Intrusion Detection Systems (IDS) Part I - (network intrusions; attack symptoms; IDS tasks; and IDS architecture)

www.windowsecurity.com/articles-tutorials/intrusion_detection/Intrusion_Detection_Systems_IDS_Part_I__network_intrusions_attack_symptoms_IDS_tasks_and_IDS_architecture.html

by **Przemyslaw Kazienko & Piotr Dorosz** [Published on *7 April 2003* / Last Updated on *7 April 2003*]

Due to a growing number of intrusions and since the Internet and local networks have become so ubiquitous, organizations increasingly implementing various systems that monitor IT security breaches. Intrusion Detection Systems (IDS) are those that have recently gained a considerable amount of interest. This is an introductory article to this topic. It gives an overview of several types of detectable attacks, symptoms that help in intrusion detection, describes IDS tasks, different architectures and concepts in this field.

Additional information on Intrusion Detection Systems may be found within the WindowSecurity.com IDS FAQ.

What is an Intrusion Detection System?

An Intrusion Detection System (abbreviated as IDS) is a defense system, which detects hostile activities in a network. The key is then to detect and possibly prevent activities that may compromise system security, or a hacking attempt in progress including reconnaissance/data collection phases that involve for example, port scans. One key feature of intrusion detection systems is their ability to provide a view of unusual activity and issue alerts notifying administrators and/or block a suspected connection. According to Amoroso [1], intrusion detection is „a process of identifying and responding to malicious activity targeted at computing and networking resources". In addition, IDS tools are capable of distinguishing between insider attacks originating from inside the organization (coming from own employees or customers) and external ones (attacks and the thread posed by hackers).

The Polish term “wykrywanie intruzow ” does not really translate the English term “intrusion detection” [1]). “Wykrywanie” (detection) of persons is a problem, of course (see Mukherjee and Heberlein [2]), but the very problem is associated with their criminal activities, for instance intrusions and/or security breaches and therefore “detection of intruder activity” would explain such things better. Obviously, intruder identification is considered an important ability (very often associated with the method of response against an intrusion) and task of an IDS, but rather less significant than the intrusion detection itself.

What is not an IDS?

Contrary to popular market(ing) belief and terminology employed in the literature on intrusion detection systems, not everything falls into this category. In particular, the following security devices are NOT IDS:

- Network logging systems used, for example, to detect complete vulnerability to any Denial of Service (DoS) attack across a congested network. These are network traffic monitoring systems.
- Vulnerability assessment tools that check for bugs and flaws in operating systems and network services (security scanners), for example Cyber Cop Scanner.
- Anti-virus products designed to detect malicious software such as viruses, Trojan horses, worms, bacteria, logic bombs. Although feature by feature these are very similar to intrusion detection systems and often provide an effective security breach detection tool.
- Firewalls {1}
- Security/cryptographic systems, for example VPN, SSL, S/MIME, Kerberos, Radius etc.

A taxonomy of attacks and intrusions

Since intrusion detection systems deal with hacking breaches, let us take a closer look at these dangerous activities. To assist in the discussion of their taxonomy, some definitions will be helpful although they may vary [1]:

- Intrusion – a series of concatenated activities that pose threat to the safety of IT resources from unauthorized access to a specific computer or address domain;
- Incident – violation of the system security policy rules that may be identified as a successful intrusion;
- Attack – a failed attempt to enter the system (no violation committed).
- Modeling of intrusions – a time-based modeling of activities that compose an intrusion. The intruder starts his attack with an introductory action followed by auxiliary ones (or evasions) to proceed to successful access; in practice, any attempts undertaken during the attack by any person, for example by the IT resource manager can be identified as a threat.

Generally, attacks can be categorized in two areas:

- Passive (aimed at gaining access to penetrate the system without compromising IT resources),
- Active (results in an unauthorized state change of IT resources).

In terms of the relation intruder-victim, attacks are categorized as:

- Internal, coming from own enterprise's employees or their business partners or customers,
- External, coming from outside, frequently via the Internet.

Attacks are also identified by the source category, namely those performed from internal systems (local network), the Internet or from remote dial-in sources). Now, let us see what types of attacks and abuses are detectable (sometimes hardly detectable) by IDS tools to put them in the ad-hoc categorization. The following types of attacks can be identified:

- Those related to unauthorized access to the resources (often as introductory steps toward more sophisticated actions):
  - Password cracking and access violation,
  - Trojan horses,
  - Interceptions; most frequently associated with TCP/IP stealing and interceptions that often employ additional mechanisms to compromise operation of attacked systems (for example by flooding); man in the middle attacks),
  - Spoofing (deliberately misleading by impersonating or masquerading the host identity by placing forged data in the cache of the named server i.e. DNS spoofing)
  - Scanning ports and services, including ICMP scanning (Ping), UDP, TCP Stealth Scanning TCP that takes advantage of a partial TCP connection establishment protocol.) Etc.
  - Remote OS Fingerprinting, for example by testing typical responses on specific packets, addresses of open ports, standard application responses (banner checks), IP stack parameters etc.,
  - Network packet listening (a passive attack that is difficult to detect but sometimes possible),
  - Stealing information, for example disclosure of proprietary information,
  - Authority abuse; a kind of internal attack, for example, suspicious access of authorized users having odd attributes (at unexpected times, coming from unexpected addresses),
  - Unauthorized network connections,
  - Usage of IT resources for private purposes, for example to access pornography sites,

    - Taking advantage of system weaknesses to gain access to resources or privileges,

- Unauthorized alteration of resources (after gaining unauthorized access):
    - Falsification of identity, for example to get system administrator rights,
    - Information altering and deletion,
    - Unauthorized transmission and creation of data (sets), for example arranging a database of stolen credit card numbers on a government computer (e.g. the spectacular theft of several thousand numbers of credit cards in 1999),
    - Unauthorized configuration changes to systems and network services (servers).
- Denial of Service (DoS):
    - Flooding – compromising a system by sending huge amounts of useless information to lock out legitimate traffic and deny services:
        - Ping flood (Smurf) – a large number of ICMP packets sent to a broadcast address,
        - Send mail flood - flooding with hundreds of thousands of messages in a short period of time; also POP and SMTP relaying,
        - SYN flood – initiating huge amounts of TCP requests and not completing handshakes as required by the protocol,
        - Distributed Denial of Service (DDoS); coming from a multiple source,
    - Compromising the systems by taking advantage of their vulnerabilities:
        - Buffer Overflow, for example Ping of Death — sending a very large ICMP (exceeding 64 KB),
        - Remote System Shutdown,
- Web Application attacks; attacks that take advantage of application bugs may cause the same problems as described above.

It is important to remember, that most attacks are not a single action, rather a series of individual events developed in a coordinated manner.

You are at risk

To recognise possible attacks, examine systems for any abnormal behavior [3]. This may be helpful in detecting real attacks. Let us take a closer look at the types of symptoms that are helpful in tracing intruders.

Utilizing known vulnerabilities

In most cases, any attempt to take advantage of faults in organization security systems may be considered as an attack and this is the most common symptom of an intrusion. However the organization itself may “facilitate” the task of attackers, using tools which aid in the process of securing its network – so called security and file integrity scanners. They operate either locally (assisting system administrators in vulnerability assessment) or remotely but may also be deliberately used by intruders.

Since these tools are often a double-edged sword and are available for both the users and hackers, accurate monitoring of the usage of file integrity scanners and known vulnerability scanners is needed, to detect attacks in progress or trace damages from successful attacks. Hence, the following technical issue arises:

- Detection of file integrity scanners. The available file integrity testing tools operate in a systematic manner so that it is possible to use modeling techniques and specialized tools for detection purposes, for example the anti-SATAN software, Courtney.

- A good correlation between scanning and usage is required – scanning for flaws may further use a service featuring such flaws, this may be a precursor of an attack to come.

Recurrent abnormal network activity

An intruder actually trying to compromise a system often uses a large number of exploits and makes many unsuccessful attempts. His activities differ from those of the user working with the system [4] Mans00]. Any penetration-testing tool should be able to identify suspicious activities after a certain threshold has been exceeded. Then, an alert may be produced and diffused. This passive technique allows detection of intruders without discovering a clear picture of the event (exploits involved, tools, services, software configuration, etc.), by only quantitatively examining network activities.
Passive methods used in intrusion detection are driven from databases of recurrent attack signatures that should consider the following technical aspects [1]:

- Repetition thresholds to help distinguish between legal and suspicious activities (that trigger alerts). Network activities can be identified using multiple parameter values derived, for example, from the user profile or Session State.
- Time between repeat instances – a parameter to determine the time to elapse between consecutive events, for example, an activity is to be considered suspicious if within a two-minute interval, three consecutive unsuccessful login attempts are made.
- Constructing a database of repetitive attack signatures. An attack may involve neutral activities (mostly in the reconnaissance phase) and/or those misleading the IDS defense devices. In such a case, construction of an attack signature may be impossible or very difficult.

Mistyped commands or answers in automated sessions

Network services and protocols are documented in a precise manner and use determining software tools. Any incompability with known patterns (including typical human errors such as misprints occurring in network packets) may be valuable information to detect services that are possibly being targeted by an intruder.

If the system audit facility uses, for example, send mail relaying, then the relevant log sequence behaves in a regular and predictable manner. However, if the log indicates that a specific process has given illegal commands, it might be a symptom of either a non-malicious event or a spoofing attempt.

The examining of hostile attempts may include:

- Detection of attempts to recover mistyped commands or answers followed by re-launching them,
- Detecting several failed attempts to observe syntax protocols followed by successful ones,
- Detecting adaptive learning attempts to capture errors committed by the same object (service, host). After a certain period, these errors will cease.

Directional inconsistencies in traffic

Any directional inconsistency in packets or sessions is one of the symptoms of a potential attack.
Considering the source address and location (outbound or inbound) can identify the direction of a packet. Session flow is identified by the direction of the first packet of that session. Therefore, a request for service on a local network is an incoming session and a process of activating a Web based service from a local network is an outgoing session.

The following directional inconsistencies may be considered as attack evidence indicators:

- Packets originating in the Internet (incoming) and identified by their local network source address –

request for service incoming from outside, for which the packets have their internal source address. This situation may indicate a possible outside IP spoofing attack. Such problems can be routinely solved at routers that can compare the source address with the destination location. In practice, few routers support this security option since this is the domain of firewalls.

- Packets originating in a local network (outgoing) and sent to an external network with an external destination address – this is a reverse case. An intrusion attempt is accomplished from outside and targeted at an external system.
- Packets with unexpected source or destination ports – if the source port of an incoming or outgoing request is not consistent with the type of service, this may indicate an intrusion attempt (or system scanning). Example: requesting Telnet Service on port 100 in an environment where it is expected that such a service cannot be supported (if available at all). Directional inconsistencies are most likely to be detected by firewalls that simply drop illegal packets. However, firewalls are not always merged with intrusion detection systems, therefore it is expected that the latter will also remedy the above problem.

Unexpected attributes as an intrusion symptom

The most frequent cases are the ones where one is expected to deal with a set of attributes of packets or specific requests for services. It is possible to define the expected attribute pattern. If encountered attributes do not match this pattern, this may indicate a successful intrusion or intrusive attempt.

- Calendar and time attributes – in certain environments, specific network behavior may occur regularly at a certain time of day. If this type of regular behavior gets disrupted, the case needs to be checked. As an example we will use a company where bank transfers are made each Friday afternoon. In this way, electronic data interchange transactions working at this time and on this day can be considered a normal activity. If Friday were a holiday, any transfer occurring would require a check [1].
- System resource attributes. Certain intrusions involve defacing of specific sets of system attributes. A brute force password crack is associated with heavy CPU usage whilst DoS attacks do the same with system services. Heavy usage of system resources (processor, memory, disk drive, system processes, services, and network connections), particularly at an abnormal time, should always be a valuable intrusion indicator.
- Packets with unexpected TCP acknowledgement settings. If there is an ACK-flag set through a packet and no previous SYN-packet has been sent, this fact can be associated with an intrusion attempt (or service scan). Such a situation may also be a symptom of packet damage, a malfunctioning network of software elements, and not an attack attempt.
- Service mix attributes. Often it is possible to define a standard set of inbound and/or outbound services provided to a specific user. For example, if the user is on a business trip, he is expected to use email and file transfer options only. Any attempt, on his account, via Telnet to access various ports may indicate an intrusion.

There is also a more general notion than service mix, namely user and service profiles that help in distinguishing between typical and unexpected attributes. A signature file that holds a common set of services for a specific user may also store additional information composed of multiple attributes. These may include typical system-related working hours of the user, location of the workstation (site in geographic context, IP addresses), intensity of using resources, typical session duration by individual services.

Unexplained problems as intrusion indicators

A potential intruder may design its malicious activity with side effects that will cause odd behavior of the system. Monitoring such side effects is difficult since their location is hardly detectable [1], [5]. Below there are some examples of:

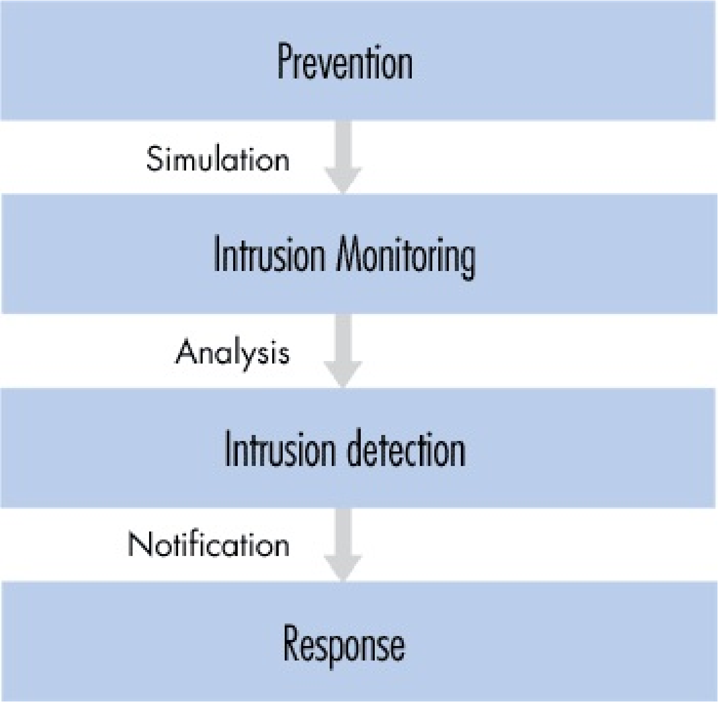
- Unexplained problems with system hardware or software, for example server down, particularly daemons not running, unexplained system restart attempts, changes to system clock settings.

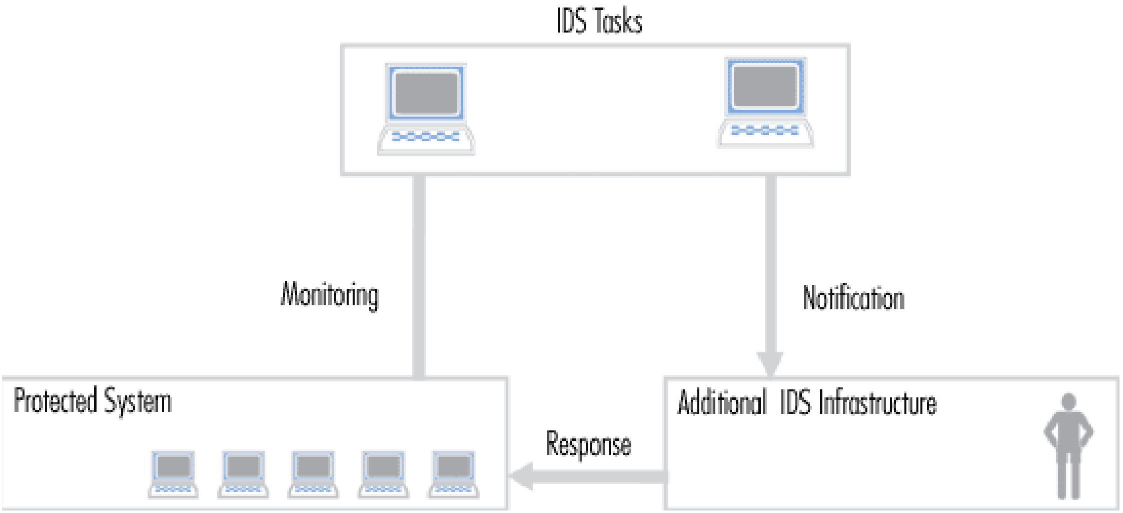
- Unexplained system resource problems: file system overflow; abnormal consumption of CPU usage.
- Odd messages from system daemons, system daemons not running or disturbed (particularly superuser daemons and those designed to monitor the system state, for example Syslog). Such symptoms are always suspicious.
- Unexplained system performance problems (routers or system services, for example long server response times).
- Unexplained user process behavior, for example unexpected access to system resources.
- Unexplained audit log behavior. Audit logs that shrink in size (unless intentionally reduced by the system administrator).

Tasks to be performed

The main task of intrusion detection systems is defense of a computer system by detecting an attack and possibly repelling it. Detecting hostile attacks depends on the number and type of appropriate actions (Fig.1). Intrusion prevention requires a well-selected combination of “baiting and trapping” aimed at both investigations of threats. Diverting the intruder’s attention from protected resources is another task. Both the real system and a possible trap system are constantly monitored. Data generated by intrusion detection systems is carefully examined (this is the main task of each IDS) for detection of possible attacks (intrusions).

(Fig.1) Intrusion detection system activities





(Fig.2) Intrusion detection system infrastructure [3]

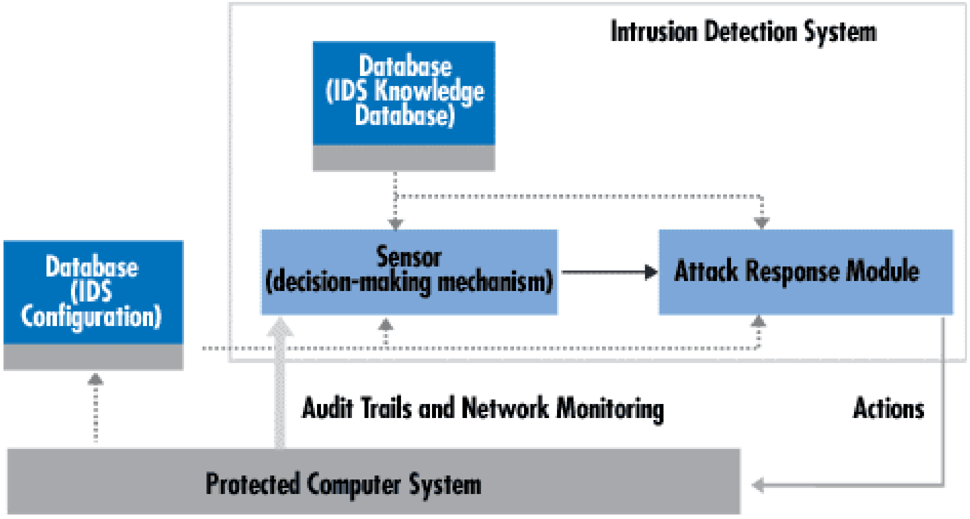
Once an intrusion has been detected, IDS issues alerts notifying administrators of this fact. The next step is undertaken either by the administrators or the IDS itself, by taking advantage of additional countermeasures (specific block functions to terminate sessions, backup systems, routing connections to a system trap, legal infrastructure etc.) – following the organization's security policy (Fig.2). An IDS is an element of the security policy.

Among various IDS tasks, intruder identification is one of the fundamental ones. It can be useful in the forensic research of incidents and installing appropriate patches to enable the detection of future attack attempts targeted on specific persons or resources.

Intrusion detection may sometimes produce false alarms, for example as a result of malfunctioning network interface or sending attack description or signatures via email.

Structure and architecture of intrusion detection systems

An intrusion detection systems always has its core element - a sensor (an analysis engine) that is responsible for detecting intrusions. This sensor contains decision-making mechanisms regarding intrusions. Sensors receive raw data from three major information sources (Fig.3): own IDS knowledge base, syslog and audit trails. The syslog may include, for example, configuration of file system, user authorizations etc. This information creates the basis for a further decision-making process.


Intrusion Detection System
Database (IDS Knowledge Database)
Database (IDS Configuration)
Sensor (decision-making mechanism)
Attack Response Module
Audit Trails and Network Monitoring
Actions
Protected Computer System

(Fig.3) A sample IDS. The arrow width is proportional to the amount of information flowing between system components [6]

The sensor is integrated with the component responsible for data collection (Fig.4) — an event generator. The collection manner is determined by the event generator policy that defines the filtering mode of event notification information. The event generator (operating system, network, application) produces a policy-consistent set of events that may be a log (or audit) of system events, or network packets. This, set along with the policy information can be stored either in the protected system or outside. In certain cases, no data storage is employed for example, when event data streams are transferred directly to the analyzer. This concerns the network packets in particular.

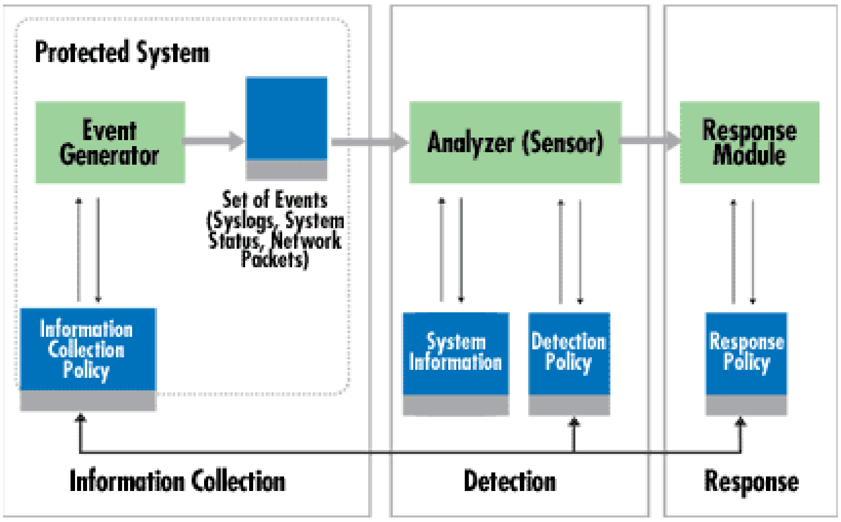
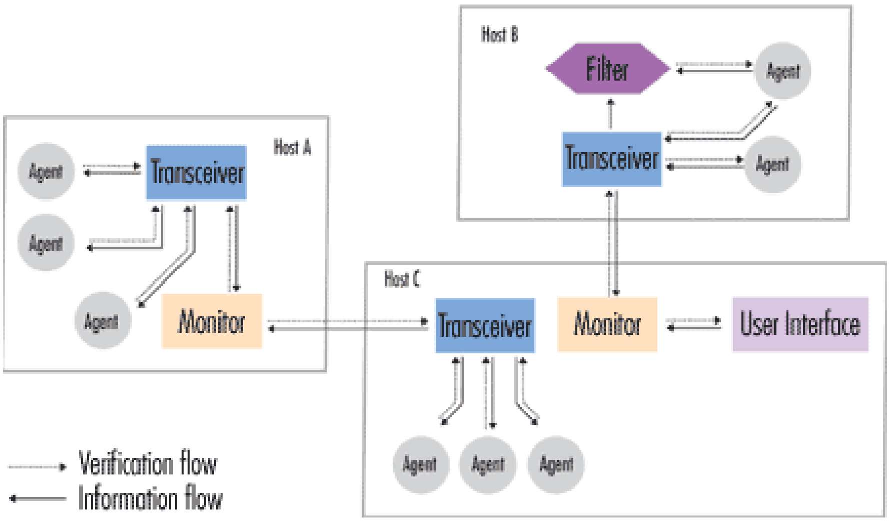


Fig.4 IDS components [7]

The role of the sensor is to filter information and discard any irrelevant data obtained from the event set associated with the protected system, thereby detecting suspicious activities. The analyzer uses the detection policy database for this purpose. The latter comprises the following elements: attack signatures, normal behavior profiles, necessary parameters (for example, thresholds). In addition, the database holds IDS configuration parameters, including modes of communication with the response module. The sensor also has its own database containing the dynamic history of potential complex intrusions (composed from multiple actions).

Intrusion detection systems can be arranged as either centralized (for example, physically integrated within a firewall) or distributed. A distributed IDS consists of multiple Intrusion Detection Systems (IDS) over a large network, all of which communicate with each other. More sophisticated systems follow an agent structure principle where small autonomous modules are organized on a per-host basis across the protected network [8]. The role of the agent is to monitor and filter all activities within the protected area and — depending on the approach adopted — make an initial analysis and even undertake a response action. The cooperative agent network that reports to the central analysis server is one of the most important components of intrusion detection systems. DIDS can employ more sophisticated analysis tools, particularly connected with the detection of distributed attacks [9]. Another separate role of the agent is associated with its mobility and roaming across multiple physical locations. In addition, agents can be specifically devoted to detect certain known attack signatures. This is a decisive factor when introducing protection means associated with new types of attacks [10]. IDS agent-based solutions also use less sophisticated mechanisms for response policy updating [11].

One multi-agent architecture solution, which originated in 1994, is AAFID (Autonomous Agents for Intrusion Detection) — see Fig.5. It uses agents that monitor a certain aspect of the behavior of the system they reside on at the time. For example, an agent can see an abnormal number of telnet sessions within the system it monitors. An agent has the capacity to issue an alert when detecting a suspicious event. Agents can be cloned and shifted onto other systems (autonomy feature). Apart from agents, the system may have transceivers to monitor all operations effected by agents of a specific host. Transceivers always send the results of their operations to a unique single monitor. Monitors receive information from a specific network area (not only from a single host), which means that they can correlate distributed information. Additionally, some filters may be introduced for data selection and aggregation [12], [10].



(Fig.5) An AAFID compliant representation of an intrusion detection system employing autonomous agents [12]

If you would like us to email you when one of our authors releases another article on WindowSecurity.com, subscribe to our 'Real-Time Article Update' by clicking here. Please note that we do NOT sell or rent the email addresses belonging to our subscribers; we respect your privacy!

References

[1] E. Amoroso, Wykrywanie intruzów, Wydawnictwo RM, Warszawa 1999 (in Polish).
[2] B. Mukherjee, T.L. Heberlein, K.N. Levitt, Network intrusion detection, IEEE Network 8 (3), 1994, pages 26-41.
[3] P. Dorosz, P. Kazienko, Systemy wykrywania intruzów, VI Krajowa Konferencja Zastosowan Kryptografii ENIGMA 2002, Warsaw 14-17 05.2002. , p. TIV 47-78, http://www.enigma.com.pl/konferencje/vi_kkzk/index.htm. (in Polish)
[4] G. Mansfield, K. Ohta, Y. Takei, N. Kato, Y. Nemoto, Towards trapping wily intruders in the large, Computer Networks 34, 2000, pages 659-670.
[5] C. Stoll, Kukulcze jajo, Rebis, Poznan 1998. (in Polish)
[6] H. Debar, M. Dacier, A. Wespi, Towards a taxonomy of intrusion-detection systems, Computer Networks 31, 1999, pages 805-822.
[7] E. Lundin, E. Jonsson, Survey of research in the intrusion detection area, Technical report 02-04, Department of Computer Engineering, Chalmers University of Technology, Göteborg January 2002, http://www.ce.chalmers.se/staff/emilie/papers/Lundin_survey02.pdf.
[8] C. Krügel, T. Toth, Applying Mobile Agent Technology to Intrusion Detection, ICSE Workshop on Software Engineering and Mobility, Toronto May 2001, http://www.elet.polimi.it/Users/DEI/Sections/Compeng/GianPietro.Picco/ICSE01mobility/papers/krugel.pdf.
[9] C. Krügel, T. Toth, Distributed Pattern Detection for Intrusion Detection, Conference Proceedings of the Network and Distributed System Security Symposium NDSS '02, 2002, http://www.isoc.org/isoc/conferences/ndss/02/proceedings/papers/kruege.ps.
[10] J.S. Balasubramaniyan, J.O. Garcia-Fernandez, D. Isaco, E. Spafford, D. Zamboni, An Architecture for Intrusion Detection using Autonomous Agents, 14th IEEE Computer Security Applications Conference ACSAC ’98, December 1998, pages 13-24, http://www.cs.umbc.edu/cadip/docs/NetworkIntrusion/tr9805.ps.
[11] D.J. Ragsdale, C.A. Carver, J.W. Humphries, U.W. Pooh, Adaptation techniques for intrusion detection and intrusion response systems, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2000, pages 2344-2349, http://www.itoc.usma.edu/ragsdale/pubs/adapt.pdf.
[12] E.H. Spafford, D. Zamboni, Intrusion detection using autonomous agents, Computer Networks 34, 2000, pages 547-570.

{1} Today, many firewalls are provided with intrusion detection system tools as additional software packets. However, an IDS is not a firewall that sits between the internal and internal organization’s networks to separate them for security reasons.