Address = 172.16.244.21.

Mac Address = 00:03:47:29:a8:ff

- The other machine is the Agent Station of the domain:

IP Address = 172.16.244.20.

Mac Address = 00:03:47:2a:71:1c

Both machines have the Aglet Server running. The Agent Station has Java Packet Capture application running too.

9.2.3 Third Domain – Intermediate Domain

The domain is 172.25.5. It has one machine.

The machine is an Agent Station and victim of the domain:

IP Address = 172.25.5.201.

Mac Address = 00:03:47:a2:13:58

It has both Aglet Server and Java Packet Capture application running.

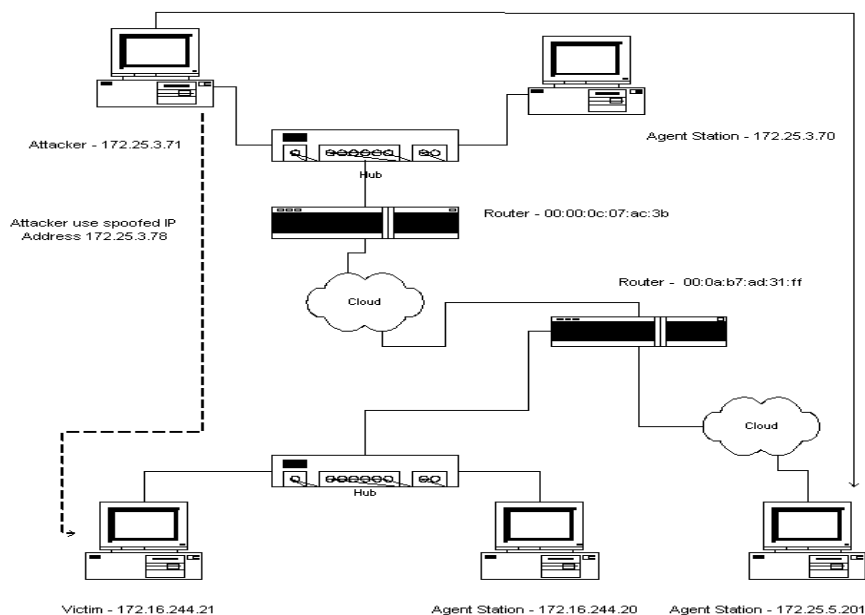


Figure 9.1: Experiment Network Topology

9.3 Attack/Trace Process Description

1. Attacker with IP address 172.25.3.71 starts an attack targeted to two victims with IP address 172.16.244.21 and 172.25.5.201. The attacker uses spoofed IP address 172.25.3.78. The attack packet feature is as follows:

```
1055578596:307462 172.25.3.78->172.16.244.21
protocol(6) priority(0) hop(46) offset(0)
ident(242) jpcap.EthernetPacket@50bd4d
00:0a:b7:a8:61:ff->00:03:47:29:a8:ff (2048)
```

2. Tracing process for attack towards 172.16.244.21
 - Java Packet Capture residing on the Agent Station 172.16.244.20 captures the network packets. It captures the attack.
 - The Agent station Java Packet Capture application sends an attack detection message to the sensor agent running on the victim 172.16.244.21. It also writes the packet features in a threshold file in c:\thersholds folder on the victim's machine.
 - The sensor agent sends a request to the manager agent at 172.16.244.20 to start a new trace process.
 - The manager agent creates a new tracing agent and sends it to the victim 172.16.244.21.
 - Tracing agent will read the threshold file and extract source Mac address of attack packet.
 - Tracing agent will compare the source Mac address to Mac addresses found in Ether_IP.txt file. The Mac is found to be Mac of a router.
 - The tracing agent will extract the IP addresses of all Agent Stations linked to that router. In this test its two Agent Stations, 172.25.3.70 and 172.25.5.201.
 - The tracing agent will dispatch itself to the second domain having Agent Station 172.25.3.70.
 - The tracing agent arrives at Agent Station and checks logs created in c:\logs folder for packet traces.
 - The tracing agent finds traces for the attack packet on 172.25.3.70.
 - The tracing agent will dispatch itself to the third domain having Agent Station 172.25.5.201.
 - The tracing agent arrives at Agent Station and checks logs created in c:\logs folder for packet traces.
 - The tracing agent doesn't find traces for the attack packet on 172.25.5.201.
 - The tracing agent retracts back to its home domain Agent Station 172.16.244.20 and sends the manager agent a request to create a new trace process. It informs its manager about the remote domain where new tracing process should be created: 172.25.3.70.
 - The tracing agent stops tracing.
 - The manager at Agent Station 172.16.244.20 sends a message to manager at 172.25.3.70 requesting it to start a new trace process for the attack packet. It sends the manager at 172.25.3.70 the attack packet features.
 - The manager at 172.25.3.70 writes the packet to folders c:\PacketsVisited and creates a new tracing agent.

- The tracing agent reads the packet features from file created at the c:\Packets Visited folder. It compares the attack packet features against that found in the logs folder.
 - The packet is found in the logs saved.
 - The source Mac is retrieved from the logs and compared against those found in the Ether_IP.txt file.
 - Tracing agent finds that Mac address corresponds to Mac address of a machine in the domain with IP address 172.25.3.71.
 - The tracing agent sends message to manager agent at 172.25.3.70 specifying that the attacker is found at IP address 172.25.3.71.
 - The manager agent at 172.25.3.70 sends a message to manager agent at 172.16.244.20 specifying that the attacker is found at IP Address 172.25.3.71
 - The tracing process stops and manager at 172.16.244.20 writes the result.
3. Tracing process for attack towards 172.25.5.201
- Java Packet Capture residing on the Agent Station 172.25.5.201 captures the network packets. It captures the attack.
 - The Agent station Java Packet Capture application sends an attack detection message to the sensor agent running on the victim. It also writes the packet features in a threshold file in c:\thersholds folder.
 - The sensor agent sends a request to the manager agent to start a new trace process.
 - The manager agent creates a new tracing agent and sends it to the victim 172.25.5.201.
 - Tracing agent will read the threshold file and extract source Mac address of attack packet.
 - Tracing agent will compare the source Mac address to Mac addresses found in Ether_IP.txt file. The Mac is found to be Mac of a router.
 - The tracing agent will extract the IP addresses of all Agent Stations linked to the router. In this test it's only one 172.25.3.70.
 - The tracing agent will dispatch itself to the other domain having Agent Station 172.25.3.70.
 - The tracing agent arrives at Agent Station and checks logs created in c:\logs folder for packet traces.
 - The tracing agent finds traces for the attack packet on 172.25.3.70.
 - The tracing agent retracts back to its home domain Agent Station 172.25.5.201 and sends the manager agent a request to create a new trace process in other domain.
 - The tracing agent stops tracing.
 - The manager at Agent Station 172.25.5.201 sends a message to manager at 172.25.3.70 requesting it to start a new trace process for

the attack packet. It sends the manager at 172.25.3.70 the attack packet features.

- The manager at 172.25.3.70 writes the packet to folders c:\PacketsVisited and creates a new tracing agent.
- The tracing agent reads the packet features from file created at the c:\Packets Visited folder. It compares the attack packet features against those found in the logs folder.
- The packet is found in the logs saved.
- The source Mac is retrieved from the logs and compared against those found in the Ether_IP.txt file.
- Tracing agent finds that Mac address corresponds to Mac address of a machine in the domain with IP address 172.25.3.71.
- The tracing agent sends message to manager agent at 172.25.3.70 specifying that the attacker is found at IP address 172.25.3.71.
- The manager agent at 172.25.3.70 sends a message to manager agent at 172.25.5.201 specifying that the attacker is found at IP Address 172.25.3.71.
- The tracing process stops and manager at 172.25.5.201 writes the result.

9.4 Time Statistics

9.4.1 Attacker Using IP Spoofing Tool

The network load was kept low during the IP spoofing attack. It was kept at maximum of 300 kbps.

Total Trace Time at 172.16.244.20	
Start Time	12:07:39:256
End Time	12:08:17:817
Total Trace Time at 172.25.5.201	
Start Time	12:06:22:205
End Time	12:08:00:926

9.4.2 Attacker Using Flooding Tool

The network load was kept high. The network load at 172.16.244.21 was 700 Kbps and at 172.25.5.201 was 1.3 Mbps.

Total Trace Time at 172.16.244.20	
Start Time	02:48:02:306
End Time	02:49:01:212

Total Trace Time at 172.25.5.201	
Start Time	03:48:22:660
End Time	03:49:07:294

10. CONCLUSION AND FUTURE WORK

The improved technique described in this paper has succeeded in tracing back and identifying an attacker under the following two conditions: attacker spoofs his IP address and attacker floods the network. It could traceback an attacker in LAN, Inter-connected networks, and WAN. Using mobile agents have made the system more attack resistant than distributed architecture described in [2]. If part of agent system gets corrupted, the rest can function properly. Also, during traceback process, huge amount of data must be analyzed. Mobile agents are usually of smaller size than the data to be analyzed. Thus, it's easier and more efficient to transfer agents to data during network congestion caused by flooding attack, than transferring data to computation. This would help in solving network latency problem. Another very important feature of the developed traceback system is its ability to traceback an attacker while an attack is taking place or even after it has been completed.

The implemented traceback system could spawn heterogeneous systems as it is coded using java agents. It doesn't require any special hardware. It also, doesn't require any changes in TCP/IP protocol. Only agent system and platform need to be deployed in each LAN as described above. The traceback process doesn't involve any costly router computation.

On the other hand, there are several limitations to the proposed technique that must be taken in to consideration in the future work. The limitations are as follows:

- Tests were conducted on The American University in Cairo's network. Due to limited resources and permissions, tests were conducted over network of three inter-connected LANs. In order to extend the experiments, a single LAN was used to simulate an environment of multiple LANs (i.e. a LAN was divided in to several LANs by having more than one independent Agent Station and separate traceback systems). Experiments could be extended to cover larger real network topologies across WAN.
- The traceback system must be deployed on almost every LAN in WAN. Otherwise, tracing agent will not be able to function properly.
- Timeout mechanism must be implemented at two levels: timeout of messages, and timeout of tracing agent.

- Security of mobile agents must be taken into consideration. There are several mobile agent security methods that can be used and are presented in [9].
- Having efficient packet logging system is important to make sure that data doesn't get lost. At time of a flooding attack, logs are kept in huge amount and some data may not be examined or even get lost. Efficient logging mechanism is required to make sure that the tracing agent could succeed in examining all data properly for finding actual attacker. In the above mentioned experiment, packet logging has been considered a problem. At high flooding attack, data was lost and sometimes the experiment failed in identifying the actual attacker (i.e. just stopped at closest point to attacker).
- Although MAC address spoofing isn't currently a common act, still it should be taken in to consideration for future work.
- Implement controllers' coordination and management system.

11. REFERENCES

- [1] Asaka Midori, Shunji Okazawa, and Atsushi Taguchi(June 1999), A Method of Tracing Intruders by Use of Mobile Agents. Proc. INET 99.
- [2] Baba, Tatsuya and Shigeyuki Matsuda (April 2002), Tracing Network Attacks To Their Sources. NTT Data Corporation, IEEE.
- [3] Lau, Felix, Staurt Rubin, Michael Smith and Ljiljana Trajkovic. Distributed Denial of Service Attacks.
- Aglets Specification 1.1 Draft. IBM Corporation, 1998.
- [4] CERT Advisory(1996), UDP Port Denial-of-Service Attacks.
<http://www.cert.org/advisories/CA-1996-01.html>
- [5] CERT Advisory(1996), TCP SYN Flooding and IP Spoofing Attacks.
<http://www.cert.org/advisories/CA-1996-21.html>
- [6] JPCap(2003), Java Package for Packet Capture.
<http://netresearch.ics.uci.edu/kfujii/jpcap/doc/index.html>
- [7] Savage, Stefan, David Wetherall, Anna Kerlin, and Tom Anderson.(2000), Practical Network Support for IP Traceback. University of Washington,.
- [8] Jansen, Wayne and Tom Karygiannis(1999), Mobile Agent Security. National Institute of Technology.

[9] Libnet. Libnet Packet Assembly. www.packetfactory.net/projects/libnet/

[10] Misoskian Packet Builder. Attack tools and Protection.
<http://www.angelfire.com/my/bulat/download.html>

INTELLIGENT AND MOBILE AGENT FOR INTRUSION DETECTION SYSTEM : IMA-IDS

Farah Barika farah.barika@ragingbull
Nabil El Kadhi el-kad_n@epitech.net
Khaled Ghedira khaled.ghedira@isg.rnu.tn
Laboratoire SOIIE, Tunisie & Laboratoire LERIA, France.

Abstract: Over the years computer systems have successfully evolved from centralized monolithic computing devises supporting static applications into distributed computing called Networks, therefore our systems are becoming more open and subject to set of security threats. Thus, a key challenge is to provide all computer systems with the appropriate mechanisms to offer security services such as authentication, secret preservation and automatic attacks detection commonly known as Intrusion Detection. Intrusion Detection Systems (IDS) are used to discover several kinds of attacks. Commercial solutions are generally centralized and suffer from significant limitations when used in high-speed networks. This is one of our major motivations to use the distributed model based on mobile agent platforms. We believe that agent characteristics will help collecting efficient and useful information for IDS. Thus, in this paper, we propose an Intelligent Mobile Agent model for distributed IDS called IMA-IDS. Before introducing our global Intelligent Mobile Agent IDS architecture, we will first argue for the use of mobile agents in IDS and then, we will choose an agent platform which can offer security mechanisms needed by IDS solutions. Last we will demonstrate the feasibility of our model with a working prototype. The result obtained from the implementation of our IMA-IDS is a proof of the suitability of using Agent concept for IDS architecture.

Keywords: Network, Security, Intrusion detection system, Distributed intrusions detection, Intelligent mobile agent.

1. INTRODUCTION

Security is crucial to the success of active networking especially when the current network is characterized by a dynamic nature and an increasing distribution. Traditional network relies on security mechanisms and policies deployed on the underlying operating system. Nevertheless, these measures are insufficient and they present, in general, a set of flaws that result in security vulnerabilities [CLM + 99]. The field of automated computer security intrusion detection gives result to Intrusion Detection System (IDS for short). The goal of IDS is to analyse events on the network and identify manifestations of attacks. Commercial solutions are generally centralized and suffer from significant limitations when used in high-speed networks. The identification of distributed intrusions requires cooperation of different sensors so it is advisable to consider mobile devices as a challenge to intrusion detection. Our motivation is to distribute intrusion detection using mobile and intelligent agents. This paper is organized as follows. This is the introduction, section 2 presents the most important features of existing IDS solutions and their limitations. In section 3 we argue for the use of mobile agent. Section 4 compares the most used agent platforms and concludes by choosing Aglets platform [DM98] that offer the most interesting security features for our solution as presented in [EKBBGe03]. Finally section 5 describes our IMA-IDS by presenting its architecture and the actual prototype implementation.

2. BACKGROUNDS IN IDS

Intrusion Detection Systems (IDS) plays an important role in achieving survivability of information system and preserving their safety from attacks [VEK]. Attacks against a system are informally defined as a deliberate attempt to violate the security policy. Intrusion detection has been achieved by following two different strategies of analysis [MHL94]:

- *Anomaly detection*: relies on models of "normal" behaviours of a computer system. Behaviour profiles may be focused on users, applications or networks. Anomaly detection compares the defined profiles against the actual usage patterns to detect "abnormal" activity patterns. These patterns will be considered as intrusions.
- *Policy detection*: relies on a set of attack descriptions called attack-signatures. Both anomaly and policy detection present advantages and disadvantages. In fact, policy detection is limited by attack definitions. Anomaly detection permit

detection of previously unknown attacks; this advantage causes a large number of false positive occurring when an IDS sends an alarm for an event that is not an intrusion [VCF02]. Commercial IDS products such as NetRanger [httpb], RealSecure [iteB7] and Omniguard Intruder Alert [httpa] are in general based on policy detection.

IDS are usually classified as network-based (NIDS) or host-based (HIDS) IDS [MHL94]. The most important difference between these two IDS categories is the fact that NIDS rely on information obtained by monitoring the network, while the HIDS perform their analysis on information collected at a single host. Since HIDS works above the network layer, it is unable to detect some kinds of attacks [Ran01]. Otherwise, NIDS infer their decision from low-level network packets travelling among hosts [HDL + 90] and are generally able to detect such attacks.

2.1 IDS Requirements

In [JMKM99], authors have defined a set of desirable characteristics for an IDS by focusing two themes : functional and performance requirements. In the following section, we summarize some of these characteristics.

2.1.1 Functional Requirements

- IDS must continuously monitor and report intrusion.
- IDS should have a very low false alarm rate.
- IDS should provide enough information to repair the system in the case of detection of intrusion. Notice that this characteristic depend on IDS goals. In fact, many IDS solutions focus only on alerting administrators without suggesting any corrective actions.
- IDS must detect and react to distributed and coordinated attacks. This detection feature is one of the most difficult because it needs a huge distributed amount of information in addition to the hard task of synchronisation between different hosts.
- The IDS should be adaptive to network topology and configuration changes.

2.1.2 Performance Requirements

- Intrusion should be detected in real-time as it should be reported immediately in order to minimize network damage.
- The IDS must be scalable in order to handle additional computational and communication loads.

2.2 IDS Limitations

The most common IDS shortcomings include the following :

- High number of false positives.
- Lack of efficiency : usually when an IDS is faced with a very large number of events in the network, it slows down a system or drops network packets.
- Vulnerability to attacks : many IDS have hierarchical structures. This fact gives attackers the opportunity to harm the IDS by cutting off a control branch or even by tacking out the root command.

In addition to the forementioned shortcomings, our solution aims to overcome the following limitations :

- Many of the existing network and host based IDS perform data collection and data analysis essentially by using a monolithic architecture [BGFI + 98]. The centralized detection scheme suffer from a number of problems :
 - A central analyser presents a favourable target to attackers. If an intruder manages to decapitate it, the entire network loses protection.
 - A high network load leads to excessive data traffic, the system suffers from scalability problems. A single analyser unit limits the network size.
 - Since network data collection is performed in a host different than the one in which analysis is performed, intruders can perform insertion and evasion attacks [PN98].

Intrusions can be conducted through several steps that occur at different hosts, and consequently cannot be detected by a single sensor. The cooperation of different sensors becomes an imperative for the identification of distributed intrusions. Thus, mobile devices offer a new approach to IDS implementation. Typically, mobile agent technology can solve the set of the shortcomings mentioned above.

3. USEFUL CHARACTERISTICS OF MOBILE AGENTS

Agent Systems are used in various applications such as workflow, scheduling and optimisation [Ghe93]. It is advisable to define what is an agent. We refer to [Pal98] :

- An agent is a physical or a logical entity characterized by the following attributes :

- Autonomy : agents are independently running entities, they operate (in ideal cases) without human control.
- Mobility : agents are able to suspend processing on one platform and to move to another one where they resume execution.
- Rationality: agents embody the capacity to analyse and solve a problem in a rational manner.
- Reactivity: agents perceive their environment and adapt their behavior in a dynamic way to match, as soon as possible, new environment parameters.
- Inferential capability: agents are able to share a set of knowledge in order to achieve a specific goal.
- Pro-activeness: agents can decide to adapt their behaviour to their environment.
- Social ability: agents are able to meet and interact with other agents. The interaction and collaboration between agents is achieved by an agent communication language and it may depend on an on-tology.

Accordingly to the previous attributes, we will argue for the use of Mobile Agent to improve the characteristics of the IDS and to overcome the limitations described in section 2.2.

- Reducing Network Load : Existing IDS are faced with the problem of performing a huge amount of data over transfer. Abstracted forms of this data are usually sent from all locations in the network to the central site in order to be processed. Sending a huge amount of data causes an increase of a network loads. Mobile agents offer the opportunity to overcome this problem by eliminating the need of so much data transfer. The processing program (agent) can be dispatched to the host containing crucial data. This will reduce network traffic since an agent is smaller than the processed data.
- Overcoming Network Latency: Mobile agents are able to dispatch from a host to carry out operations directly to the remote point of interest, thus agent scans provide an appropriate response faster than a hierarchical IDS that has to communicate with a central coordinator based elsewhere on the network.
- Asynchronous Execution and Autonomy: Agents can be stopped and started without disturbing the rest of the IDS. Notice that the mobile agents are able to continue to operate autonomously even if the host platform where it was created is not available or is disconnected from the network. Mobile agent frameworks provide IDS with the possibility of continuing to work even when a central controller is down.

- **Dynamic Adaptation:** Mobile agents can be retracted, cloned, dispatched, killed or put to sleep as network's configuration; topology and traffic characteristics change over time. As the number of nodes in the network increases, agents can be cloned and dispatched to these new computing elements.
- **Robust Behaviour:** Mobile agents have the ability to react dynamically to security conditions making it easier to build robust distributed systems.
- **Scalability:** Distributed mobile agents IDS are one of several options that allow computational load and diagnostic responsibilities to be distributed throughout the network [JMKM99]. This improves scalability and maintains fault-resistance behaviour.

4. RELATED WORK

The idea of distributing the intrusion detection system using agent software is not entirely new. However, most of the related work emphasize static agents instead of mobiles ones. Applying mobile agent technology to IDS gives results to only a few research projects. In 1999, a project at The Information-Technology Promotion Agency (IPA) in Japan involved an Intrusion Detection Agent (IDA) System [AOTG99]. IDA is a classic host-based system that relies on mobile agents mainly to trace intruders among the various hosts involved in an intrusion. In the same year the project Micael [DQDCP99] pursued a more ambitious aim where the entire system is based on mobile agents. Nevertheless, only the architecture description has been presented and no details have followed so far. In 2000, an IDS framework based on mobile agents has been described in [BDSM00]. Unfortunately, detection is dealt with superficially. In 2002, [TAP + 02] describes an IDS designed as mobile application that roams the network to detect attacks and track intruders. IMA-IDS is a distributed intrusion detection system using mobile and intelligent agents. It is too tedious to detect a malicious action through the network, especially when considering multiple distributed events and even simultaneous one. Most of the current IDS assume that the environment is static, whereas in reality the environment is dynamic and unpredictable. So to detect, without mistakes, an intrusion and to make the appropriate decision at an optimal time a cooperation of different sensors and analysers is required. Thus, to apply agent mobility to avoid shortcomings of current IDS.

5. OUR SYSTEM : IMA-IDS

In this paper, we advocate the idea that in the security domain, especially when we are faced to the contemporary computer distributed environment, a mobile agents framework enhances the performance of IDS and even offers it new capabilities. We argue this point of view by explaining our system called IMA-IDS (Intelligent Mobile Agents for Intrusion Detection System). IMA-IDS main purpose is to achieve in automatically and real time the intrusion detection by mobile, intelligent and cooperative entities which are the agents.

5.1. Architecture Overview

In this section, we will introduce IMA-IDS architecture by recalling the main important properties to be verified by almost all IMA-IDS components:

- Information collection and filtering: Agents have to guarantee that collected information is of a good quality. The most important question here is how to evaluate information quality. We believe that information can be classified by:
- Information relevance: does the agent collect the important and useful information for the requested work? Is there any guarantee of information correctness? formal agent behaviour proof can be used here to ensure these properties.
- Information / event indication: remember that IDS are of two kinds: On line and post analysis IDS. In each case, agents must be able to report, as soon as necessary, any significant event for intrusion detection. We also believe that we need to define a kind of database rules that will allow agents to decide whether or not to report an event and to which agent (typically analyser agent) should this information be pointed up.
- *Information trust level*: one of the major drawbacks of actual IDS (signature or behavioural IDS) is positive and negative errors or missed attack signalisation. Ensuring that analysis agents have the appropriate information with all correlated events for analysis will help to reduce such errors. By assigning a trust level to each event depending on agent source, any Analyser agent can, in addition, decide whether or not to take in account such information.
- *Agent communication protocol*: in addition to the classic agent communication models (authentication, private or public channel), a specific protocol schema for crucial information communication will be introduced. In fact, suppose that an agent A asks for crucial information (M) possessed by the agent B. Suppose also that agent B

classifies this information as Top Secret so he asks for a hard cipher channel communication for M. If agent A considers M as just secret or public, agent B can even refuse communicating M to A, oblige A to upgrade to top secret class or communicate M to A and forbid any further communication of M by A. As presented by figure 1, IMA-IDS includes, in addition to the manager agent, three agents' categories :

- *Collector agent*: This kind of agent will be cloned and distributed throughout the network. This agent patrols the network and collects all the events occurring in the host to which it is related. Notice that the goal is to have specialized collector agent. The idea is that the collector agent will be interested in a set of event categories. So many collector agents can run on the same host, depending on applied analysis. It is also possible to merge collector agent abilities in to a single agent.
- *Correlator agent*: This is a particular agent that will hurry the specific information, called critical, and send it to the appropriate analyser agent without passing through the manager agent. The default communication protocol (presented later) is centralized. It supposes that a collector sends a kind of report to the manager agent. The manager will decide whether to dispatch data to the analyser or not. This communication model is inefficient for online detection since some crucial events must be taken in to account by the analyser as soon as they occur. That is why each correlator agent will use a set of rules that clearly specifies the crucial events, contexts and analyser agents concerned by an urgent reporting event mechanism.
- *Analyser agent*: Analyser agents are the engines of our solution. Several kinds of analysis such as classical signature detection, anomaly detection and a new security protocol analysis based on abstract interpretation as introduced in [EKOBY03] will be integrated. We are also working a be-haviour analyser that will use a kind of statistical model to define what can be concerned as "normal" system behaviour is also being developed.
- *Manager agent*: This agent gathers collected information and distributes it to analyser agents. This communication process does not allow online analysis. For this reason correlator agents can decide to communicate directly with analyser agents.

The administrator can distribute our IMA-IDS over any number of hosts in the network. Each host can receive any number of collector agents that monitor all events occurring in the host. All the collector agents report their results to the manager agent which transmits them to the analyser agents. The intelligence in our approach is attempted by communicating the critical events detected by the collector agents to the correlator agents. Notice that a critical event is any event liable to be part of a scenario of attack. The

correlator agents take charge of hurrying the critical events received from the collector agents and transmitting them to the concerned analyser agents. The analyser agents receive the events.

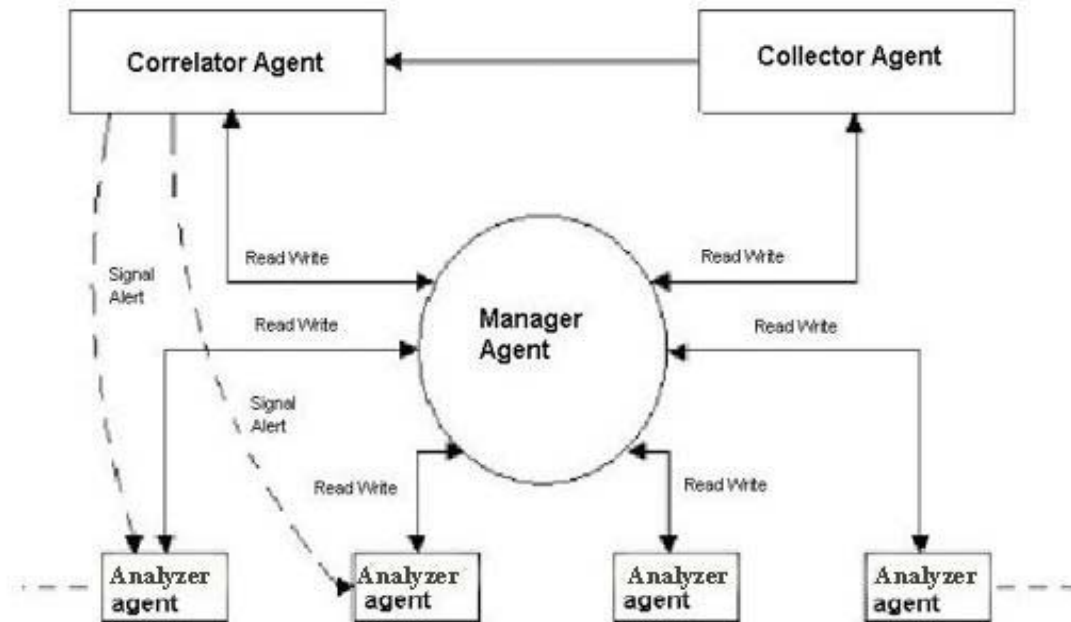


Figure IIMA-IDS Architecture

From the manager agent and those from the correlator agent, perform a higher-level analysis and correlation [EKOBY03] (Anomaly and Policy detection). The analyzer agents report their results to the manager, and they generate alarms if they detect any anomaly. Agents in our system are cooperative because they respond to requests for event or critical events from other agents.

The administrator, according to its needs and via the external interface of the manager agent can stop the execution of these agents, send them to other locations and reactivate them. An agent in IMA-IDS can make a decision to dispatch itself. In others words, it can stop its execution, move to another location and restart its execution. Also, it can clone itself especially in the case of increased network loads.

5.2. Communication Mechanism

The transmission of messages between agents is a central part of the functionality of our IMA-IDS. In order to communicate, agents in our system are able to know all information about the other agents created and running in the net-work (their locations, their number, and their identifier) by sending a request to the manager agent. To offer these capacities, and to be able to update the agent list, the manager uses the following two agents:

- Registry Agent: being present on all hosts running agents, it maintains information about the agents running in the host.
 - IdsHost Agent: it keeps track of all created and running agents.
- Agent communication can be divided into two categories :
- Peer-to-peer communication (Monocast): The message sender must know the identifier of the receiver to be able to send a message.
 - Indirect communication (Multicast): a kind of a meeting between agents coming from different hosts. The basic idea is that agents subscribe to one or more multicast message list and implement handlers for these messages. Multicasting message provides a powerful way for agent interaction and collaboration [DM98].

5.3. Prototypical Implementation

To demonstrate the feasibility of our architecture a working prototype has been implemented. Our implementation is written using the IBM Aglet [DM98] platform and Sun's Java Development Kit. This choice is not arbitrary, it is made after a comparison study of nine agent platforms [EKBBGe03] during which the security mechanisms deployed by each platform to protect agents has been scrutinized. This comparison is summarized figure 2.

- Integrity: corruption of information.
- Denial of service: effecting availability of the host or the agent process.
- Secrecy: disclosure of information.

Table 2: Security features agent platform comparison

	Authentication	Access Control	Encryption	Code Verif.
Concordia	Simple Auth.	J.S.M	SSL	ByteCode
JADE	Simple Auth.	J.S.M	SSL	ByteCode
Aglet	MAC	J.S.M	Java	ByteCode
Voyager	Auth. Server	J.S.M	No	ByteCode
AgentTCL	PGP	RSA System	RSA PGP	SafeTCL
MAP	PGP	RSA	Weak + MD5	Secured Scheme
JATLite	Simple Auth.	J.S.M	SSL	ByteCode
TACOMA	PGP	Firewall	PGP	Firewall
Grasshopper	X509	J.S.M	SSL	ByteCode

Let us consider the most important security feature for our IMA-IDS. In fact, it is important to build our solution with an open platform that allows agent migrations, cloning agents and that support different communication protocols. Those features are common to almost all agent platforms. What seemed to be most crucial are security features. The following criteria are the main focus:

- Agent authentication: Trusted and untrusted Agent authentication systems.
- Resource Access Control: Implementing or not a discretionary access control.
- Supporting encryption/decryption facilities: Using standard protocol such as SSL and supporting commonly used algorithm (DES, AES, IDEA, RSA and so on).
- Code verifier (Agent action Verifier): Including a particular byte code verifier or allowing this kind of add-on easily. We mainly plan to develop a specific Java byte code analysis as in [EKB01] for verifying applet security properties. Another alternative is to use such verifier as introduced by Michiaki Tatsubori [Tat99].

According to figure 2, there is no single best agent platform. Each platform offers some interesting services and features and suffers from some other weakness and limitations. Concordia, for example, seems to have the upper hand in the area of security but unfortunately, its excellent security provisions are not available with an evaluation licence. Thus, the best choice is Aglets.

5.3.1. Security Issues

The open-source nature of the Aglets system, which has been made available through the SourceForge opensource initiative [Kit02], allows us to intercept the adequate security actions performed by Aglets and to log all the

useful information such as the requested operation, its parameters, its outcomes and the aglet identity. Mainly, Aglets support the specifically required security mechanisms. The security model provided by Aglets supports the definition of security policies and describes how and where a secure Aglet system enforces these policies. Aglets have authenticated identities that are used to enforce the policies defined by authorities and to identify the program host or developer. Aglets framework uses an asymmetric (public_private key pair) cryptography system to exchange private keys between hosts. These keys are useful to ensure agent identities when they are transferred over the network. Thus, the agent code is signed and can be authenticated before its execution, keeping the host platform protected. In Aglets, permission is defined as the capabilities of executing aglets by setting access restrictions and limits on resource consumption. An abstract syntax for permissions in Aglets is based on JDK policy [Gon97] file definition.

5.3.2. Our Prototype

The prototype that we are testing is intended as a proof of the concept for the architecture, the security capabilities of the Aglets workbench, the communication model established in our IMA-IDS and the full mobility. The prototype implements the main structure and interaction between the agents, as well as security intended behaviour. This prototype is implemented in JDK 1.4.1_02 and uses the framework Aglets 2.0.2. The prototype has been distributed through a network of more than one hundred machines. One specific machine is used as a manager agent host. Let's recall that the prototype included for agent categories:

- *Manager Agent* : this is the scheduler of all the solutions. It uses a set of initialisation files describing the analysis-concerned domains, the IP addresses of the analysed hosts and also the agent platform domain signature.
- *Collector Agent* : collectors are for the moment composed of two categories. Event simulator agents and KCS sniffers [EKOBY03]. KCS sniffers are in fact implemented in C language and are not embedded in the agent platform because difficulties to capture network traffic in Java. The second kind of collector is, for the moment, an event generator used to simulate event generation through keyboard use as one example.
- *Correlator Agent*: Prototype correlators use empty rules in order to validate message exchanges between correlators and analysers.
- *Analyser Agent*: the prototype includes a signature-based analyser. A property propagation analyser is being developed as well. The initial

experimental results prove that the communication model seems to be well adapted to agent collaboration. In fact, by testing agent platform communication through more than 20 hosts, we didn't notice any information loss or delivery delay. Such results must also be validated by larger experimentation. The prototype was particularly efficient in defining an agent security policy. In fact, by creating some collector agent behind the authorised domain, we have been able to test the Aglet firewall functionality. Intrusion detection mechanisms can be not evaluated for the moment since analysis techniques are under development. We wish to have a complete evaluation platform in 3 or 4 months. Analysis models are based on compartmental analysis and statistical functions. The prototype includes an event generator and a set of rules to simulate correlator behaviour. During the actual experimentation, we notice that a set of conflicted rules can be a drain on correlator efficiency. In fact, we should add a set of filtering rules, as used in expert systems, in order to add a conflict solving phase when deciding which rule to be applied. For signature-based analysers, we believe that they will be no specific difficulties to define such rules that can totally rely on signature definitions. For property propagation, the use of abstract semantics as in [EK01] is ambiguous. Special care must be used to abstract parameter definitions. The compartmental analyser is more complicated. This is a work in progress and will be developed in a PHD project. For the moment we believe that we first must fix a set of "measurable" parameters that can be used as correct behaviour definition base. Opened communication port, segmentation messages, network charge are examples of useful parameters. The result obtained by the KCS project [EK03] will be used as an initial set of input to analyse correct SSH/ SSL sessions in order to extrapolate a set of useful "measurable" parameters. The following figures (3, 4) illustrate the graphic user interfaces related to our IMA-IDS prototype :

- The Manager Agent Interface, it is the main console for controlling and using the prototype. The manager agent uses a set of parameters, described in input files, in order to create and distribute collector and correlator agents in the analysed domain.
- Reception of the events by the analyser Agent from the Correlator Agent, when a signalisation rule is verified, the correlator sends a message to the concerned analyser. The message includes the crucial information for the analysis. In further use, a specific secret communication channel will be created by adding a specific library to the aglet platform.

- Reception of an alert message by the Manager Agent from the Analyser Agent. When detecting an attack in an on line or post analysis model, the analyser sends a message to the manager that can substitute the network administrator by taking some corrective actions. The prototype did not include such abilities for the moment. We are working on defining a set of rules (in a kind of decision tree) to add such intelligent behaviour to our solution.

As we can see it, a specific console is used to show any agent action such as event signalisation or message exchange between analyser and manager.

6. CONCLUSION AND FUTURE WORK

Our purpose by implementing IMA-IDS prototype is to validate agent platform use for IDS. The chosen communication model, as proven by experimentation, offers a flexible and modular agent information exchange. After the validation of the global architecture, we are now working on event signalisation and correlation rules. For the moment, event collectors are based on specific C libraries because of their efficiency. We are studying a set of signature-based IDS to deduce a set of rules to be used by correlator agents. Future work deals mainly with Analyser agents. We aim to study a set of statistical and behaviour models in order to develop a new one for describing a "correct" and an "attack free" system behaviour. We believe that these models will be more efficient when coupled with other analyser such as signature-based systems.

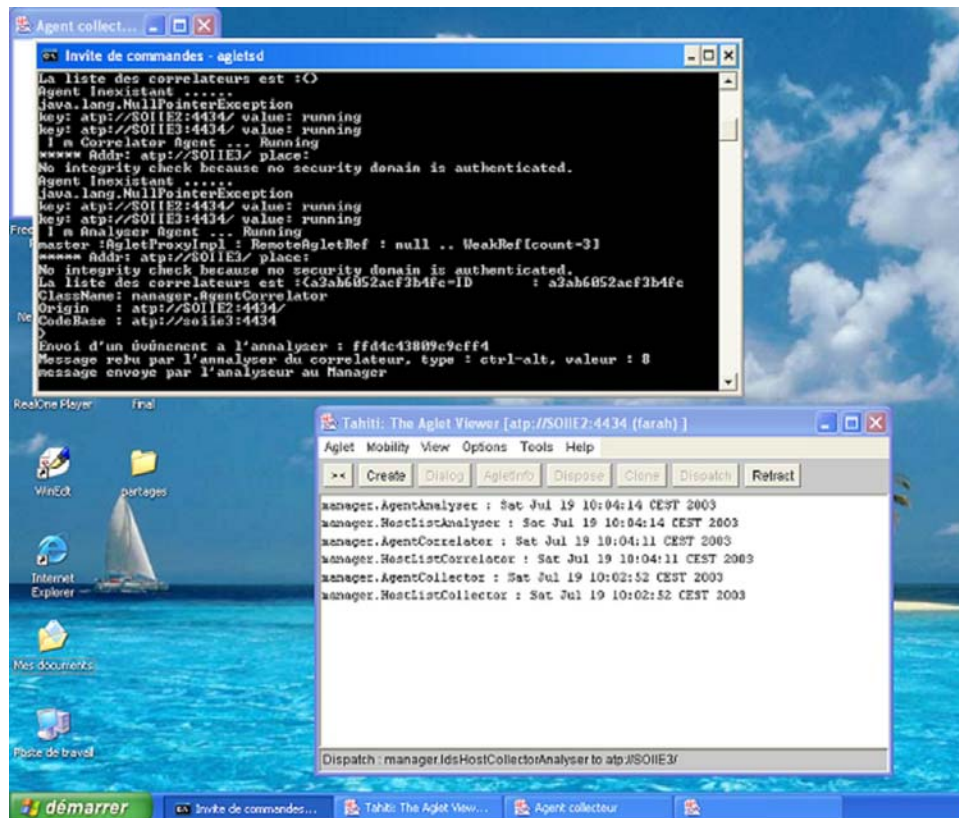


Figure 3: Reception of the events by the Analyser agent from the Correlator Agent

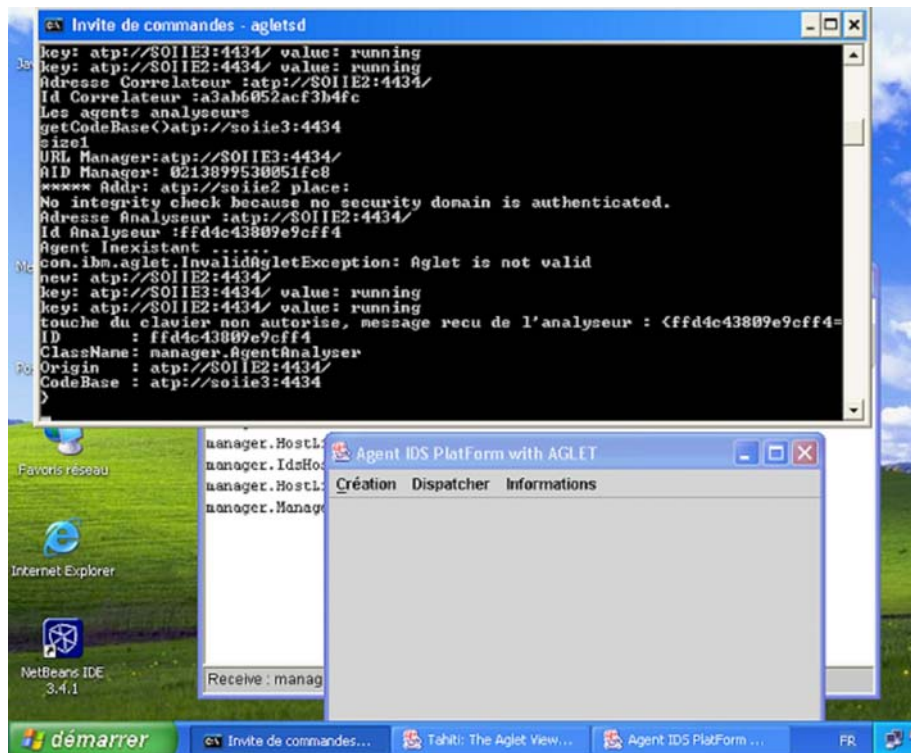


Figure 4: Reception of an alert message by the Manager Agent from the Analyser Agent

7. REFERENCES

- [AOTG99] M. Asaka, S. Okasawa, A. Taguchi, and S. Goto(1999), “A method of tracing intruders by use of mobile agents”. In INET’99.
- [BDSM00] M. C. Bernardes and E. Dos Santos Moreira(2000), “Implementation of an intrusion detection system based on mobile agents”. In International Symposium on Software Engineering for Parallel and Distributed Systems.
- [BGFI + 98] J. S. Balasubramaniam, J. O. Garcia-Fernandez, D. Isacoff, E. Spafford, and D. Zamboni(1998), “An architecture for intrusion detection using autonomous agents”. In Proceedings of the 14th Annual Computer Security Applications Conference.
- [CLM + 99] R.H. Campbell, Z. Liu, M.D. Mickunas, P. Naldurg, and S. Yi. (November 1999), “seraphim: An active security architecture for active network”. Technical report, UIUCDCS-R-99-2167, UILU-ENG-99-1756, Urbana, IL 61801.

[DM98] B. L. Danny and O. Mitsuru (1998), "Programming and Deploying Java Mobile Agents with Aglets". Massachusetts, Addison Wesley, second edition, isbn 0-201-32582-9 edition, 225 p.

[DQDCP99] J. D. De Queiroz, L. F. R. Da Costa, and L. Pirmez. Micael(1999) "An autonomous mobile agent system to protect new generation net-worked application". In 2nd Annual Workshop on Recent Advances in Intrusion Detection.

[EK01] N. EL KADHI (2001) "Automatic verification of confidentiality proper-ties of cryptographic programs".

[EKB01] N. EL Kadhi and P. Boury (2001) "Static analysis of java cryptographic applets". In Proceedings of ECOOP2001 (Budapest) Workshop on Java Formal Verification.

[EKBBGe03] N. EL Kadhi, F. A. Barika, E. Burstein, and K. Ghédira(2003) "Toward agent ids : Agents platforms security features study". In Proceedings of CSC 2003, Computer security Congress, Mexique.

[EKOBY03] N. EL Kadhi, J. Olivain, and J. Ben Younes(2003), "Kcs: A new ssh/ssl protocol analyser for key correlation system detection". In Proceedings of SCI.

[Ghe93] K. Ghédira. MASC (1993) "une approche Multi-Agents de problèmes de Satisfaction de Contraintes". PhD thesis.

[Gon97] L. Gong (July 1997), "java security architecture". Technical report, JavaSoft.

PACKET SEGA, USING STRATEGIC HACKING TO TERRORIZE COMMERCIAL AND GOVERNMENTAL ENTITIES ON THE INTERNET

Khaled M. A. Nassar knassar@nile-online.net

Wael A. Ali wali@nile-online.net

The Egyptian Company for Internet and Digital Infrastructure, Nile Online, Egypt.

Abstract: This research paper objective is to show the massive impact that hostile action like electronic terrorism has on the internet. The paper begins by explaining some of the actors in such action and their motives. Then the paper will present, step by step, the evolution from a simple hacking technique to a strategic technique as a sample of how effective and destructive hacking could be. The paper presents a strategic hacking framework that was integrated and experienced by the research team. This framework shows how information could be used to terrorize the network operations of a company. Moreover, the effect may extend to compromise the organizations business outside its cyber borders. The threat may extend in some cases where national security is compromised. Then the paper will present simple scenarios for implementing such framework in attacking organizations that are different in nature (one is governmental and the other is commercial) as well as the motive that makes someone attacks them. The scenarios will show the impact on these organizations. Afterwards the paper will be concluded and an overview of types of counter measure will take place. Finally the recommendations for enhancing research and development in the field of computer security and increasing the awareness of the community in such field will be presented.

Keywords: E-terrorism, Commercial, Governmental network.

1. INTRODUCTION

The Internet has become one of the largest investments in modern history. The Cyberspace serves many aspects of our modern life. These services include e-government, e-market, and scientific services. The growth rate of the internet is humongous but unfortunately the internet was not designed to secure such a large interactive network of interests. The threats that compromise the internet users, home users as well as organizations, varies in popularity, impact and solutions. However, no entity on the internet is one hundred percent secured.

Electronic terrorism was debated in the past years. But it was not until the 11th of September crisis that it drew huge attention to the possibility of its existence, its motives and counter measures. The purpose of this paper is to discuss the idea of e-terrorism with emphasis on how it takes place, the motives of the e-terrorists, the technical milestones in performing such an action, the possible counter measures, and a theoretical proof of concept of E-terrorism destructive impact on the cyber society.

The paper is divided into four main sections. The first section briefly goes through the various services that the internet provides, the actors of the internet warfare and their motives. The second section concentrates on the intrusion techniques and how simple techniques could propagate to cause terror on the internet. The third section is a proof of concept that intruders can terrorize Internet entities. This section will present scenarios for terrorizing both governmental and business electronic entities to conclude the research assumptions. The fourth section is a short one that will go through the countermeasures as a preface for the recommendations following it.

2. ACTORS, NETWORKS, AND MOTIVES

In this section we will review some of the variant types of networks that could be recognized on the internet. After this brief review we will explain the different reasons that could motivate an intruder to attack a network. And to complete the vision a definition of the actors who play roles in these events will take place as the last part of this section.

2.1. Electronic Entities/Services

Many organizations use the internet to upgrade their services. The internet itself is roughly a communication network. But a researcher could recognize some distinguished domains of services' networks built upon it. Governmental networks, Banks or monetary networks, scientific networks,

and other business networks appear to be most significant. Another important but different in nature networks are the social networks. All these networks can be hacked and all need security measures to decrease the risks that face them.

2.2. Motives

Networks are penetrated for many reasons. An attacker could find a couple of reasons to attack a target. For example, the attacker would want to satisfy his or her ego and talk about an operation he or she performed in public or private groups to gain the respect or admiration of other internet users. But, when he or she attacks it will be the network of an organization of an enemy company or a company that refused his or her employment application.

2.2.1. Hacktivism

Hacktivism can probably best be described as the hacking for political reasons. It's obviously a contraction of hack and activism. The theory is that some hacker will use his skills to forward a political agenda, possibly breaking the law in the process, but it will be justified because of the political cause. An example might be a Web-page defacement of some will-selected site with related message. It might be planning a virus at some company or organization that is viewed as evil. [Ryan, 2000]

2.2.2. Hacker-Nerd Connection

Probably the most widely acknowledged reason for hacking. It seems that a very large number of the hackers out there want some amount of recognition for their work. You can call it a desire for fame, you can call it personal brand building, you can call it trying to be "elite", or even the oft-cited "bragging in a chat room". [Ryan, 2000]

2.2.3. For Knowledge

In a world where a person is recognized by how much he knows it's not weird that knowledge and quest becomes a very popular. In almost every hacking website or any famous hacker lair there is a question among its FAQ section called "will you teach me how to hack?" This question is often replied to by that the newbie should read, read, and read till he is good enough to ask new questions. We could imagine how huge the number of Newbies surfing the net trying tool, script, and exploits trying to understand. Penetrating a system is an attractive thing to do for most enthusiastic technology involved people; it implies good knowledge of the penetrated

system which is an appreciated quality in the information age, at least for some.

2.2.4. Industrial Espionage

The difference between competitive intelligence and industrial espionage, for example, is significant. By definition, industrial espionage refers to illegal activities - which range everywhere from outright theft to bribery and everywhere in between. Conversely, competitive intelligence collection is governed for the most part by adherence to corporate and professional ethics which preclude the use of illegal means to obtain information.”[Nolan, 1996]

2.2.5. E-Terrorism

How serious is the danger of Internet-based terrorism? [Fisher,2002] Since Sept 11, terrorism is head news, and the computer world is waiting for the e-terrorism. [Winkler, 2001]

"As soon as someone uses the term e-terrorism they begin to lose credibility with me," says Graham Ingram, general manager for Internet security watch-dog AusCERT. "The whole idea of terrorism is to do something that creates terror. You need the physical realization of violence, and there is very little terror inspired by bits and bytes". "You might have a terrorist act which involves violence and death, and somehow interrupt the 000 emergency numbers so that the authorities couldn't respond as effectively." Ingram says. Kim Valois, security service director at IT integrator CSC not agree with Ingram. He said that e-terrorism can include the use of information systems to support terrorism.

"Any disruptions to information systems that are in public use, like banking or transport, any use of such systems to disrupt, undermine or cause damage in some way -- attacks against the power supply or the banking system -- these are all part of e-terrorism." Valois says. "However, some groups are more likely to use the Internet for information dissemination or fundraising activities."

Although internet can be used by terrorist, the occurrence of e-terrorism is still low. and this because terrorist usually need to make fear and visual image fear. If we mention Oklahoma City, with the images of buildings blown away come to mind. With Pan Am flight 103, the image of a side of a 747 comes to mind. TWA flight 847 created the image of a terrorist in a mask holding a gun to the head of a pilot. There is Samples for e-terrorism like crash of the AT&T telephone network in 1991, the power outage in the Pacific Northwest in 1998, the denial of service attacks in 2000, the Chinese

"info war" and the Code Red and Nimda worms of 2001. Consider what the following mean to you personally: Code Red and anthrax. Clearly, anthrax creates a whole different level of fear. Traditional terrorists appreciate the Internet and the resources that it offers. It provides a ready way to exchange information. So the traditional terrorist won't destroy internet. But the only exception is computer attack against companies supporting military attacks.

But there is another threat from nontraditional terrorist. They are Groups who want to damage technology or create negative effects on companies for specific reasons. For example, if someone could take down McDonald's shipping computers that are involved in getting stock to McDonald's restaurants, they could cause damage to its revenue. Any company with an international presence is a possible target for one obscure reason or another. General Marsh, the head of the now disbanded President's Commission on Critical Infrastructure Protection, declared that "banks lose billions of dollars a year to electronic thefts. Statistics about computer crimes continue to climb". [Winkler, 2001]

This all clears the danger of the new terrorism "E-terrorism" which many government officials and terrorism experts consider a serious threat to national security with the potential for causing mass confusion and loss of life. The Bush administration confirmed that it will spend \$10 million to launch a newly intensive war against cyber-terrorism, "Cyberspace," said one Bush administration official "is our next battlefield. And the president has concurred that we need to be better prepared for it." President Bush will appoint Richard Clarke, the longtime coordinator of security, infrastructure protection and counter-terrorism for the National Security Council, to the position of special advisor to the president for cyberspace security. Retired U.S. Army Gen. Wayne Downing will be appointed deputy national security advisor and "national director for combating terrorism," administration officials said. [Thomas, 2001]

2.3. Actors

In this section we will go through the different actors in the information warfare in the cyberspace. Different opinions are discussed here some opinions clearly oppose each others especially when it comes to the hackers and crackers which will be shown in the following review.

2.3.1.Hackers

A hacker is a term that means a clever programmer and someone who knows a lot about programmable systems and how to increase their capabilities. Originally a hacker was someone who makes furniture with an

axe. But in the new hacker's dictionary, which is also referred to as "the hacker's jargon", Eric Raymond (compiler and maintainer of the jargon) lists some of the hacker's characteristics. The first is that hackers enjoy learning the details of programming languages and programmable systems. The second is that they really enjoy programming, in other words programming is a hobby rather than a job to perform or theoretical issue to talk about. The hacker's ability to pick up programming quickly is considered another characteristic. Another characteristic is that they are experts in a particular language or system as in Unix hacker or C++ hacker. One of their most important characteristics that could not be described better than Raymond's word as he describes a hacker "One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations" [Raymond, 2000]. Finally they appreciate other hackers' hacks. [Raymond, 2000]

2.3.2.Crackers

The cracker is the one who penetrates systems and perform malicious actions like destroying data, denying the service of some sites, or stealing data. Raymond says that they often call themselves hackers but their involvement in vandal actions is what differentiates them from hackers. [Raymond, 2000] Moreover, a cracker could turn into a hacker in advanced phase of his or her cyber life. After outgrowing the desire to crack and penetrate systems crackers understand the real meaning of hacking and stick to the hackers' ethics. [Raymond, 2000]

2.3.3.White Hats

As described by "whatis.com" white hats are hackers who search for vulnerable systems and report these vulnerabilities to the owners of those systems. They don't oppose a threat to the internet society, on the contrary, they provide a very valuable service to the internet as they help keeping systems one step ahead of malicious hackers or crackers as we discussed earlier [searchsecurity.com, 2001]

2.3.4.Black Hats

A black hat is a cracker that penetrates systems for his/her own benefit. He or she takes advantage of the revealed data by trading or by telling about the vulnerable system to others blackhats rather than telling the responsible organization itself. [searchsecurity.com, 2001]

2.3.5.Grey Hats

The gray hat is mix between black and white hats. He or she has no malicious intends but as grey hats find out about vulnerable systems the alert the involved organizations as well as the hackers society. This could cause other crackers to penetrate the system and sabotage it.[searchsecurity.com, 2001]

2.3.6.Script Kiddies

The script kiddies got their name from their reputation of gathering malicious software “scripts” and attacking networks with such scripts. They lack knowledge and they are destructive. As noted in whatis.com, hackers contempt script kiddies because they add nothing to the art of hacking, instead, they unleash the attacks of media on the hackers’ communities.

2.3.7.Lamers

According to the jargon file a lamer is an annoying beginner who is late behind in his cracked software, like in warez d00dz lamer, or in his knowledge like in crackers lamer. It also means that he scams codes of other crackers rather than understanding the concepts and making his own. [Raymond, 2000]

2.3.8.Cyber Warriors

“The final role that hackers may play, and the most disturbing, is that of “cyber warriors.” Yes, it sounds a bit like a video game. Unfortunately, in the not too distant future, and perhaps in the present, this may be more than science fiction. There have been too many rumors and news stories about governments building up teams of cyber warriors for this to be just fiction. Naturally, the press has locked onto this idea, because it doesn’t get any more enticing than this. Naturally, the public has no real details yet about what these special troops are.”

Nearly all types of infrastructure, power, water, money, everything, are being automated and made remotely manageable. This does tend to open up the possibilities for more remote damage to be done. One of the interesting questions surrounding this issue is how governments will build cyber warriors. Will they recruit from the hacker ranks, or will they develop their own from regular troops? Can individuals with special skills expect to be drafted during wartimes? Will hackers start to get military duty offered as a plea bargain? Also will the military be able to keep their secrets if their ranks swell with the hackers who are used to free flow of information? [Rayan, 2000]

Actors in the cyberwarz are not limited to the previously mentioned. In fact there are many others which are distinct or overlapped with what was mentioned previously. Also, certain actors could claim to be of other category, for example, crackers calling themselves hackers. A very important note that must be mentioned in such content is that a person could experience many of these states as stages in his or her evolution to be a hacker. Elf Qrin [Cappello, 2000] discusses this issue more deeply clarifying the shifts between stages.

3. METHODOLOGIES

In this section we will exploit the three main levels for attacking a network. Each level contains different classes of attacks and information gathering techniques. The more an attacker sophisticates his or her attack level or methodology the more its likeliness to be a successful, clean and anonymous attack. The three methodologies are: a- simple attacks, b- professional hacking, c- strategic hacking. As we will see the attacks gets more sophisticated as we go on. And each more-sophisticated attack includes the techniques and methodologies of less-sophisticated ones.

3.1. Simple Attack

We will now describe the elements of simple attacks which is hardly could be called a methodology because it is so simple. However it is essential to understand it as independent technique due to its popularity. Script kiddies who are least sophisticated and most spread use simple techniques to crack into systems, as we discussed earlier these techniques are designed and coded by black or grey hats.

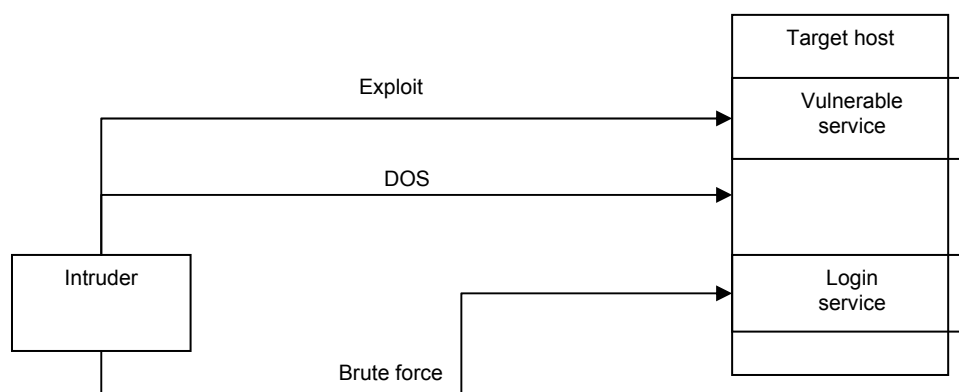


Figure 3.1.1 Simple Hacking

3.1.1. Simple Hack

A typical script kiddie will take a random alive IP address and start trying the scripts he has on. The scripts will mostly do one of two things:

- 1- Try to DOS this system.
- 2- Try to gain access to the system.

3.1.2. Gaining Access Through Exploiting a Vulnerable Service

Any given program has bugs, these bugs could impose security breaches into that system. While some exploits are just sitting there waiting to be discovered, some other exploits need a lot of hard work to make it work. Buffer overflow exploits are a good example of a well engineered exploit. In either way script kiddies do not design or code the scripts or programs to do the payload, they just use it. Buffer overflow exploit depends on an idea called smashing the stack. When an arbitrary binary is receiving input it saves it in its buffer in the memory, the same old buffer that is keeping the return address for the program. Some function like “strcpy” just don’t check the input size and compare it with the buffer. This causes buffer overflow because the input string will overflow the return address for the function and the binary will crash. If the input to such function is well engineered it could replace the return address for the function and make it point to another privileged binary to be run for the intruder. This way an intruder could get access to the remote server. [Ryan, 2000]

3.1.3. Gaining Access Through Cracking Passwords

Another popular way to gain access to some service is by cracking its password. Password attacks could be performed on three levels, 1- simple password guessing, 2- dictionary attacks, 3- brute force.

There is no significant difference in the technique itself, the difference lies in the passwords that would be tried on the targeted host. An intruder could simply try some passwords that he thinks could be the one like trying the same user-name or user-name123. In the second technique he tries a file called the dictionary file. Some times this file could be of much less words and called word file. The script tries every word in the file as a password. Some advanced scripts add common strings like "123" to the strings in the file. The third technique is brute force in which the script runs all possible combinations of characters as the password string.

There are cases where password discovery process exists in some intersection between password cracking by try-and-error and exploiting a bug in implementation. Windows 98 share password implementation bug decreases the time needed to brute-force.

3.1.4. Denial of Service

Denial of service "DOS" is consider relatively easier to perform than other types of attacks. All the attacker does is stopping the service of some organization. This could be done by stopping/hanging the server that provides it, by stopping/hanging the service itself, or by cutting the road to this service by tampering with the network path to it. The most significant common properties of DOS are: 1- Its destructive nature. 2- Relatively easy, 3- Very high degree of being anonymous.

Denial of service may not be the most malicious act in many cases. The evaluation of how malicious an attack is depends on the targeted organization.

3.2. Professional Hacking

Hacking could be done in professional manner if the intruder followed a frame work that makes the attack much more effective. "Hacking exposed" illustrated the anatomy of a hack in way that helped us a lot in writing this paper. However due to the nature of our work we face situations that are slightly different from this anatomy. Next, we will describe the Hacking exposed hack anatomy not typically as described in the book, but with the slightly different properties.

The anatomy of a hack consists of 10 process which are: a- Foot printing, b- scanning, c- enumeration, Gaining access, d- escalating privileges, e- Pilfering, f- covering tracks, g- creating backdoors [McClure, 1999] , and h- misinformation. [Ryan, 2000]

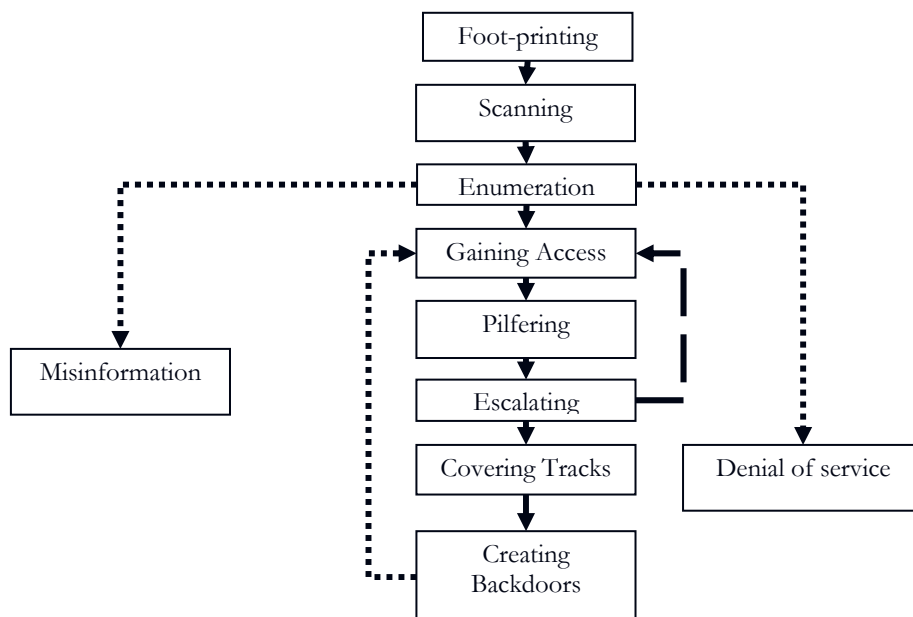


Figure 3.2.1. Professional Hacking.

3.2.1. Foot Printing

Foot printing is the initial wide-scale information gathering phase. In this phase the intruder gather all possible information about the targeted network. Gathered information include network addresses, manuals and attendance sheets and maybe even passwords from the company dumpster, hidden comments in the company's website HTML source files, network routes to the company's assets, stock and market details, merger and administration changes details.

3.2.2. Scanning

Step “b” the Scanning scans all the target networks’ resources in order to identify alive machines, their operating systems and the services running on these machines.

3.2.3. Enumeration

Enumeration tries to gather more information about every service. For example: the users and groups of the service, the version, and possible passwords.

3.2.4. Gaining Access

Gaining access is the phase where the intruder uses the knowledge gathered from the previous phases to get access to the system either by cracking a password or exploiting a vulnerable service. In this phase the intruder has an actual hand in the vulnerable system that enables him to start escalating his privileges.

3.2.5. Pilfering

Pilfering once again is an information gathering phase in which we want to penetrate trusted systems on the network. Now we could enumerate more systems and penetrate them.

3.2.6. Escalation

The Escalating privilege phase is concerned with upgrading user privileges to administrator privileges so that intruder has full power over the system.

3.2.7. Hiding Traces

Afterwards an intruder must cover his tracks, for example by deleting the logs and hiding his binaries.

3.2.8. Installing Backdoors

At last an intruder would install backdoors so he doesn’t go into the whole path again in order to own the system. [McClure, 1999]

3.2.9. Denial of Service

Two other phases are slightly different from the previous ones that are “denial of service” and “misinformation” attacks. The intruder could DOS

some services to serve a certain purpose in his attack scenario or just to stop the service if this is his only target. DOS is very helpful in accomplishing some targets, like for example: gaining access to Cisco router or disabling an intrusion detection system “IDS”.

3.2.10. Misinformation

Misinformation attacks like email relaying like DOS could be a target themselves but could also be a mean to social-engineer administrators or spoof orders to machines and destabilize operations.[Ryan,2000]

3.3. Strategic & Advanced Hacking

In the previous section we have seen how an intruder could plan to and attack a network. However, sometimes these steps are not enough to attack huge networks. If an intruder is after a huge secured banking network he should dedicate a great amount of time to plan the attack. The research has made some effort in merging professional hacking methodology with destabilizing networks methodology. The concept of destabilizing networks is based on extracting information about many aspects of the targeted network, then analyzing this information with many tools to determine the weak points in the network [Carley, 2001]. Figure 3.3.1 illustrates the merger between professional network hacking and network destabilizing techniques. The life cycle of the advanced or strategic hacking is hardly complete but it should give a clear overview of the whole process. The life cycle begins with a piece of information, for example, a trace, a company name, or an employee name. Even small amounts of information are very useful in footprinting a network. Unfortunately we will not be covering the details and steps of sub-phases like Footprinting because they are out of the scope of this paper, we will only emphasize on destructive effect that a framework like –what we call- strategic hacking would have on organizations. The framework is divided into eight main phases which are: 1- Information Gathering, 2- Analysis, 3- Reliability checking, 4- Planning for the attack, 5- Initiating the attack, 6- Escalation loop, 7- Accomplishing the objectives, and 8- Ending the attack.

3.3.1. Information Gathering

The information gathering phase is very important. The accuracy of the analysis, planning and all the coming phases depend on the accuracy of this phase.

3.3.1.1. Foot Printing

As pointed before, we will not describe the detailed steps of Foot printing. However, an important point must be emphasized in this context. The open search as a step in Foot printing renders very important information. Imagine the intruder searching for the names and emails of the targeted organization's employees. Finding more details about them the intruder will attempt to attack their home computers as an easy totally unsecured privileged hop. For example, the employee may use the same password for the work and home computer, or he could be a workaholic who connects to a backdoor on his work machine to work late hours. Information like this could have negative impact on the organization's security.

3.3.1.2. Scanning

In a large network scanning could be a very tedious process especially if the intruder has to tweak his scanners to evade intrusion detection systems. This is another reason to emphasize the importance of a good Foot printing process that could lead to a more specific domain of machines to be scanned.

3.3.1.3. Enumeration

The information gathered in this sub-phase is very critical. The intruder tends to be very careful about the information gathered because this will be the essence of exploiting vulnerable services and gaining access to the targeted system. System administrators try to make this process harder by changing the banners and some messages in their system to delude intruders, but as usual there is more in identification than banners.

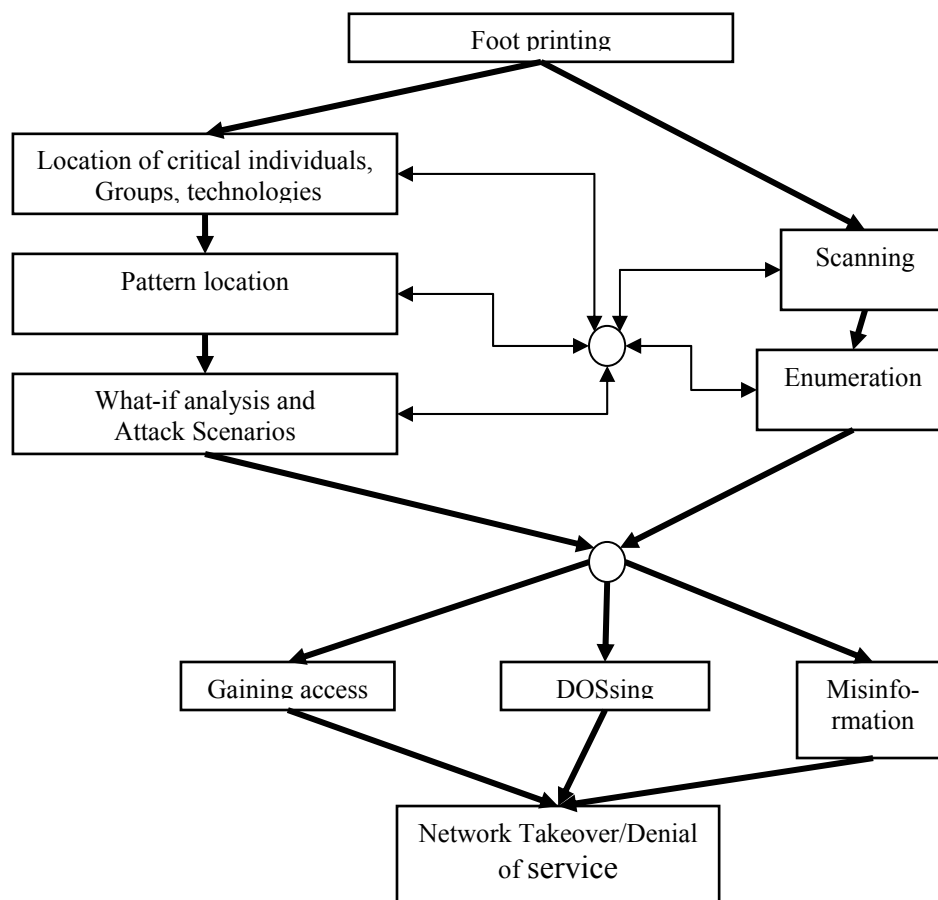


Figure 3.3.1. Simple Merger between Destabilizing Networks and Professional Hacking.

3.3.2. Analysis

The bigger the system is, the bigger are its bugs. Now that the intruder collected almost all possible data about the targeted network he begins the analysis. The analysis is dedicated to find out weaknesses in the network. The analysis is divided into three sub-phases: 1- location of critical individuals, groups and technologies, 2- pattern location, and 3- What-if analysis [Carley, 2001].

3.3.2.1. Location of Critical Individuals, Groups and Technologies

In this phase the intruder tries to locate entities and link them with their properties. For example, these groups could be identified in the targeted network: 1- Entities or groups that are if removed the network will pretty much be affected, 2- Entities or groups that are unlikely to act towards different information, 3- Entities or groups that act fast to different information, and 4- Entities or groups that have relatively more power or authority. [Carley, 2001] There are many other entities that could be checked in Carley's paper for further details. Note that the research team is using the term "Entity" sometimes instead of the term "individual" to impose a more generic concept.

3.3.2.2. Pattern Location

There are many pattern location techniques and tools that are very helpful in discovering patterns that are not visible to a man's eyes especially if combined with machine learning techniques. Some of the main points that should be identified in this analysis are: 1- the central tendency within the network, 2- basic components, 3- critical differences between sets of networks [Carley, 2001], as well as other points that should give the intruder a good idea about how to plan for the attack.

3.3.2.3. What-if Analysis and Attack Scenarios

The attacker begins to perform a what-if analysis based on the knowledge he has about the organization. "What if I target the mail server as the first hop into the network?", "What if I target the general manager computer at his home by hunting him down on an IRC channel?", and many other options. The attacker could end up with multiple attack scenarios ready for testing and initiation. As Carley [Carley, 2001] noted software agent models could render great results in such analysis, that is in our case, malicious results.

An important note is that there is a feedback relation between the information gathering phase and the analysis phase. Almost, always analyzing information raises new issues and uncompleted patterns that need more information to accomplish the analysis.

3.3.3. Reliability Checking

Now the intruder is ready to test his theory. This is important because if he is targeting a large network he is not performing a simple hack, instead he performs a sequence of misinformation, DOS, and gaining access attacks. Sometimes these attacks will be scripted or automated. The intruder—as

much as possible- will try to test these exploits. Two factors must be taken into consideration: 1- the timing of testing. 2- the efficiency of monitoring systems on the targeted test service, and 3- the behavior of those responsible of monitoring this service. On the other hand the administrator could be smart enough to recognize that something strange is going on, but could be busy enough to discard the whole thing!

3.3.3.1. Stealth-Testing the Vulnerabilities

This step is concerned with running the exploits on the well enumerated service, and making sure that when the time comes the exploit will work.

3.3.3.2. Brute Forcing Unmonitored Services

Sometimes, the intruder will need lots of time to brute force a login service. Brute forcing takes tremendous amount of time that's why it should be assigned enough time before the attack begins.

3.3.4. Planning for the Attack

Now that information gathering, analysis, and testing is done, the intruder has a good idea about what he will do and how to do it. He sits down to gather the toolkit and draw the attack trees, but certain critical issues must be put in mind while writing down his plan.

3.3.4.1. Sequence of Attack and Prerequisites

The attacker must organize his attack in way that clearly shows the prerequisites of each action. It will be even more professional to write down why these steps need each others as prerequisites. The more professional the attack plan is the more it's likely to succeed with little margin error. As pointed before, the larger the systems the larger are the bugs in it. This means that the huge attack scenario designed to attack the network has bugs that could turn a successful attack into an easy way to go to prison.

3.3.4.2. Semi-DOS, Flooding, and Reboot (Critical Timing)

An example of how timing could be critical is network devices. Some network devices when flooded or DOSsed fail opened. That is, to crash its access control system and keeping the accessibility for a certain amount of time. Even worse, some routers fall to its hardware authentication module with default password. In such cases the attacker must login to the router and

downloads the configuration file before the router restarts and gets back to its normal status.

3.3.4.3. Shifts and Using Exchange Times

Shift changing times are very useful moments to probe and start attacking the network. The first admin is getting too tired and bored to watch alert and the second is still sleepy because it's the 4:00AM shift and is trying to make coffee while listening –actually, not listening- to the first administrator telling him about the latest event or about something that he should checkout.

3.3.4.4. Attack Trees and Scenarios

At last the scenarios and attack trees find their way to documentation. The tool kit is prepared and scripts are containing all the ways that the intruders need to takeover the network. Embedding attack trees in scripts could make devastating result similar to what happened in the “nimda” worm. This work had four scenarios to gain access to a remote machine then using all these machines to attack the white house while turning the internet into a fragile worm nest by consuming its bandwidth.

3.3.5. Initiating the Attack

At the right time, the intruder begins the attack. Most of the attack techniques we discussed before, so we will be short and stress on some points.

3.3.5.1. DOS-Sing the Targets

Denial of service will be used to: 1- Disable monitoring systems. 2- Crash open access control systems. 3- Destabilize the network performance, and distract administrators and make them get busy with other things than monitoring the network.

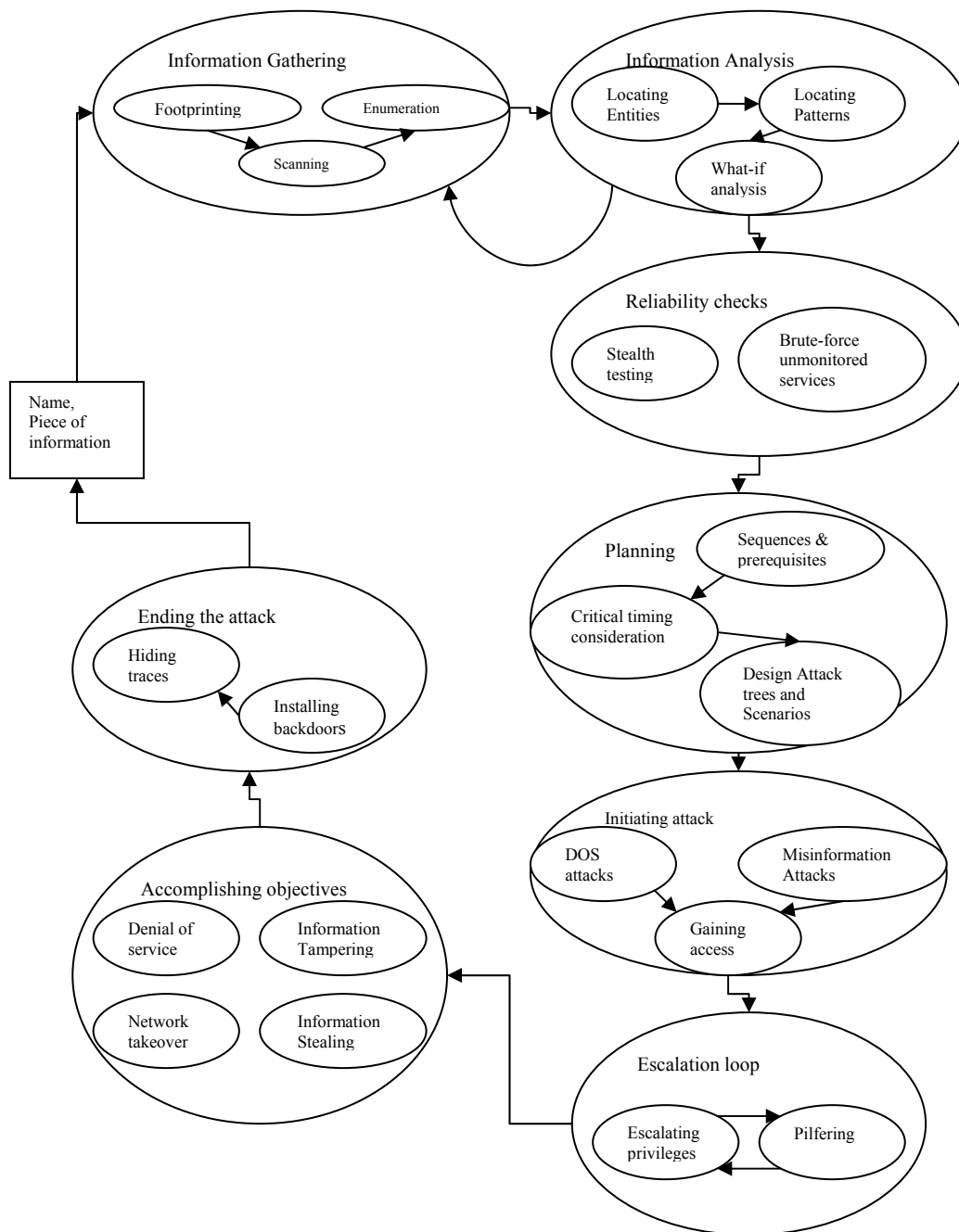


Figure 3.3.2: Advanced hacking life-cycle

3.3.5.2. Misinformation Attacks

Combined with DOS attacks misinformation attacks could be very powerful in: 1- distracting administrators. 2- Making it very hard to trace the real attacker network address. 3-making Denial of service un-block able by changing the packets' source addresses rapidly.

3.3.5.3. Gaining Access

When the attacker finally receives a success acknowledgement from a vulnerable server through a cracked password or an exploitable service he know that he has little time to escalate his privileges to gain full control of the compromised server. The escalation loop begins.

3.3.6. Escalation Loop

This loop could iterate many times till the intruder gains full control of the targeted system. This system could mean a server or the whole network.

3.3.6.1. Pilfering

The intruder pilfers more system information that will be –most of the time- easier to get now that he has a hand in the network. He looks for trust relationships, passwords by sniffing, and more.

3.3.6.2. Escalation

Now the intruder uses this pilfered information to escalate his privileges. And the loop goes on till he accomplishes his objective which could vary from an attacker to another according to their motives.

3.3.7. Accomplishing the Objectives

The main technical objectives we may think of are: 1- Taking over a system, 2- Deny a systems service, 3- Stealing critical information, and 4- Tampering with critical information. They may not be all technical objectives for an intruder but they are sufficient for the needs of this paper.

An attacker could have more than one objective to accomplish. For example if he wants to ruin a companies reputation, he could deface its website (data tampering), shuts down its mail server preventing it from communicating with customers and suppliers (denial of service), and downloads its clients database to be distributed (stealing critical information). Accomplishing all theses technical objectives will accomplish his main goal “ruining the company reputation”. Also if the intruder managed to take over

the network or at least a great deal of it he'll be able to pay it some really disturbing visits later on.

The intruder may also accomplish objectives other than his main to make it seem like he is doing something else to elude the forensics people so he would be more anonymous and/or be able to come back in a less dangerous environment.

3.3.8. Ending the Attack

Now the intruder begins the last step. There are two objectives for him now: 1- keeping himself anonymous, and 2- providing a way that enables him to come back for some malicious actions later on.

3.3.8.1. Installing Backdoors

Backdoors vary in nature, they could be normal services secretly running on a different port, or they could be special binaries made specifically for this purpose. It is important that these backdoors be hidden from the eyes of the administrators.

3.3.8.2. Hiding Traces

The very last step to finish the job is hiding the traces. Clearing logs, removing intermediate accounts, and deleting or hiding the malicious binaries used to compromise the systems. The attacker logs of the network and the internet after this long sophisticated job.

4. SCENARIOS AND CONCLUSIONS

In this section two common scenarios will be presented to show how strategic hacking could be used to electronically terrorize important governmental and commercial entities over the internet. The first scenario is concerned with terrorizing governmental agency on the internet by taking over its network. The second scenario will attack the business of a commercial company by a simple but effective attack that is denial of service.

4.1. Takeover Scenario

4.1.1. Actors, Motives and Assumed Structure

The targeted network: A governmental unit that provides computerized license renovation.

The online service is not yet running but is being developed.

The time: Nothing specific.

Intruders: Someone who has interest in making the electronic government project fail. So, he hired a professional team of intruders (hackers/crackers) to do the job for him.

Motive: He wants to take over the network so he could manipulate it anytime he wants to prove something. And if have to he would want to destroy all the data in the network.

Figure 5 illustrates the details of the Targeted Network.

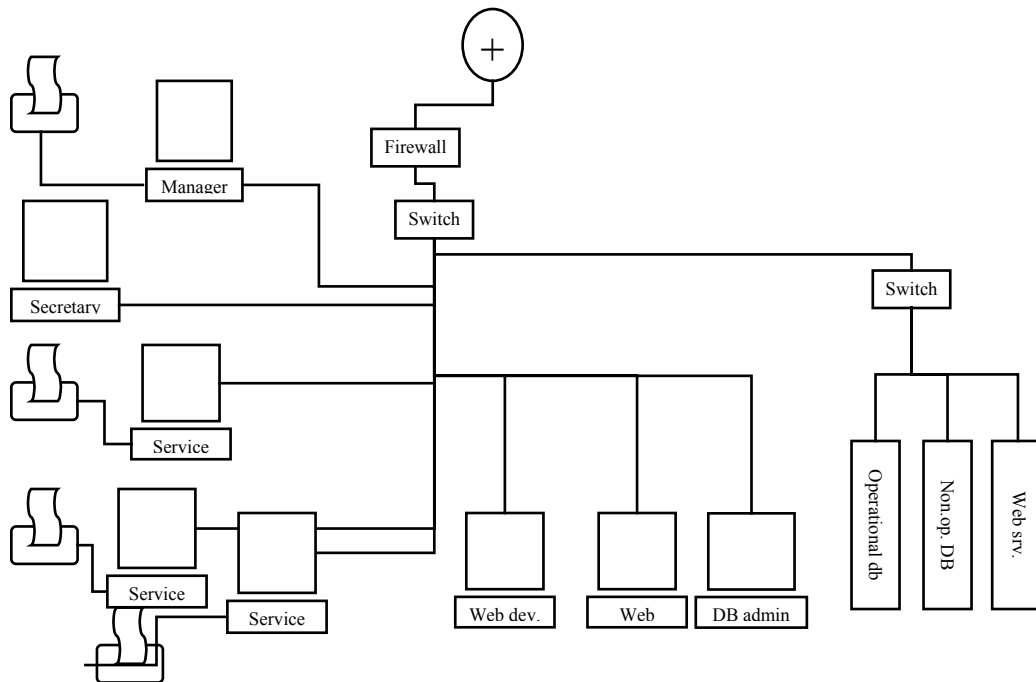


Figure 4.4.1. Targeted Network

4.1.2. Attack Scenario

The number of different combinations of attack scenarios is too large to be counted in this paper, but we will try to show in every step a sample of what could happen.

4.1.2.1. Information Gathering

4.1.2.1.1. Foot Printing

The Intruders team begins to gather all possible information about the unit. The team comes with the following results:

- 1- The secretary e-mail from the website.
- 2- The IP addresses for the network xxx.xxx.xxx.0-31. All real IP's.
- 3- A good idea about how the system works by going to the physical place and asking how to renovate a license.
- 4- The intruder notices that there is a room called "the server room".
- 5- The developer is a graduate of a different OS platform than administrators. This could mean non-standard Operating systems.

4.1.2.1.2. Scanning

The team starts stealth-scanning the range and gets the following results:

- 1- The firewall is badly configured to block only suspicious ports. It should have been a "deny all allow only what you need" policy.
- 2- The attackers presume that the firewall also allows all outgoing traffic.
- 3- The Machines scanning results are as following:
 - a. Web server is listening on: 1- 80, static pages. 2- 8080, some dynamic pages the developer is developing for the forthcoming service.
 - b. Operational database server: apparently the SQL server port is filtered as it shows from the scan.
 - c. Non-operational DB server: SQL port is opened as well as terminal server port.

4.1.2.1.3. Enumeration

Enumeration results:

- 1- Web server and operational database servers are updated with patches and have antivirus.
- 2- Non-operational database server is not.
- 3- Manager machine is sharing the printer and a writable folder.
- 4- All service machines have the names service-3 and username and password service.
- 5- All client machines are windows 2000.

4.1.2.2. Analysis

4.1.2.2.1. Location of Critical Individuals, Groups and Technologies

- 1- The secretary machine usually is less secured but has more information about the company than the whole company.
- 2- Web developer machine usually has more privileges than normal users but the developer most of time is not keen on security as administrator.
- 3- The technical group (web admin, database admin and developer) has access on the servers group.
- 4- The server group is in a separate room (maybe on a separate hub).
- 5- Only two users on the administrators' machines. This implies that the two different administrators (DB, web) most probably know the system's, web's, and database's password to be able to fill in for each others.

4.1.2.2.2. Pattern Location

- 1- The service machine is a pattern.
- 2- Having terminal service on the non-operational database could be a pattern on other servers.
- 3- The password for the servers could be similar (a pattern). If we could sniff one, we would get the rest.

4.1.2.2.3. What-if Analysis and Attack Scenarios

The intruders have a couple of options as an entrance point:

- 1- Send the secretary a Trojan horse.
 - a. They could find critical information about the manager, the company, and maybe even backup of the source codes and databases.
 - b. They could find old password or any other critical in mail boxes.
- 2- Attack the un-armored web server on the developer machine.
 - a. They will be able to get the source code and designs.
 - b. This goes for all client machines: they will sure gather new information and use it to sniff at least local password, brute force other machines, and make misinformation and DOS attacks.
- 3- Attack the SQL server on the non-operational server.
 - a. They maybe able to sniff passwords of the hub.

- b. Download the data of the server.
- c. Know the structure of the operational database and try to send queries.

4.1.2.3. Reliability Checking

4.1.2.3.1. Stealth-Testing the Vulnerabilities

The team tried a couple of exploits and they worked.

4.1.2.3.2. Brute Forcing Unmonitored Services

The intruders were not able to brute force NetBios services due to the firewall so they postponed it until they have a hand on the network.

4.1.2.4. Planning for the Attack

4.1.2.4.1. Sequence of Attack and Prerequisites

The intruders' team decided to attack the network with the three options they have. The attack will take place after the units working hours. They made sure the secretary got her fancy screen saver Trojan in the afternoon when she is too tired to work or think.

4.1.2.4.2. Attack Trees and Scenarios

The attack tree is drawn every intruder knows his part. The tool kit is ready.

4.1.2.5. Initiating the Attack

4.1.2.5.1. Gaining Access

Administrators are not available so there is no real need for misinformation attacks. And there are no IDS to DOS. They gain access nice and easy.

4.1.2.6. Escalation Loop

4.1.2.6.1. Pilfering

The intruders install their sniffers. They brute-force "netbios" from within the network. They gather and download every bit of

information they are able to get. They also find out that web server and operational database server have terminal service on.

4.1.2.6.2. Escalation

They get access to the service clients. They get access to operational SQL server the famous exploit of the SQL worm, and so on.

4.1.2.7. Accomplishing the Objectives

After three or more escalation loops in about a week the intruders group have almost all the passwords of the network. They are ready to do whatever their employer asks them to do.

4.1.2.8. Ending the Attack

4.1.2.8.1. Installing Backdoors

The intruders install 2 instances of “netcat” on the server. The first one will act as a server. The other one will act as a client that tries to connect every week to a previously compromised server by the intrusion team.

4.1.2.8.2. Hiding Traces

The team executes a root-kit that erases the logs, hide the binaries and erase any users they may have added to some systems.

4.1.3. Impact on the Organization

Invading the privacy of at least thousands of citizens which could very much compromise the electronic government project in Egypt.

4.2. Denial of Service Scenario

4.2.1. Actors, Motives, and Assumed Structure

The targeted network: A company that makes online reservations for vacations.

The time: right before summer vacations begin.

Intruders: the web admin of a competitor website.

Motive: the intruder website is not making enough profit, so he decides to stop the better competitor’s service to direct the customers to his website.

Figure 6 illustrates the details of the Targeted Network.

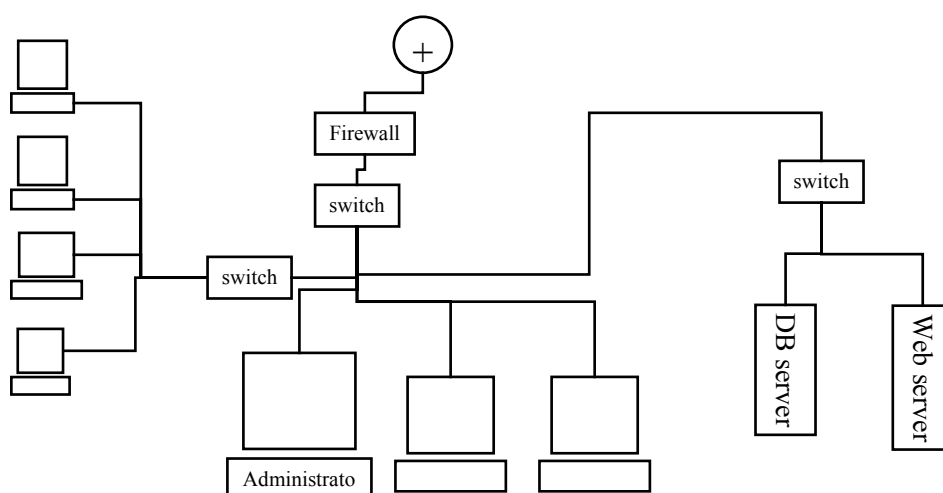


Figure 4.2.1. Targeted Network.

4.2.2. Attack Scenario

The scenario is not a complicated one although it's a very destructive and it costs the targeted organization a lot of money, time, and effort.

4.2.2.1. Information Gathering

4.2.1.1.1. Foot Printing

The Intruder begins to gather all possible information about the unit. He got the following results:

- 1- The IP of the website.
- 2- The IP addresses for the network xxx.xxx.xxx.0-12. All real IP's.

4.2.1.1.2. Scanning

The Intruder starts stealth-scanning the range and gets the following results:

1. The firewall is badly configured to block only suspicious ports. It should have been a "deny all allow only what you need" policy.
2. The Machines scanning results are as following:
 - a. Web server is listening on the following ports: 7 echo, 19 chargen, and 80 http.
 - b. Operational database server: apparently the SQL server port is filtered as it shows from the scan.

4.2.1.1.3. Enumeration

Enumeration results:

1. Web server and operational database servers are updated with patches and have antivirus.
2. The router is vulnerable to “EIGRP DOS” Attack.

4.2.2.2. Analysis

The scenario doesn’t require analysis as it’s purpose is only a destructive and it’s easy to be done.

4.2.2.3. Planning For the Attack

The intruder has two ways to attack the network:

- 1- Initiate DOS attack on the router.
- 2- Use the “echo-charge” vulnerability to DOS the web server.

The intruder chooses to use only one attack per time just to make the administrator confused.

4.2.2.4. Initiating the Attack

The intruder starts the attack by:

- 1- send a forget packet to the web server “charge” service that pretends it came from the web server “echo” service. This will cause a loop between the two services consuming the full processing resources.
- 2- Send forget “EIGRP” packets to use the “EIGRP DOS” router vulnerability.

4.2.2.5. Escalation Loop

There is no escalation loop. However, the intruder will repeat the attack scenario whenever the need arises.

4.2.2.6. Accomplishing the Objectives

The objective of the mission which is denial the online service of this company is accomplished.

4.2.3. Impact on the Organization

Let's assume that the company makes on average 100\$ per hour, the intruder manages to DOS it for at least 16 hour/day. So it will cost the company about 1600 \$ per day.

4.3. Conclusions

Malicious hacking or cracking is not a harmless hobby, in fact it, malicious internet intrusions cause companies great deals of money, effort, and time. Malicious hacking could be motivated by many motives; one of them is electronic terrorism. Advanced or strategic hacking is could be used to terrorize commercial as well as governmental organizations by stopping them from providing their services. However, keeping the organization from providing its services may not be the greatest threats it faces on the internet. A well design attack may compromise the organization's integrity. Such attacks could also threaten national projects like electronic government. Facing such threats is a must.

5. COUNTERMEASURES AND RECOMMENDATIONS

This small section will list some of the different groups pf countermeasures and make recommendations based upon the conclusions of this paper. However, there are a variety of countermeasures which is also out of the scope of this paper. So, we will only emphasis important point regarding countermeasures.

5.1. Countermeasures

We could divide countermeasures techniques into two main groups for simplicity, 1- Basic, and 2- intelligent. The basic grouped contains techniques for armoring the systems, building access controls, backup, logging, fall recovery, and intrusion detection. The intelligent grouped contains techniques for: making honey pots, intelligent intrusion detection systems, and forensic analysis.

Two important issues in designing a good security solution are: 1- Good policy and procedures, and 2- integration. We cannot give a good policy/procedure designing too much credit. Whatever security tools are installed and operated, these tools could be a waste of time if there is not well designed policy and procedures behind their installation. The other important

note is the integration between these security tools to perform more efficient security system and more reliable detection.

5.2. Recommendations

The cyberspace is a new world with lots of potential and opportunities. To be pioneers in this world or at least to have our share of it we should know. Egypt must invest in the human to be specialized in the cyberspace internals. Hacking as pointed out in the first section is about understanding this cyberspace. And, for any organization, in order to be able to secure itself it must invest in internet underground knowledge. Unfortunately this is very expensive even for big organizations. The recommendations of this paper is easy to write yet hard to accomplish. And they are:

- 1- Developing a research and development institution for cyber security that should provide solutions and consultation services for the governmental as well as the private organizations.
- 2- Increasing the awareness of people in the field of cyber security to increase the possibility of a new generation that could explore and develop this new space.

5. REFERENCES

[Cappello, 2000] Cappello(2000), HACKER STAGES v1.0.,
www.elfqrin.com/docs/HackerStages.html.

[Carley, 2001] K. M. Carley, Ju-Sung Lee and David Krackhardt (2001), Destabilizing Networks, Connections 24(3):31-34.

[Fisher, 2002] Fisher, V.(2001), E-Terrorism: An online war?,
<http://news.zdnet.co.uk/story/0,,t269-s2126759,00.html>.

[McClure, 2001] McClure s, Scambray j. (1999), Hacking exposed, Osborne / McGraw-Hill.

[Nolan, 1996] Nolan, J.(1996), What is Competitive Intelligence and What Can It Do To Us, <http://www.intellpros.com/lib/what.html>.

[Raymond, 2000] Raymond, E.(2000), The jargon lexicon,
www.catb.org/~esr/jargon/html/H/hacker.html.

[Ryan, 2000] Ryan R.I, Stace C., Mudge (2000), Hack proofing your network, Syngress, Media.

[searchsecurity.com, 2001] www.searchsecurity.com.

[Thomas, 2001] Thomas, B.(2001), U.S. to Intensify Effort against Threat of Computer Terrorism, www.latimes.com/news/nationworld/nation/la-100901cyber.story.

[whatis.com] www.whatis.com

[Winkler, 2001] Winkler, I.(2001), Are companies really ready for e-terrorism?, www.zdnet.com.au/newstech/security/story/0,2000048600,20261574,00.htm

Author Index

A

Abd El-Mageed T.	49
Abdel Wahab M. S.	77,151
Abo El-Fotouh M. A.	77,151
Ahmed M.	197
AlAschkar S. E.	229
AlGamal S.	49
Ali A. A.	49
Ali W. A.	311
Azer M. A.	63

B

Barika F.	293
-----------	-----

C

Chadwick D.	5
Courtay O.	217
Charbonneau A.	197

E

El-Hadary O. S.	229
El-Hadidi M. T.	103
El-Kadhi N.	293
El-Kassas S.	275
El-Keissi G.	275
El-Mageed T.	49
El-Said G. R.	123
El-Soudani M. S.	63
Eloff J. H. P.	261

G

Ghedira K.	293
Guette G.	217

S

Samarati P.	165
-------------	-----

T

Tolba M. F.	77,151
-------------	--------

V

Venter H. S.	261
--------------	-----

Vyskoc J.	95
-----------	----

Y

Youssef A. M.	63
---------------	----