# Importance of cryptography in network security 2D1441 Seminars in Theoretical Computer Science

Susan Ström, 740916-7546, susie@kth.se
Oskar Wiksten, 781228-6677, owi@kth.se

26th May 2003

## 1 Introduction

Some decades ago, computers were large machines isolated in a room, used mainly for performing functional programs that often were of a mathematical character. Within a short amount of time, the need was recognized for transfering data between computers, and the LAN[1] was born. At this time, each LAN was isolated from other unknown LANs, and there was a high level of trust within each LAN. The interconnection of separate LANs was the next logical step, and the concept of inter-networking was born. Initially, the administrators of these few LANs needed close cooperation with eachother in order to make the systems work; there was still a high degree of trust between systems and systems administrators.

Today, the Internet is a decentralized network of networks. Anyone can connect to the Internet and send out just about anything, even if it is potentially detrimental to the unsuspecting recipient. This places a large burden on each host and each organization that has Internet connectivity to be responsible and take the necessary precautions to ensure network security.

While there is enormous amounts of activity within the field of IT security, much remains to be done. Perhaps one of the most important steps towards better IT security is increasing the security mindedness of all IT specialists. Considering the short history of network communication, it is not surprising that security has taken a back seat to getting the most efficient product out on the market faster than anyone else. Since implementing security mechanisms in distributed applications creates extra overhead (requiring more memory, handshaking, more CPU for calculation of keys, etc.), and requires a complete understanding for IT security on the whole, both product efficiency and product-launching speed are compromised by adding network security features.

[1] Local area network

Since the unsuspecting customers did not initially ask for security for certain applications, they did not get security for these applications.

There are two general categories of network security mechanisms: add-on, entailing literally a separate bit of hard-ware added on to the host, and built-in. The problem with the former type of security mechanism is that each user must actually take the extra step and acquire, configure, and make use of the security mechanism correctly. Thus, the more security measures are built into a system from the beginning, the better the odds are that they will function as they were intended. Again, the burden of maintaining a correct security paradigm must remain with the application developers and IT specialist, not with the typical user. Until network communications mature and the typical user becomes much more savvy to security issues, this paradigm must assume a hostile environment in which the typical user takes no precaution whatsoever to protect themselves from the many threats completely unknown to them.

Cryptography is the strongest tool we know of in implementing security services, but as we will see, it is useless against certain types of attacks. So, what is the status of network security today? What roll does cryptography play in network security? Is it alone sufficient, relieving all IT specialists of the burden of maintaining a responsible attitude toward network security? Or, is it actually the configurations performed by these specialist that will take us to the next level of security? This paper aims at answering these question.

## 2 IT security overview

In order to evaluate cryptography's role in network security, it is important to understand the inherent vulnerabilities in network communication systems. These potentially exploitable aspects of networks have lead the development of cryptographic mechanisms designed specifically to combat certain types of weaknesses. As we will see, no one particular mechanism can provide IT security; each correct combination of cryptographic mechanisms inhibits the exploitation of one or more known threat.

We will begin this section by defining the services necessary for achieving network security, and continue by taking a look at the inherent vulnerabilities that exist in todays networks. When one of these vulnerabilities is taken advantage of, intentionally or unintentionally, then we have the subject of the next section, cryptographically preventable attacks. At this point, the stage is set for understanding the needs that todays security protocols aim to serve, and it is in this light that we will take a brief look at several cryptographically-based protocols in use today.

### 2.1 Security Services

Security services are services the add to the security of a system and are intended to counter certain types of security attacks. A common way of classifying these services is as follows:

1. Authentication achieving authentic communication involves ensuring that everyone is who they claim to be, not only initially, but during the whole period of communication. All data messages come from where they claim to come from.
2. Access Control once authentication is provided, access control is the service of controlling/limiting access to host systems and applications in a network.
3. Data confidentiality This service protects data from passive attacks. This involves not only the content of the data packet, but where the packets are going to and coming from, as well.
4. Data Integrity The main purpose of this service is to detect when data has been altered in transit.
5. Non-repudiation Non-repudiation prevents either party of a communication (the sender or the receiver) from denying sending a message, and both parties can prove that the other has sent and/or received the message.

In addition to these, availability is also very important and difficult to provide. Resources should always be available to those with appropriate access privileges. An attacker should not be able to interrupt, or unnecessarily slow down, access to the resource.

## 2.2 Vulnerabilities

There are a number of factors which contribute to the inherent insecurity of networks; their size, complexity, and decentralized administration; their unsuspecting and curious users; and the busy programmers that made it possible to connect the two. These vulnerabilities include [1]:

- wiretapping
- impersonation (ip-spoofing)
- message confidentiality violations
- message integrity violations
- code integrity violations
- denial of service (DOS)

Traditionally, computer security specialists have tried to minimize the exploitation of these known vulnerabilities. Unfortunately, vulnerabilities are often discovered (or at least taken seriously) only after they have been exploited.

### 2.3 Attacks

Vulnerabilities are also called threats, because the risk of their being exploited exists. A threat becomes an attack once the dreaded has occurred. There are many different reasons why one would launch an attack [2], making the task of preventing and diminishing the effects of such attacks all the more difficult. For each vulnerability, there are countless numbers of attacks. Some "classic" attacks are:

| | |
|---|---|
| Masquerade | Pretending to be someone you are not. An attack on authentication, data integrity. |
| Bypassing | Controls circumventing access control |
| Authorisation violation | Circumventing both authorization and access control |
| Trojan horse | A program that has covert activity beyond what it appears to be doing, protentially compromising all security services. |
| Trapdoor | A program that has a secret entry point. |

## 3 Applied Cryptography

Cryptography is widely used in networks. It can be applied anywhere in the TCP/IP stack, though it is not common at the physical level. The level at which cryptography is applied directly effects its transparency to the user, and its purpose. Cryptography, as we will see, is used for much more than data confidentiality. Indeed, none of the above mentioned security services would be possible to offer without cryptography. As mentioned previously, these services rely on a combination of various security mechanisms, most of which rely on cryptography in one form or another. Cryptography is also used in complicated protocols that help to achieve different security services, thus called security protocols. These protocols also rely on various cryptographically-based mechanisms to achieve the desired result.

### 3.1 Cryptographically-based mechanisms

The building blocks of computer security are cryptographically-based mechanisms. These mechanisms have many different implementations, and can in turn use other security mechanisms. Some of these mechanisms are:

- encryption - used heavily to accomplish all security services.
- access control mechanisms - commonly access control list (ACL) or user capability lists (UCL).
- data integrity mechanisms - these mechanisms often aim at detecting, rather than correcting, data that has been altered (intentionally or unintentionally) in transit. Relies heavily on hash-algorithms and encryption.

- authentication exchanges - mechanisms to realize authentic communication. In the client/server model, this can include various handshaking protocols that implement the above three mechanisms. Can include digital signatures (a combination of data integrity and encryption mechanisms to achieve non-repudiation).
- traffic padding - a technique that aids in data confidentiality.

## 3.2 Cryptographically-based protocols

Combining the above cryptographically-based mechanisms, security can be achieved for different purposes.

For each protocol:
- what does it do? brief description.
- what security services does it provide?
- what security mechanisms does it need to provide these services?

### 3.2.1 SSL

The SSL[2] protocol uses an encrypted tunnel to create a secure channel for exchange of arbitrary data, thus making it applicable as a transport mechanism for other non-secure protocols. SSL provides all of the above mentioned security services, except access control (not applicable). In addition to these services, SSL provides protection against replay attacks.

### 3.2.2 SSH

Similar to SSL, SSH[3] uses an encrypted tunnel for exchange of data, that can be used as a transport layer for other non-secure protocols. SSH provides all of the security services by using all of the above mentioned cryptographically based mechanisms.

### 3.2.3 Kerberos

Kerberos is a complex protocol used in open distributed systems to provide mutual authentication for both the client and the server. It is unique in comparison to similar authentication protocols in that it only uses symmetric keys. All of the security services are provided, except non-repudiation. Kerberos makes use of all of the above mentioned cryptographically based mechanisms.

### 3.2.4 SET

SET[4] is a set of security protocols designed to protect credit card transactions over the Internet. It supplies all of the security services except for access control and uses of all of the above mentioned cryptographically based mechanisms

[2] Secure Sockets Layer
[3] Secure SHell
[4] Secure Electronic Transaction

except of course access control mechanisms. The key feature of the protocol is that the merchant only gains access to the information necessary to carry out the order (order information), and the bank(s) only gain(s) access to the information necessary to approve the transaction (payment information). This is accomplished through a clever combination of cryptographically based mechanisms.

### 3.2.5 PGP

PGP[5] is used for encrypting the content of an email inside a regular SMTP email, with the use of asymmetric encryption for convenient key exchange. It supplies authentication, data confidentiality, and data integrity, and thus uses the corresponding security mechanisms.

## 4 Cryptography's role in IT Security

As can be seen above, cryptography plays a critical role in providing all five security services. Indeed, without cryptography, the cryptographic mechanisms that provide these services would not exists. The use of cryptography reflects the problems that were in focus when the cryptographically based mechanisms and protocols were developed. That is, complicated routines were developed to thwart complicated attacks. All of this looks good in theory, but due to constraints on budgets, performance requirements, and finite memory in computing resources, security features are not prevalent in today's networks.

One of the reasons for this deficiency could be the complicated nature of these mechanisms and protocols. For example, to successfully use PGP, two users that would like to communicate with each other must properly configure their software. Another reason is that certain solutions are not compatible with others. For example, an SSL connection through a firewall prevents the firewall from being able to examine the contents of the packets, thus undermining its entire purpose. There are yet other reasons why cryptography is not used much as it ought to be, having to do with global politics and export restrictions on algorithms and software.

While a system that relies on a cryptographic implementation can be secure against some types of attacks, there are still other factors that can degrade the overall security of the system. For example, a system using a secure cryptographic implementation for user authentication could be made obsolete if the user chooses a weak password.

## 5 Reality

In addition to the "classic" attacks that exploit inherent vulnerabilities of networks, there is a whole other class of attacks that in some cases can be far worse

[5] Pretty Good Privacy

than breaking a cryptosystem. Such attacks can include using weaknesses of the implementation of the cryptosystem to circumvent the security, or attacks that rely on implementation flaws in software related to the security of the system. These can all be attributed in one form or another to the “human factor”.

## 5.1 Frequency

In October 2002, SANS/FBI released a list of the top twenty security current vulnerabilities found on systems connected to the Internet, listing the top twenty security flaws that through exploitation make up the majority of successful attacks on computer systems connected to the Internet.

The list can be found at [7] `http://www.sans.org/top20` and consists of two parts;

| | Windows | Unix |
|---|---|---|
| 1 | Internet Information Services (IIS) | Remote Procedure Calls (RPC) |
| 2 | Microsoft Data Access Components (MDAC) Remote Data Services | Apache Web Server |
| 3 | Microsoft SQL Server | Secure Shell (SSH) |
| 4 | NETBIOS - Unprotected Windows Networking Shares | Simple Network Management Protocol (SNMP) |
| 5 | Anonymous Logon - Null Sessions | File Transfer Protocol (FTP) |
| 6 | LAN Manager Authentication - Weak LM Hashing | R-Services - Trust Relationships |
| 7 | General Windows Authentication - Accounts with No Passwords or Weak Passwords | Line Printer Daemon (LPD) |
| 8 | Internet Explorer | Sendmail |
| 9 | Remote Registry Access | BIND/DNS |
| 10 | Windows Scripting Host | General Unix Authentication - Accounts with No Passwords or Weak Passwords |

For a more detailed explanation of each vulnerability, consult the online list.

Noteworthy is the third UNIX vulnerability, SSH. Despite the fact that SSH is a cryptographic protocol, it is implemented in software, and thus vulnerable to the following types of attacks.

### 5.1.1 Types of attacks

The vulnerabilities found in the list can be divided into four groups;

### Failure to handle unanticipated input

Software that relies on correctly formatted input can in some cases be compromised by supplying input that is crafted in such a way that the implementation behaves differently than the way it was designed to, thus circumventing the security of the application. Examples of this include the directory traversal attack; requesting an url of `http://host.com/../../file` can in some cases trick webservers into traversing outside of the web root directory.

### Buffer overflows

Consider the following C code

```
int a[10];
a[20] = 42;
```

The assignment will cause some memory location that has not been allocated to the `a` variable to be changed, thus changing the behavior of some other part of the program. In cases where the "overflow" of the buffer is retrieved as input from a user, an attacker could take advantage of this and could make the software behave in a way that was not intended. In severe cases, this can also lead to an attacker changing the actual software, and thus can make the software execute arbitrary code.

To prevent buffer overflows, developers can use tools that by analyzing the code can spot potential overflows, or using programming constructs that ensure that this never happens. Some programming languages have built-in bounds-checking for code such as above, that won't allow such code to compile or execute.

### Improper configuration

Most software can be configured to be secure, or modified and configured to be secure. However, lack of configuration or incorrect usage of the software seems to be quite common.

For example, some software come with default values for accounts and passwords, which makes them a prime target for an attacker.

### Weak passwords

While the deployment of a cryptographic solution into a computer system can greatly improve the security of the system, cryptographic solutions are only as secure as its keys. Thus, if a key is found to be insecure or can easily be found, the cryptographic strength is irrelevant. Unfortunately, this problem appears to be quite common, and the current remedies to this problem seems to work poorly.

### 5.1.2 Prevention

The majority of the attacks on the vulnerability list are attacks that are targeted at a specific implementation flaws of the software, and can thus be prevented by applying the latest upgrades to the software. Other attacks could be easily prevented by not using the protocol/software, or using a secure replacement. For example, the use of FTP could be easily replaced by other sufficiently secure protocols supplying equivalent services, such as SFTP or SCP provided by the secure shell package.

However, the four groups defined above are insecurities in software that cannot and never will be prevented by an equivalent cryptographic system, since the implementation of the cryptographic system also would be subject to flaws of this kind.

# 6 Simplicity

Breaking in to a system (by other means than what was intended) has traditionally been a complex task involving steps that few people had the resources or knowledge to accomplish. However, some of the vulnerabilities on the above list are of a kind that "anyone" could succeed in attacking, including some that grant complete access to the target host. For some vulnerabilities there are even tools available on the Internet for conducting automated attacks on a range of hosts.

## 6.1 Impact

While some attacks that are directed at non-cryptographic flaws are of the kind that causes denial-of-service on the target software, most of the vulnerabilities have an impact that grants the attacker access to resource that otherwise would be restricted, and in some cases complete access to the target host. For example, both the top #1 vulnerabilities consists of software flaws that could grant the attacker complete control of the target host. In most cases a total systems compromise is far worse than for example breaking a SSL tunnel or an encryption scheme of a single client connection. The severity of these vulnerabilities stresses the importance of awareness of current threats for administrators deploying systems that potentially could be targeted by an attacker.

## 6.2 Remedy

Considering the severity of the vulnerabilities listed, the tasks needed to prevent an exploit seems quite trivial. Modern software updates usually require little or no work to successfully apply. However, the vulnerability list reflects a reality that is quite different than what would be expected, since most of the vulnerabilities would be prevented by applying software updates. It would seem that it is in everybody's interest to keep secure systems, and thus in everybody's interest to keep the software up to date, but the fact it isn't so could be explained by lack of knowledge of the current threat situation or indifference towards system integrity.

## 7 Empirical study

To verify the frequency of the attacks, a host connected to the Internet supplied with an IDS[6] and firewall blocking unused ports was set up to capture and log attempts of known attacks. While the logs of a single host surely cannot be used to determine an overall status of attack frequency, it can still be used to some extent to verify some of the most frequent attacks on random hosts connected to the Internet. The IDS deployed is a default install of Snort[8], with rule-set from the Debian Linux distribution on host 217.215.92.213. A summary from 6 Apr - 8 Apr 2003 of captured attempts of attacks can be found in appendix A.

Noteworthy is the high number of attacks targeted at the IIS web server, which seems to confirm the SANS/FBI list most frequent attack. The majority of the noted attacks targeted at the IIS web server are probably automated trojans and worms probing random Internet hosts rather than someone actively trying to break into the system. In particular, the code-red virus causes logs of this nature as it tries to propagate.

The firewall logs of connection attempts to blocked ports on the analyzed host from 8 Apr 2003 can be found in appendix B. Noteworthy is the high number of connection attempts to port 445 (SMB), which also appears high on the SANS/FBI list (windows #4).

## 8 Conclusion

While the analyzed cryptographic implementations satisfy some or all of the recommended IT security services, they are not designed to deal with the vulnerabilities identified by the SANS/FBI vulnerability list. Certainly, if cryptographically based protocols did not exist, the list of top 20 vulnerabilities would reflect an entirely different set of problems.

The vast majority of the vulnerabilities on the SANS/FBI vulnerability list take advantage of IT security problems for which cryptography is powerless. These problems have arisen due, in part, to lack of concern for system security by developers and users. Buffer overflow and exploitable default values are but two examples of irresponsible software development and implementation that easily could have been avoided. Since these vulnerabilities exist, it is up to each user to keep their software up-to-date with the latest patches and upgrades. But, a major problem for IT security seems to be that patches are not applied, despite their importance and availability.

An attacker would probably use the easiest point of entry for an attack; if some insecure software is not up-to-date, the vulnerability could lead to the attacker gaining complete access over the target host, which in many cases can

[6] Intrusion Detection System

be far worse than breaking a cryptographic system implemented on that host. The attacks on insecure software can in some cases even be done with tools downloaded from the Internet, making it trivial for "anyone" to conduct such an attack. It would seem that the combination of the simplicity of conducting the attacks with the severity of their impact would lead to a general awareness in IT security, but the vulnerability list presented in this paper shows quite the opposite since most of the attacks could easily be prevented.

Some of the problems that have arisen due to implementation flaws could have been reduced if the developer were security-aware when designing the program. Using tools to check for buffer overflows, or using programming languages that reject code that would allow an attacker to execute arbitrary code without sufficient security precautions can in some cases help the developer in counteracting attacks of this nature.

In addition to software flaws, weak passwords and improper configuration are also major problems for network security. It would seem that these problems have developed due to lack of knowledge or indifference towards system security, which is rather odd since the impact of these attacks is just as debilitating as other attacks. Some software come with default values for critical security tasks that, if left unchanged, could lead to an easy way for attacker to gain access. Default configuration of systems should always be secure, and changing the configuration in such a way that the security of the system is lowered should at least notify the user of the situation.

These vulnerabilities are of a kind that cannot, and never will be, managed by using a cryptographic equivalent solution, since the cryptographic implementation itself could contain these flaws, and the human factor still remains.

In conclusion, we find that the status of current network security is insufficient. While cryptography plays an critical role, solving many of the foreseen problems, it alone is not sufficient for overall IT security. Fortunately, our outlook on this problem is positive; it seems as though much of the complicated technical work has already been done. What remains is to make use of this work, and use it correctly. This generally calls for an increased awareness of both users and administrators surrounding security issues. With increased awareness comes increased knowledge. When both users and administrators agree on the importance of IT security, passwords will be chosen with greater care, software will be developed more carefully and configured properly, and security protocols will become more prevalent in networks. As the IT society develops and improves, so will overall network security.

# References

[1] Pfleeger, Charles P.. Security in Computing. Prentice Hall PRT, New Jersey. c. 1997.

[2] Stallings,William. Network Security Essentials: Applications and Standards. Prentice Hall, Inc., New Jersey. C. 2000.

[3] SSL Vulnerabilities. Steven McLeod and Dr. Michael Cohen.
http://www.dsd.gov.au/talks/Auscert2002.pdf

[4] SSH Transport layer protocol; IETF 2002
http://www.ietf.org/internet-drafts/draft-ietf-secsh-transport-15.txt

[5] SSH Authentication protocol; IETF 2002
http://www.ietf.org/internet-drafts/draft-ietf-secsh-userauth-16.txt

[6] The RSA algorithm and PGP; Godmar Back 1996, AMPS Winter 96, Univ. of Utah
http://citeseer.nj.nec.com/back96rsa.html

[7] Top twenty internet security vulnerabilities; SANS/FBI Oct 2002
http://www.sans.org/top20/

[8] Snort; The Open Source Network Intrusion Detection System
http://www.snort.org/

```
The log begins from: Apr 6 07:44:18
The log ends     at: Apr 8 12:09:06
Total events: 164
Signatures recorded: 16
Source IP recorded: 36
Destination IP recorded: 1
Portscan recorded: 21


The number of attacks from same host to same
destination using same method
=========================================================================
  # of
 attacks  from              to                method
=========================================================================
   48     217.215.45.186    217.215.92.213    WEB-IIS cmd.exe access : {TCP}
   16     217.232.183.61    217.215.92.213    WEB-IIS cmd.exe access : {TCP}
   6      130.239.18.173    217.215.92.213    MISC Large ICMP Packet : {ICMP}
   6      217.215.45.186    217.215.92.213    WEB-IIS CodeRed v2 root.exe access : {TCP}
   5      130.239.18.141    217.215.92.213    MISC Large ICMP Packet : {ICMP}
   3      217.208.141.166   217.215.92.213    ICMP superscan echo : {ICMP}
   3      217.215.25.197    217.215.92.213    WEB-IIS ISAPI .ida attempt : {TCP}
   3      217.215.45.186    217.215.92.213    WEB-FRONTPAGE /_vti_bin/ access : {TCP}
   3      151.26.47.116     217.215.92.213    SCAN Proxy attempt : {TCP}
   3      217.215.25.197    217.215.92.213    WEB-IIS cmd.exe access : {TCP}
   3      217.215.45.186    217.215.92.213    WEB-IIS scripts access : {TCP}
   2      217.232.183.61    217.215.92.213    WEB-IIS CodeRed v2 root.exe access : {TCP}
   2      62.183.153.184    217.215.92.213    SCAN FIN : {TCP}
   2      217.159.2.190     217.215.92.213    WEB-IIS cmd.exe access : {TCP}
   2      217.159.2.190     217.215.92.213    WEB-IIS ISAPI .ida attempt : {TCP}
   2      195.130.236.122   217.215.92.213    ICMP Destination Unreachable (Communication
Administratively Prohibited) : {ICMP}
   2      130.239.18.137    217.215.92.213    MISC Large ICMP Packet : {ICMP}
   2      217.133.194.223   217.215.92.213    WEB-IIS ISAPI .ida attempt : {TCP}
   2      216.137.3.105     217.215.92.213    SCAN Proxy attempt : {TCP}


Percentage and number of attacks from a host to a
destination
============================================================
        #  of
  %    attacks   from              to
============================================================
36.59    60      217.215.45.186    217.215.92.213
12.20    20      217.232.183.61    217.215.92.213
 3.66    6       130.239.18.173    217.215.92.213
 3.66    6       217.215.25.197    217.215.92.213
 3.05    5       130.239.18.141    217.215.92.213
 2.44    4       217.159.2.190     217.215.92.213
 2.44    4       62.183.153.184    217.215.92.213
 1.83    3       217.208.141.166   217.215.92.213
 1.83    3       151.26.47.116     217.215.92.213
 1.83    3       217.133.194.223   217.215.92.213
 1.83    3       216.137.3.105     217.215.92.213
 1.22    2       170.208.15.82     217.215.92.213
 1.22    2       217.218.250.198   217.215.92.213
 1.22    2       217.219.124.130   217.215.92.213
 1.22    2       217.168.173.174   217.215.92.213
 1.22    2       217.148.3.118     217.215.92.213
 1.22    2       217.37.147.185    217.215.92.213
 1.22    2       217.113.232.9     217.215.92.213
 1.22    2       195.130.236.122   217.215.92.213
 1.22    2       217.2.95.176      217.215.92.213
 1.22    2       217.37.195.19     217.215.92.213
 1.22    2       217.218.141.50    217.215.92.213
 1.22    2       218.145.25.76     217.215.92.213
 1.22    2       219.140.142.221   217.215.92.213
 1.22    2       209.123.49.142    217.215.92.213
 1.22    2       130.239.18.137    217.215.92.213
 1.22    2       217.118.50.94     217.215.92.213
 1.22    2       217.196.66.162    217.215.92.213
 1.22    2       217.215.206.105   217.215.92.213
 1.22    2       217.45.247.25     217.215.92.213
 1.22    2       64.159.75.220     217.215.92.213
 1.22    2       217.219.226.162   217.215.92.213
 1.22    2       217.136.125.212   217.215.92.213


Percentage and number of attacks from one host to any
with same method
==============================================================
        #  of
  %    attacks   from              method
==============================================================
29.27    48      217.215.45.186    WEB-IIS cmd.exe access : {TCP}
 9.76    16      217.232.183.61    WEB-IIS cmd.exe access : {TCP}
 3.66    6       130.239.18.173    MISC Large ICMP Packet : {ICMP}
 3.66    6       217.215.45.186    WEB-IIS CodeRed v2 root.exe access : {TCP}
 3.05    5       130.239.18.141    MISC Large ICMP Packet : {ICMP}
 1.83    3       217.215.25.197    WEB-IIS ISAPI .ida attempt : {TCP}
 1.83    3       217.208.141.166   ICMP superscan echo : {ICMP}
 1.83    3       217.215.45.186    WEB-FRONTPAGE /_vti_bin/ access : {TCP}
 1.83    3       217.215.45.186    WEB-IIS scripts access : {TCP}
 1.83    3       151.26.47.116     SCAN Proxy attempt : {TCP}
 1.83    3       217.215.25.197    WEB-IIS cmd.exe access : {TCP}
 1.22    2       62.183.153.184    SCAN FIN : {TCP}
 1.22    2       217.232.183.61    WEB-IIS CodeRed v2 root.exe access : {TCP}
 1.22    2       217.133.194.223   WEB-IIS ISAPI .ida attempt : {TCP}
 1.22    2       217.159.2.190     WEB-IIS ISAPI .ida attempt : {TCP}
 1.22    2       130.239.18.137    MISC Large ICMP Packet : {ICMP}
 1.22    2       217.159.2.190     WEB-IIS cmd.exe access : {TCP}
 1.22    2       195.130.236.122   ICMP Destination Unreachable (Communication
Administratively Prohibited) : {ICMP}
 1.22    2       216.137.3.105     SCAN Proxy attempt : {TCP}


Percentage and number of attacks to one certain host
=================================================================
        #  of
  %    attacks   to                method
=================================================================
53.66    88      217.215.92.213   WEB-IIS cmd.exe access : {TCP}
14.02    23      217.215.92.213   WEB-IIS ISAPI .ida attempt : {TCP}
 7.93    13      217.215.92.213   MISC Large ICMP Packet : {ICMP}
 4.88    8       217.215.92.213   WEB-IIS CodeRed v2 root.exe access : {TCP}
 4.88    8       217.215.92.213   SCAN Proxy attempt : {TCP}
 2.44    4       217.215.92.213   WEB-IIS scripts access : {TCP}
 2.44    4       217.215.92.213   INFO - Possible Squid Scan : {TCP}
 2.44    4       217.215.92.213   WEB-FRONTPAGE /_vti_bin/ access : {TCP}
 1.83    3       217.215.92.213   ICMP superscan echo : {ICMP}
 1.22    2       217.215.92.213   SCAN FIN : {TCP}
 1.22    2       217.215.92.213   ICMP Destination Unreachable (Communication
Administratively Prohibited) : {ICMP}


The distribution of attack methods
===============================================
        #  of
  %    attacks   method
```

```
===============================================
53.66    88      WEB-IIS cmd.exe access
14.02    23      WEB-IIS ISAPI .ida attempt
 7.93    13      MISC Large ICMP Packet
 4.88    8       WEB-IIS CodeRed v2 root.exe access
 4.88    8       SCAN Proxy attempt
 2.44    4       WEB-IIS scripts access
 2.44    4       INFO - Possible Squid Scan
 2.44    4       WEB-FRONTPAGE /_vti_bin/ access
 1.83    3       ICMP superscan echo
 1.22    2       ICMP Destination Unreachable (Communication Administratively Prohibited)
 1.22    2       SCAN FIN


Portscans performed to/from HOME_NET
===================================
  # of
 attacks  from
===================================
 19      62.183.153.184
```

```
Firewall log 20030408

fwlogwatch summary
Generated Tue Apr 08 14:11:50 CEST 2003 by root.
55 (and 144 older than 86400 seconds) of 379 entries in the file "/var/log/messages" are
packet logs, 11 have unique characteristics.
First packet log entry: Apr 07 15:22:44, last: Apr 08 13:58:48.
All entries were logged by the same host: "ast".
All entries are from the same chain: "-".
All entries have the same target: "-".
All entries are from the same interface: "eth1".

1 tcp packet port 1 (tcpmux) SYN
4 tcp packets port 22 (ssh) SYN
1 tcp packet port 22 (ssh) ---r--
1 tcp packet port 111 (sunrpc) SYN
4 tcp packets port 113 (auth) SYN
1 tcp packet port 135 (-) SYN
3 tcp packets port 443 (https) SYN
29 tcp packets port 445 (smb) SYN
4 tcp packets port 5432 (postgres) SYN
1 tcp packet port 5432 (postgres) ---r--
6 udp packets port 135 (-) -

fwlogwatch summary
Generated Tue Apr 08 14:11:50 CEST 2003 by root.
55 (and 144 older than 86400 seconds) of 379 entries in the file "/var/log/messages" are
packet logs, 21 have unique characteristics.
First packet log entry: Apr 07 15:22:44, last: Apr 08 13:58:48.
All entries were logged by the same host: "ast".
All entries are from the same chain: "-".
All entries have the same target: "-".
All entries are from the same interface: "eth1".

Apr 07 22:10:24 1 tcp packet from 213.215.152.181 port 1 (tcpmux) SYN
Apr 08 13:41:46 4 tcp packets from 130.237.227.34 port 22 (ssh) SYN
Apr 08 13:42:10 1 tcp packet from 130.237.227.34 port 22 (ssh) ---r--
Apr 07 21:26:29 1 tcp packet from 200.72.24.12 port 111 (sunrpc) SYN
Apr 07 18:15:40 4 tcp packets from 212.181.52.4 port 113 (auth) SYN
Apr 07 15:22:44 1 tcp packet from 217.208.141.166 port 135 (-) SYN
Apr 07 22:11:55 1 tcp packet from 80.181.183.15 port 443 (https) SYN
Apr 07 21:36:18 2 tcp packets from 218.5.77.223 port 443 (https) SYN
Apr 08 00:41:14 2 tcp packets from 24.91.229.4 port 445 (smb) SYN
Apr 07 17:04:06 3 tcp packets from 64.134.19.75 port 445 (smb) SYN
Apr 07 23:00:31 3 tcp packets from 66.215.137.20 port 445 (smb) SYN
Apr 08 13:58:39 3 tcp packets from 68.60.89.167 port 445 (smb) SYN
Apr 08 06:23:30 3 tcp packets from 80.116.14.154 port 445 (smb) SYN
Apr 07 21:23:33 3 tcp packets from 142.51.18.201 port 445 (smb) SYN
Apr 07 23:38:49 3 tcp packets from 195.235.134.163 port 445 (smb) SYN
Apr 07 21:26:12 3 tcp packets from 209.99.240.90 port 445 (smb) SYN
Apr 07 22:59:02 3 tcp packets from 211.18.239.39 port 445 (smb) SYN
Apr 08 00:52:30 3 tcp packets from 219.38.212.32 port 445 (smb) SYN
Apr 08 04:30:14 4 tcp packets from 130.237.226.103 port 5432 (postgres) SYN
Apr 08 04:30:52 1 tcp packet from 130.237.226.103 port 5432 (postgres) ---r--
Apr 08 02:07:02 6 udp packets from 67.83.127.213 port 135 (-) -
```