

# Web-style Wireless IDS attacks

***by Sergey Gordeychik***

gordey @ ptsecurity.com

***Table of Contents***

<b>Introduction</b>	<b>3</b>
<b>WIDS architecture</b>	<b>4</b>
<b>Sources of threats</b>	<b>4</b>
<b>Hacking through air gaps</b>	<b>5</b>
<b>Intrusions on a local network</b>	<b>7</b>
<b>Operator intrusions</b>	<b>9</b>
<b>Conclusion</b>	<b>10</b>
<b>About the author</b>	<b>11</b>
<b>About Positive Technologies</b>	<b>11</b>
<b>References</b>	<b>12</b>

## ***Introduction***

Wireless intrusion detection systems (WIDS) are not yet as popular as their wired counterparts, but current trends would suggest that their number is set to grow. One positive factor in this respect is the integration of such programs with active network equipment and Management awareness of the risks associated with the unauthorised use of wireless devices. This awareness has led to an increase in the number of WIDS installations - even where wireless networks are not used.

In view of this situation, specialists in the field of security are now aware of the need to evaluate not only the quality features of any product, but also of the need to predict any possible negative influence arising from its implementation on the security of a corporate network.

This article looks at the results of research into wireless intrusion detection systems from the point of view of the specialist in the field of applications security. Design faults discovered are not discussed in the article as their correction requires significant effort on the part of the manufacturer.

## ***WIDS architecture***

A modern system of detecting wireless intrusion is a fairly complicated solution based on two- or three-tier architecture - often based on Web technologies.

WIDS architecture is based on sensors which collect, and sometimes process, wireless traffic as part of the monitoring process. Sensors can be based on standard operating systems or "specialised software and hardware platforms" (in most cases Linux). As a rule, sensors are quite intelligent devices which support TCP/IP and have sophisticated control interfaces.

The Sensors interact with the data collection component (server), and transfer to it information on detected intrusions or intercepted packets. The server processes information received, and performs the functions of detecting intrusions and correlating security-related events. A standard DBMS (database management system) is normally used to store information. To manage the system and monitor events, a control console is used in the form of a "fat" or "thin" client.

Thus, WIDS is a distributed system which is potentially vulnerable to intrusions not only in a wireless area.

## ***Sources of threats***

It is possible to give the article a more formally scientific tone with the formation of an "intruder model", i.e. to define basic anthropogenic sources of threats. For WIDS these include external intruders interacting with the system through radio ether, internal intruders who have access to a local network, and operators who have certain limited opportunities to manipulate system components.

The architecture of the typical wireless intrusion detection system is discussed in the article and the vectors of the intrusions are shown in the figure 1.

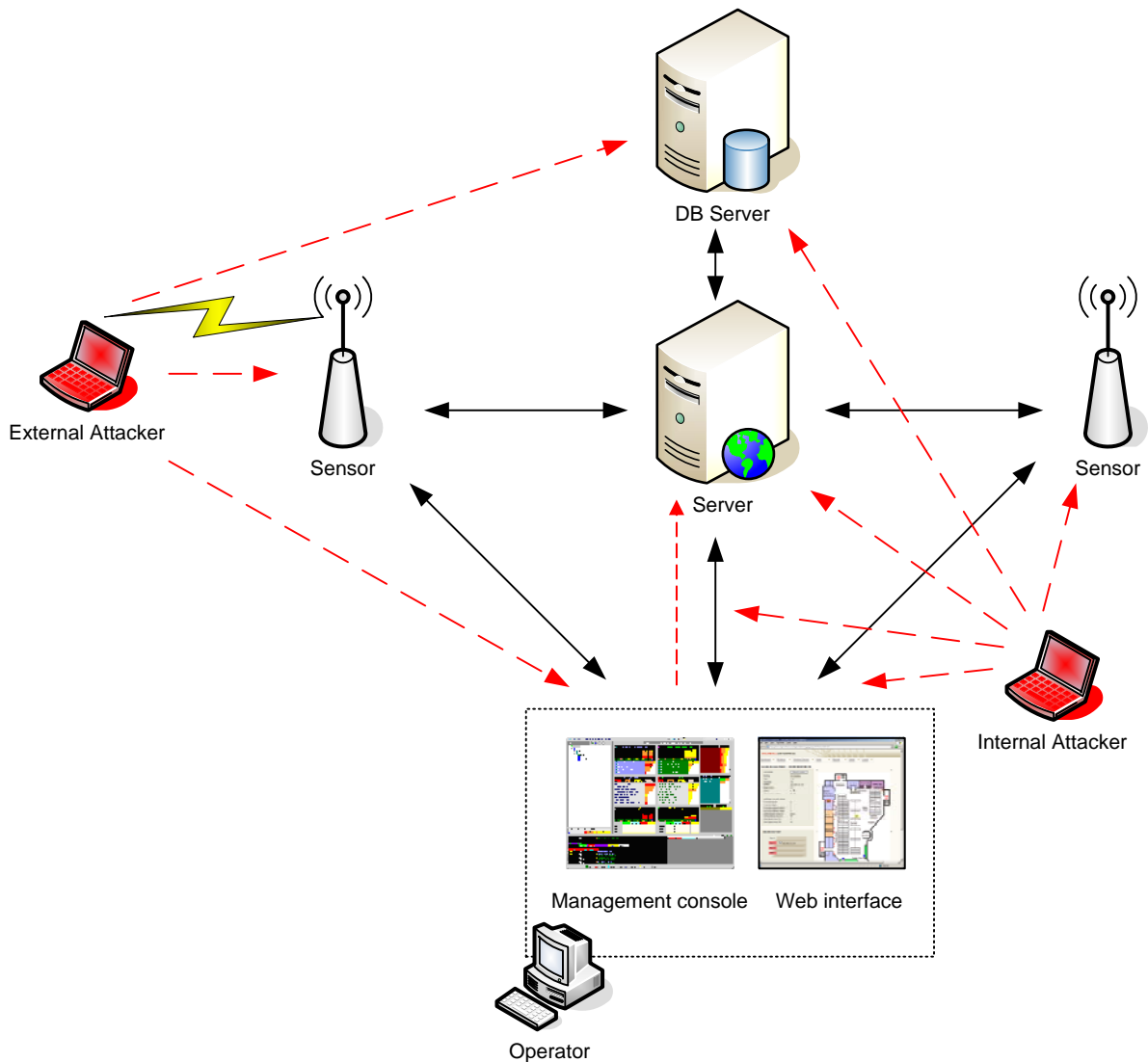


Figure 1. WIDS architecture and therats

### ***Hacking through air gaps***

The principal mechanisms by which external intruders impacts on the wireless intrusion detection system are based on the creation of 802.11 frames, processing of which leads to non-standard situations. The experience of wired intrusion detection systems [1], and also the packet sniffers Ethereal/Wireshark, shows that the presence of vulnerabilities in

"vivisectors" of complex network protocols is entirely normal. The state machine of 802.11 link layer is fairly complicated, so as to confuse the developers. Vulnerabilities in Kismet [2], coupled with recent publications [3] on vulnerabilities in drivers of wireless clients, are compelling people to consider the probable presence of such problems in WIDS sensors. However, this has only a weak connection with the theme of the article.

The data received from a non-trusted source is saved in a database and there is the probability of it not being processed correctly. And as a result there is a possibility of an intruder carrying out SQL Injection-type attacks. By adding to the packets fields of special symbols it is possible to terminate the initial SQL query and add SQL operators to it. In practice, such an intrusion can be carried out by creating fake access points or peer-to-peer networks with SSID like:

```
‘;insert into ...
```

A fundamental, but surmountable restriction to the use of this type of vulnerability is the SSID length (32 bytes).

At present, such vulnerability is processed as part of the policy of responsible disclosure and will possibly be published at a later date. However, the reader can verify the WIDS response to the crafted wireless networks using DBMS tracking tools (SQL Profiler or similar), for example:

```
iwconfig ath0 mode master essid ‘;--
```

A further widely-accepted Web vulnerability characteristic of wireless intrusion detection systems is Cross-Site Scripting. Information on a detected intrusion appears in the control console, often Web browser-based. Accordingly, an intruder can select as the SSID of the fake access point a magical sequence of symbols:

```
"><script>alert()</script>
```

And it launches a script in the browser of the operator or administrator which can be controlled by the intruder. In such a case, 32 bytes is adequate to specify external server as a source of the script. The results of such intrusion may be many and varied - from the theft of authentication data through to carrying out certain actions related to customising of the WIDS setup - in place of the operator. Such vulnerabilities was detected in the Web interface of the Airmagnet Enterprise server [4]. The conditions of the

stored XSS arose when the wireless networks SSID appeared in the access control lists Enterprise Server

<https://<servername>/Amom/Amom.dll/BD>

Where a "fat" client is used the situation can be complicated. For example, the AirMagnet control console for displaying information about an intrusion uses an embedded Internet Explorer object and inserts in the HTML template SSID of access points (or the client) without screening. If the browser works in the security zone Local Machine, the insertion of scripts may lead to serious consequences. Further details about risks associated with use in applications of the object Internet Explorer working in the security zone My Computer can be found in [5] and [6].

In practice, all tested solutions are vulnerable to Cross-site Request Forgery (CSRF) attacks. However, this vulnerability is so widespread that it was not even considered worth mentioning.

Of course, implementation of these intrusions requires that the intruder has information about the type of WIDS used, but this issue is fairly well described in the publication [7].

### ***Intrusions on a local network***

An internal user has far more opportunities than his external intruder counterpart. Since WIDS control interfaces for sensors and servers are fully functional Web interfaces, it is highly probable that an intruder will be able to find in these applications the entire range of vulnerabilities from Web Application Consortium Threads Classification [8].

Examples include vulnerabilities [9], in Cisco WLSE and so on. In the sensor control interface AirMagnet SmartEdge Sensor a persistent Cross-Site Scripting vulnerability was detected in audit journals reviewing interface:

<https://<sensorip>/AirMagnetSensor/AMSensor.dll/XH>

**WebServer Log**

In order to carry out an intrusion in this case, the name of the user entered at the time of authentication is used. The non-persistent variant XSS is present in 404-error pages:

[http://<sensor IP>/xss<script>alert\(\)</script>](http://<sensor IP>/xss<script>alert()</script>)

[https://<sensor IP>/xss<script>alert\(\)</script>](https://<sensor IP>/xss<script>alert()</script>)

One further vector of intrusions which an internal intruder can use is network interaction between system components, such as collection of data from sensors, saving events in a DBMS, remote-control and browsing events. Naturally, this traffic is sufficiently critical for vendors to deal with its protection using such reliable protocols as SSL.

However, concern about convenience of users is compelling manufacturers to use self-signed certificates rather than use a proper PKI-style verification process. For example, the control console Airmagnet accepts practically any certificate in the server response. This allows an intruder who has satisfied the "man in the middle" conditions to decipher traffic (including user passwords) transmitted between the control console and the server by using generally accessible tools such as ettercap or Cain [10]. Below is an example of traffic intercepted and deciphered.

[Client-side-data]

GET /AMom/AMom.dll/UA HTTP/1.1

Accept: \*/\*

AMUser: admin <STATIONID>

AMBuild: 4694

User-Agent: AirMagnet

Host: <serverip>

Connection: Keep-Alive

Authorization: **Basic YWRtaW46MTEExMTEEx**

[Server-side-data]

HTTP/1.1 200 OK

Date: Mon, 20 Mar 2006 12:53:12 GMT

Server: Apache/2.0.52 (Win32) mod\_ssl/2.0.52 OpenSSL/0.9.7a

Content-Length: 301

Keep-Alive: timeout=15

Connection: Keep-Alive

Content-Type: text/html

[Server-side-data]

<html>

3	2	AirMagnetSensor	111111	16777215	1	0
---	---	-----------------	--------	----------	---	---



## **Conclusion**

I would like to conclude by giving several minor recommendations for specialists selecting or configuring a wireless intrusion detection system.

1. Check the system response to non-standard traffic in the wireless network. Several examples of such traffic were given in the article. In addition different fuzzers, for example [11] can be used.
2. Pay attention to the level of privileges used by the WIDS to work with the DBMS. The consequences of intrusions could be very serious if a superuser account is used.
3. When planning a network infrastructure for the WIDS, be aware of the requirements for separation of networks. Transfer control traffic to a separate segment/VLAN.
4. Switch off unused control protocols on remote sensor. The use of telnet in 2006 can only be justified by constructing a honeypot.
5. Scan the network interfaces of the WIDS sensors and servers using a vulnerabilities scanner which supports Web applications. I guarantee that, in most cases, you will get a nasty shock. It is important that you make a backup copy of the system. A scanner may inadvertently obtain access to a remote controls and cause mayhem by pressing all available buttons.
6. Pay serious attention to the management workstation. The author uses the following approach, which is easily realised by using a proxy server:
  - the browser used for working in the corporate network does not have access to Internet resources.
  - the browser working with the Internet is restricted to use of corporate resources, and works in a "sandbox".
7. It is also possible to block the execution of scripts in the security zone My Computer [12] or to use Terminal Server for keeping client applications separate.
8. Try to use the WIDS system as a critical business application and fulfil the requirements formulated in the security policy for the given class of product. In addition to a special review of the policy, it is a unique opportunity to be with the IT specialists and users who carry out the requirements of the policy every day.

***About the author***

Sergey Gordeychik is the System Architects of Positive Technologies ([www.ptsecurity.com](http://www.ptsecurity.com)), where he is responsible for application, wireless and mobile security. Mr. Gordeychik is an author of "Wireless Security", "Auditing Web-applications security" and "Securing Microsoft Windows-based Enterprise" training courses in Security Training Centre Informzaschita ([www.itsecurity.ru](http://www.itsecurity.ru)). He is regular author of "Windows IT Pro/RE" magazine, SecurityLab ([www.securityfocus.ru](http://www.securityfocus.ru)) and other. Mr. Gordeychik is also contributor of Web Application Security Consortium (WASC).

***About Positive Technologies***

Positive Technologies is a private company specializing in network information security. Its head office is located in Moscow, Russia.

The company has two main concentrations: provision of integrated services used in protecting computer networks from unauthorized access; and development of the MaxPatrol security scanner and its complementary products. The company's Russian and Ukrainian customers include largest banks, state organizations, leading telecommunication and industrial companies.

Our two concentrations both complement and enrich each other. The enormous practical experience of the leading Russian security specialists employed by the company allows us to create products of the highest quality. An excellent product can provide effective, successful, and quick resolutions of information-security problems.

Besides this, the company owns a leading Russian Internet portal [www.securityfocus.ru](http://www.securityfocus.ru) for information security that it uses for analytic and educational purposes.

## **References**

[1] Vulnerabilities in Snort 2.4

<http://www.security.nnov.ru/soft/6810.html?l=EN>

[2] Kevin Finisterre, «New Kismet Packages available - SayText() and suid kismet\_server issues»

<http://www.security.nnov.ru/docs3012.html>

[3] Johnny «Cache», David Maynor «Device Drivers: Dont build a house on a shaky foundation»

[www.blackhat.com/presentations/bh-usa-06/BH-US-06-Cache.pdf](http://www.blackhat.com/presentations/bh-usa-06/BH-US-06-Cache.pdf)

[4] AirMagnet Enterprise

<http://www.airmagnet.com/products/enterprise.htm>

[5] «SPI Dynamics WebInspect Cross Application Script Injection Vulnerability»

<http://www.securityfocus.com/bid/14385/references>

[6] SPI Dynamics, «Feed Injection in Web 2.0»

<http://www.spidynamics.com/assets/documents/HackingFeeds.pdf>

[7] Joshua Wright, «Weaknesses in Wireless LAN Session Containment»

[http://i.cmpnet.com/nc/1612/graphics/SessionContainment\\_file.pdf](http://i.cmpnet.com/nc/1612/graphics/SessionContainment_file.pdf)

[8] Web Application Security Consortium, Threats Classification

<http://www.webappsec.org/projects/threat/>

[9] Cisco Security Advisory: Multiple Vulnerabilities in the WLSE Appliance

<http://www.cisco.com/warp/public/707/cisco-sa-20060419-wlse.shtml>

[10] Cain & Abel

<http://www.oxid.it/cain.html>

[11] Raw Wireless Tools Homepage

<http://rfakeap.tuxfamily.org/>

[12] How to strengthen the security settings for the Local Machine zone in Internet Explorer

<http://support.microsoft.com/kb/833633>