Design and Implementation of Personal Firewalls for Handheld Devices

Jianyong Huang, Willy Susilo and Jennifer Seberry
Centre for Computer Security Research
School of Information Technology and Computer Science
University of Wollongong, Wollongong, NSW 2522, Australia
Email: {jyh33, wsusilo, jennie}@uow.edu.au

Abstract

Personal Digital Assitants (PDAs) have become one of the important tools in our life. Their popularity are due to their small size and mobility which enable them to be carried anywhere. Along with their popularity, handheld devices are starting to become the target for the attackers, who are mainly interested in gaining the data stored in handheld devices. Therefore, security of handheld devices have attracted a lot of attention in an effort to protect the sensitive information stored in handheld devices. Securing handheld devices is a daunting task. It requires a careful design since the devices have very limited computational power and battery life. In this paper, we aim to review the security threats to handheld computers and propose several possible solutions. We performed some experiments to test our proposed solution in an iPaq Pocket PC.

1 Introduction

Personal Digital Assistants (PDAs) have become one of the important tools for computing, communications and multimedia applications. Their popularity are due to their size and mobility which enable the PDAs to be carried everywhere. To obtain the information available, PDAs can be connected to the Internet. This situation will make the risk of having attacks on PDAs higher, especially if we cannot really know who are the other people that are using the Internet, and they might be targetting to attack our PDAs as their resources.

Recently, the PDA itself has evolved with technology and an explicit example of this technology is a Smart Phone. A smart phone can perform the task that a PDA can do, but in addition to that, the phone can be used to communicate. The other concern appears with this technology since recently some children have a smart phone with them. The latest psychological threat has shown that children bullying has caused a big problem [5], in which a child is abused by being sent several MMS (Multimedia Message Service) or SMS (Short Message Service) that indicates child bullying. According to the statistical data, at least sixteen children commit suicide each year after bullying in the United Kingdom [3], which is of great concern at this moment. In this paper, we refer a smart phone as a kind of handheld device, and we use the term *handheld device* to represent either a PDA or a smart phone.

Another issue appears from the malicious code which is related to the ability of a handheld device to be connected to the Internet [9]. The Internet connections will make the Pocket PC enabled handheld devices vulnerable to attackers and viruses. Unlike Palm OS, Pocket PCs are not known yet to become the victim of the virus or Trojan horse. From the experience, we learn that Palm OS is very vulnerable against these type of attacks [8]. For example, *Palm.Liberty.A* is the first Trojan horse program that attacked was reported to attack Palm OS successfully. The Liberty Crack Program could delete all of the applications on a Palm PDA once it is activated [12]. Other discovered viruses, such as *Vapor* [12] and *Phage* [12], also could make Palm devices infected. *Vapor* could change file attributes to hidden, and as a result, you cannot see your files even they are still in your handheld. *Phage* works by overwriting Palm's executable files. By replicating itself, this virus keeps spreading to application files until they are all affected. All these viruses aimed at handheld devices show that handheld computers are susceptible to malicious attacks like other computing platforms.

In this paper, we aim to investigate the threats on handheld devices and propose a solution to protect against these threats. In particular, we are interested to protect the information that flows in the handheld devices. We aim to analyze the contents of the information that handheld devices have and determine whether the contents are appropriate to be protected by the handheld user. Securing handheld devices is a daunting task. The security mechanism must be carefully design so that it is suitable for a handheld device, since it has a very limited computational power, memory and battery life.

The rest of this paper is organized as follows. In section 2, we briefly review the existing handheld devices. We

concentrate at the two main operating system available, namely Palm OS and Windows CE in this section, although we are aware that there are several other operating systems that the current handheld devices use such as Linux or Symbian OS. In section 3, we provide the basic overview of a firewall and illustrate the importance of a personal firewall to protect such a system. In section 5, we describe the design and implementation of personal firewall in a Pocket PC operating system. We note that this design can be easily adopted to another operating system such as Symbian or Linux. Section 6 concludes the paper.

Our Contributions
In this paper, firstly we examine the threats to handheld devices. We also propose two feasible solutions to secure handheld devices. One solution is to use a desktop computer to act as a bastion host to protect handheld devices. The second proposed solution is by building a personal firewall for handheld devices. We will highlight the difficulty of building such a system and proceed with our design and implementation on a Pocket PC handheld device.

2 Handheld Devices

In this section, we briefly describe two main operating systems used in handheld devices, namely Windows CE and PalmOS, which currently dominate the market. A Windows CE device (which is usually referred to a Pocket PC device) uses a proprietary operating system from Microsoft which is known as the Windows CE operating system. The latest version of Windows CE operating system is Pocket PC 2002 which is designed to run in either a Strong ARM or an Intel XScale processor. Palm operating system has evolved to its current version Palm OS 5 which is used by several devices such as Palm Tungsten or Sony Clie handheld. There are several other operating systems that are available for handheld devices. Psion PDA uses EPOC and MSI's Eznow PDA uses Mine OS. Sharp Zaurus uses Embedded Linux with Linux kernel 2.4.x as its operating system. Newton OS is used by Apple's MessagePad and eMate operating system. The current smart phones, such as Ericsson P800 or Nokia 7650, uses the Symbian operating system.

In the following discussion, we concentrate on Pocket PC devices. Any handheld devices will have similar features.

A Pocket PC can be connected to the Internet via several methods: using a modem connection, using a wired or wireless LAN connection, using a Bluetooth connection, an Infrared connection with a cellular phone, or using a TCP/IP connection with the assistance of a desktop PC. As we mentioned in section 1, the Internet connections will make the Pocket PCs vulnerable to attackers and viruses. It is noted in [16] [15] that firewalls are important to ensure security in wireless networks.

In the case of a Pocket PC device, a synchronization software called *ActiveSync* is used to enable the communication between a Pocket PC PDA and a desktop computer. This synchronization ensures that both the PDA and desktop computer have the latest information. The current version of ActiveSync is 3.7. Pocket PCs can be connected with desktop PCs by using infrared, Ethernet networking or modems [13]. ActiveSync offers a convenient way to Synchronize data between a PDA and a desktop PC, but it may also potentially deliver viruses between a PDA and a PC. Viruses may spread via email, software installation or file transfer from a desktop PC to a PDA and vice versa. McAfee's VirusScan for Pocket PC can protect desktop computers from infection by scanning the files on handheld devices whenever they are connected to desktop PCs [20]. PC-cillin for Wireless 2.0 [7] also offers virus protection to handhelds running Palm, Pocket PC and EPOC. PC-cillin for Wireless works by scanning data and devices for viruses during beaming, sychronization, or Internet access.

3 Review of Firewalls

A *Firewall* is a component or set of components that restricts access between a protected network and the Internet, or between other sets of networks [24]. A typical scenario of a firewall is illustrated in Figure 1. A bastion host is a computer system between a protected network and the Internet. The bastion host is designed to defend against attacks on the protected network. The perimeter network, also called De-Militarized Zone, is a network between an internal network and an external network. The perimeter network offers additional protection for the internal network.

There has been much work contributed to firewalls and their applications [2] [4]. In the next section, we will describe the types of firewalls.

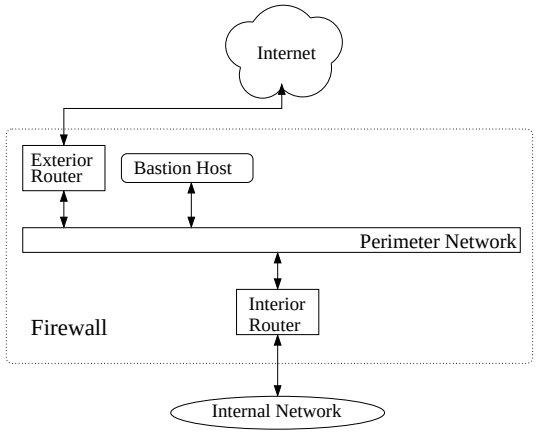

Figure 1. *Firewall Architecture*

3.1 Types of Firewalls

According to [24], there are three basic types of firewalls, known as packet-filtering firewall, circuit-level firewall and

application-level firewall. The brief description of each firewall type will be described in the following section.

3.1.1 Packet-Filtering Firewall

Packet-filtering firewalls allow or block packets based on some or all of these four fields: the source IP address, the destionation IP address, the TCP/UDP source port and the TCP/UDP destination port[24]. To transfer data across a network, the data must be divided into small pieces called packets. A *packet*, which is the fundamental unit of data transfer on the Internet, contains only a few hundred bytes of data. In TCP/IP model, an IP (Internet Protocol) packet consists of two parts: the IP header and the IP body. Two important pieces of information, the IP source address and destination address, are contained in the IP header. At TCP layer, a TCP packet is contained in the body of an IP packet. The TCP packet is made up of the TCP header and the TCP body. The source port and destination port are stored in the TCP header. The body of an IP packet might contain a UDP packet. A UDP packet also contains a UDP header and a UDP body. The UDP source port and destination port are stored in the UDP header. Packet-filtering firewall operates at OSI network layer.

Figure 2 shows a sample packet-filtering rule that allows incoming and outgoing SMTP (Simple Mail Transfer Protocol) connections so that email can be delivered. Rule 1 allows an external host to send a request to port 25 on a server inside the protected network. Rule 2 would allow that server on the protected network to reply to the external host. Rule 3 and 4 allow the SMTP connection in the reverse direction, where the SMTP server on the protected network wants to establish a connection to port 25 on an SMTP server on the external network. Rule 5 disallows any other connections.

The advantage of the packet-filtering firewall is that it is simple and easy to implement as all filtering rules can be configured in a network router. The disadvantage is packet filters make decision only based on packet header information, not on the payload session of the packet. Once the packet filter fails, the whole protected internal network will be transparent to attackers.

Rule	In / Out	Pro− tocol	Source Address	Dest. Address	Source Port	Dest. Port	Action
1	In	TCP	External	Internal	>=1024	25	Permit
2	Out	TCP	Internal	External	25	>=1024	Permit
3	Out	TCP	Internal	External	>=1024	25	Permit
4	In	TCP	External	Internal	25	>=1024	Permit
5	*	*	*	*	*	*	Deny

Figure 2. *A Sample Filtering Rule*

3.1.2 Circuit-Level Firewall

A circuit-level firewall establishes a circuit between the client and the server without interpreting the application protocol [24]. The advantage of a circuit-level firewall is that it supports a wide range of different protocols using TCP and protects against packet framentation problem. The disadvantage of a circuit-level firewall is that it relays TCP connections based on their source and destination addresses but cannot provide control on whether the information of these connections is legitimate or not.

The SOCKS package [10] is an example of the circuit-level firewall. SOCKS is a networking proxy protocol that can be used to create connections through a firewall. We refer readers to [10] to get more information about SOCKS.

3.1.3 Application-Level Firewall

An application-level firewall is a proxy server that understands the particular application it is providing proxy services for, and it intercepts network traffic for a specific kind of application [24]. Application-level firewalls are generally implemented to work at application level. The main difference between the packet-filtering firewall and the application-level firewall is that the application-level firewall must understand the application. Application-level firewalls are more secure than packet-filtering firewalls as they can be programmed to allow or deny network traffic based on information contained in the payload session of the packet, not just the header information [18]. Moreover, the application-level firewall does not allow direct connections between an internal host of the protected network and an external host on the outside network. All communications between an internal host and an external host are handled by the proxy. Thus, the application-level firewall hides the network information of the internal network and reduces potential threats to the internal network. The disadvantage of application-level firewalls is each proxy service requires its own proxy. For example, FTP, HTTP and TELNET requires their own proxies, namely FTP proxy, HTTP proxy and TELNET proxy. As a result, the application-level firewall must understand each proxy service it provides.

The TIS (Trusted Information Systems) Firewall Toolkit (FWTK)[6] is an instance of the application-level firewall. FWTK inludes a number of proxy servers of differnet types such as FTP, Telnet, Rlogin, HTTP, Gopher, SMTP and NNTP.

3.2 Personal Firewalls

A personal firewall is a software application used to protect a single computer. Personal firewalls are essential necessary for those users who have connections such as Digital Subscriber Line (DSL) or cable modem. These *always-on* connections normally have static Internet Protocol (IP) addresses that makes the system vulnerable to potential hackers.

There are two basic types of personal firewalls on the market. The first type is application-level firewall that

monitors inbound and outbound network traffic and applies access control on internet traffic. The other type is packet-filtering firewall that monitors the network traffic based on their TCP/IP header information [25].

Examples of existing Application-level firewall are Symantec[21]'s *Norton Personal Firewall*, McAfee[11]'s Personal Firewall and Zone Lab[23]'s *ZoneAlarm*. Products of Packet-filtering firewall are Network Flight Recorder[17]'s *BackOfficer Friendly* and NetworkICE[22]'s *BlackICE Defender* [25].

We refer the reader to [1] for a complete comparison between the existing products.

4 Possible Solutions

To resolve problems described in section 1 and 2, there are some possible solutions that can be deployed in handheld devices. The first solution is to use a desktop computer to act as a bastion host to protect the handheld device. In the second proposed solution, we provide a better protection mechanism, i.e. by incorporating a personal firewall for handheld devices.

4.1 Desktop Computers Act as Bastion Hosts

One possible solution is let a desktop computer act as a *bastion host*. ActiveSync 3.6 provides a feature, called *Pass-Through Connection*, to enable a Pocket PC 2002 device access the Internet while the device is connected to the desktop PC using serial, infrared, or USB connections. When the Pocket PC is connected to a desktop PC, people can use Interenet Explorer to browse web sites or check emails. As a Pocket PC handheld device can connect to the Internet via a TCP/IP connection with the assistance of a desktop computer, we can have the desktop PC stand between the handheld device and the Internet. Thus, the desktop PC can control the handheld's inbound and outbound network traffic based on users' security rules. The architecture is shown in Figure 3. However, this scenario is not perfect as the desktop PC must be trusted. Moreover, due to its mobility feature, the handheld device can be carried anywhere and to find a trusted desktop PC would create a problem.

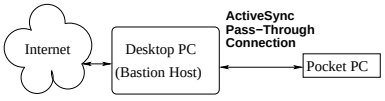

Figure 3. *A Desktop PC Acts as a Bastion Host*

4.2 A Better Solution: Personal Firewalls

A better solution is that a personal firewall is built into a handheld computer. With a personal firewall, the user can adjust his requirements or policies on handling the cases that he will encounter. The personal firewall can be tuned to suit the user's need. In the next section, we will describe the design and implementation of a personal firewall in a Pocket PC handheld device.

5 Design and Implementation of Personal Firewalls

Before describing our design and implementation of personal firewalls in handheld devices, we recall the possible technologies that can be used in Pocket PC 2002 operating system. Theoretically, this technology can be employed to perform the packet filtering required in the system. However, as we will justify in this section, there is no direct API call that can be used in the implementation provided by Microsoft and therefore, some other alternatives need to be investigated.

5.1 Packet Filtering in Pocket PC 2002

Windows 2000 offers an important network Application Programming Interface (API), known as Packet Filtering API (PFAPI). PFAPI is defined in *fltdefs.h* in Microsoft Platform SDK. PFAPI is quite useful as it enables programmers to filter out TCP/IP packets by using PFAPI, programmers can easily control inbound and outbound network traffic. Plooy showed an implementation of a flexible packet-filtering firewall, called *netBlock*, with PFAPI in a couple of hundred lines of C++ [19]. To implement a packet-filtering firewall, the netBlock only uses five (out of a total of sixteen) PFAPI functions: PfCreateInterface(), PfAddFiltersToInterface(), PfBindInterfaceToIPAddress(), PfUnbindInterface(), and PfDeleteInterface().

Unfortunately, Pocket PC 2002 Software Development Kit (SDK) does not provide PFAPI, which makes implementation of packet filtering in Pocket PC 2002 much more difficult than in Windows 2000. One possible solution to implement a packet-filtering firewall in Pocket PC 2002 is to develop an intermediate Network Driver Interface Specification (NDIS) driver, which will be outlined in the next section.

5.2 Review of NDIS

NDIS stands for *Network Driver Interface Specification*. Microsoft and 3Com jointly developed NDIS, and the current version of NDIS is 5.x. Microsoft networking protocols use the NDIS interface to communicate with network card drivers. The NDIS implementation on Windows CE is a subset of the NDIS 4.0 implementation used on Windows NT[14].

The Windows CE NDIS architecture consists of several components: the NDIS wrapper component, Protocol Drivers, Network Interface Card (NIC) Miniport Drivers, and Intermediate Miniport Drivers [14]. Figure 4 shows the network architecture in Windows CE. The NDIS wrapper component is contained in NDIS.DLL file and provides the operating environment for drivers that use NDIS.

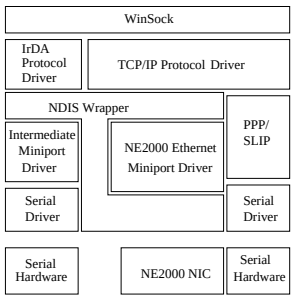

Figure 4. *Windows CE Network Architecture*

Two NDIS protocol drivers, TCP/IP and IrDA, are supported by Windows CE. Windows CE NDIS protocol driver are exposed to applications through the Windows Sockets (Winsock) API. A protocol driver allocates packets, copies data from sending applications into packets, and sends these packets to the lower-level driver via the NDIS wrapper component. The protocol driver also provides a lower level interface (specified by NDIS) to receive incoming packets from the Miniport driver. By using the NDIS interface, the NDIS protocol driver communicates with underlying miniport driver and bind to network adapter. A NDIS miniport driver has two basic functions. One function is to manage a network interface card, including sending and receiving data through the NIC. Another function is to provide an interface to communicate with higher-level drivers, such as intermediate drivers and transport protocol drivers. NDIS intermediate drivers include a protocol driver interface at their lower edge and a miniport driver interface at their upper edge. An intermediate miniport driver is typically used to *filter packets*, translate between different network media and balance packet transmission across more than one NIC. Thus, it is a good idea to implement packet filtering at an intermediate miniport driver as it can capture and filter network packets.

5.3 Design and Implementation of Personal Firewall on Pocket PC

One important role of an intermediate driver is to pass network packets between a protocol driver and a miniport driver. So, it is a good place to filter network packets at an imtermediate driver.

Writing an intermediate driver for Windows CE needs the Windows CE Driver Development Kit (DDK). Windows CE DDK is included in Windows CE Platform Builder 3.0. In our design, the underlying miniport is the NE2000 driver, and the upper protocol is TCP/IP. After the miniport driver loads, the intermediate driver loads and binds to the network adapter. A protocol driver then sends down requests to query the capabilities of the adapter. The intermediate driver simply passes these requests down to the adapter, and send the responses back up to the protocol driver.

Every NDIS driver must provide a function called **DriverEntry**. DriverEntry is called by the system to load the driver, and is responsible for initializing the driver. The DriverEntry builds the relationship between the intermediate driver and the NDIS library. The intermediate driver performs two fundamental tasks in its DriverEntry function. One task is to call function NdisMInitializeWrapper to tell the NDIS library that the driver is about to regester itself. Another task is to register the intermediate driver's version number, then call function NdisIMRegisterLayeredMiniport and NdisRegisterProtocol to register entry points with the NDIS library. Figure 5 shows the procedure for registering an intermediate driver and initializing the NDIS library.

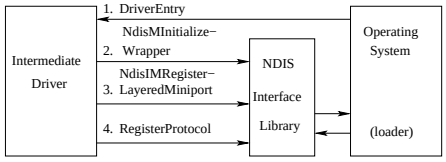

Figure 5. *Register an Intermediate Driver*

5.3.1 Filter Incoming Network Packets

When the network adapter receives an incoming packet, it calls the intermediate driver to handle this packet. The intermediate driver then copies this packet into its own preallocated packet, and send the new packet to a protocol driver. As each packet contains its source IP address and destination IP address in the header session, the intermediate driver can extract these two addresses from each packet and perform filtering based on source and destination IP addresses. In other words, the intermediate driver can filter incoming network packets before passing them up to the protocol driver (shown in Figure 6). Figure 7 illustrates an example of filtering incoming network traffic.

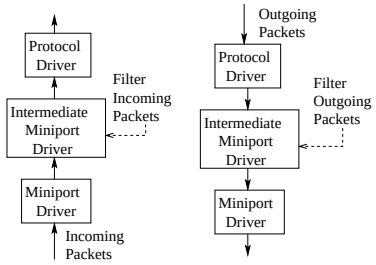

Figure 6. *Filter Incoming and Outgoing Packets*

5.3.2 Filter Outgoing Network Packets

When the intermediate driver receives an outgoing packet from the protocol driver, it copies this packet to its preallocated packet, and send this new copy down to the miniport driver. The miniport driver then send this packet out. As mentioned in 5.3.1, each network packet includes its

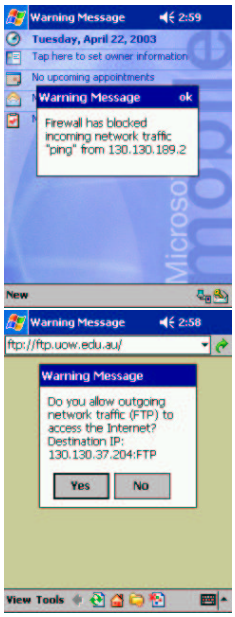

Figure 7. *Filtering Incoming and Outgoing Network Traffic*

source and destination IP addresses in its header session, the intermediate driver can extract these two IP addresses from the packet header and perform the filtering based on source and destination IP addresses. Thus, the intermediate driver can filter the outgoing network packets before passing them down to the adapter (shown in Figure 6). Figure 7 illustrates an example of filtering outgoing network traffic.

6 Conclusion

Handheld computers are important computing tools due to their size and light weight. However, handheld devices also become attackers' targets as they become more popular. We reviewed some security threats to handheld computers and proposed several possible solutions. The first solution is to use a desktop computer as a bastion host. The second one is to build a personal firewall in a handheld computer. We highlighted the difficulty of building a personal firewall in a Pocket PC operating system. We described an implementation in an iPAQ Pocket PC which is equipped with a Strong ARM processor with 206 MHz.

References

[1] Firewalls: how secure are they? *Australian PC USER*, pages 52–56, September 2002.

[2] Steven M. Bellovin and William R. Cheswick. Network firewalls. *IEEE Communications Magazine*, pages 50–57, September 1994.

[3] Bullyonline. Bullying and suicide. *http://www.bullyonline.org/stress/suicide.htm.*

[4] William R. Cheswick, Steven M. Bellovin, and Aviel D. Rubin. *Firewalls and Internet Security: Repelling the Wily Hacker*. Addison Wesley Professional, second edition, 2003.

[5] Leah Creighton. Text message bullies prey on children. *The Sunday Telegraph*, November 3, 2002.

[6] Trusted Information Systems (TIS) Firewall Toolkit (FWTK). *http://www.fwtk.org.*

[7] Trend Micro Inc. *PC-cillin for Wireless.* http://www.trendmicro.com/en/products/desktop/pc-cillin/evaluate/overview.htm.

[8] Kingpin and Mudge. Security analysis of the palm operating system and its weaknesses against malicious code threats. *Proceeding of the 10th USENIX Security Symposium*, pages 135–151, 2001.

[9] Neal Leavitt. Malicious code moves to mobile devices. *Computer*, 33(12):16–19, December 2000.

[10] Permeo Technologies Ltd. *The Source for SOCKS Technology.* http://www.socks.permeo.com/.

[11] McAfee Security. http://www.mcafee.com.

[12] McAfee Security. *Virus Information. http : //www.mcafee.com/anti − virus/default.asp.*

[13] Frank McPherson. *How to Do Everything with Your Pocket PC.* McGraw-Hill/Osborne, 2nd edition edition, 2002.

[14] Microsoft Corporation. *Microsoft Windows CE Developer's Kit (Microsoft Professional Editions).* Microsoft Press, 1999.

[15] Sandra Kay Miller. Facing the challenge of wireless security. *Computer*, 34(7), July 2001.

[16] U. Murthy, O.Bukhres, W. Winn, and E. Vanderdez. Firewalls for security in wireless networks. *Proceedings of the 31st Hawaii International Conference on System Science (HICSS'98)*, 1998.

[17] NFR Security. http://www.nfr.com/.

[18] Terry William Ogletree. *Practical Firewalls.* QUE, June 2000.

[19] Ton Plooy. Packet filtering with iphlpapi.dll. *Windows Developer Magazine*, Volume 11 Number 10, October 2000.

[20] McAfee Security. *VirusScan Wireless.* http://www.mcafeeb2b.com/products/virusscan-wireless/default.asp.

[21] Symantec Corporation. http://www.symantec.com/.

[22] Internet Security Systems. http://www.iss.net/.

[23] Zone Labs Inc. http://www.zonelabs.com/.

[24] Elizabeth D. Zwicky, Simon Cooper, and D. Brent Chapman. *Building Internet Firewalls.* O'Reilly & Associates, 2nd edition, 2000.

[25] Tina Zych. Personal firewalls: What are they, how do they work? *http://www.sans.org*, August 22, 2000. 11.