

# Система DLP

Черновик

## 1. Введение

Документ описывает общую концепцию DLP-систем, приблизительные принципы их работы и способы взаимодействия с пользователями, администраторами и сотрудниками отдела информационной безопасности.

Система *DLP* ([англ. Data Leak Prevention](#)) - специальная информационная система, созданная для предотвращения утечек конфиденциальной информации вовне. DLP-системы строятся на принципах анализа потоков данных, пересекающих периметр защищаемой информационной системы. При детектировании в этом потоке конфиденциальной информации срабатывает активная компонента системы, и передача сообщения блокируется.

Распознавание конфиденциальной информации в *DLP*-системах может производиться различными способами: анализом формальных признаков посредством строгих алгоритмов (регулярных выражений, метрик, шаблонов, т.п.) и применением нечеткой логики (методов машинного обучения, семантического анализа текстов). Первый способ позволяет избежать ложных срабатываний, но зато требует предварительной классификации документов, внедрения меток, сбора сигнатур и т.д. Второй способ предполагает ложные срабатывания, но потенциально способен обнаруживать трудноформализуемые угрозы.

### 1.1. Цель

*Указать цель создания документа.*

### 1.2. Область действия

*Краткое описание области действия документа: с какими проектами и документами он связан, все остальное, что может повлиять на документ и на что он сам влияет.*

### 1.3. Ссылки

*Полный список всех документов, на которые есть ссылки в других частях данного документа.*

### 1.4. Обзор

*Описание того, что содержится в остальной части документа и объяснение, как документ организован.*

## 2. Определения

*Перечень определений и сокращений, присутствующих в тексте. Могут быть представлены в любом удобном порядке.*

## 3. Концепция DLP

Пример срабатывания DLP от Symantec на файле Excel. Видно, как в превью файла цветом выделены фрагменты текста, связанные со сработавшими правилами проверки.

Firefox window displaying the Symantec Data Loss Prevention (DLP) interface. The browser address bar shows the URL: `https://localhost/ProtectManager/EndpointIncidentDetail.do?value(variable_1)=incident.id&value(operator_1)=in&value(operand_1)=3843&value(state_menuID)=sa`.

The interface displays details for Incident 00003843, titled "Endpoint Copy to Network Share". The status is "New" and the severity is "High".

**Policy Matches:**

Policy	Matches
Credit Card Tuning - AMEX [ view policy ]	10
AMEX (Data Identifiers)	10

**Incident Details:**

Server	Primary Detection
Occurred On	12/4/13 1:16 PM
Reported On	12/4/13 1:16 PM
Is Archived	No [ Do Not Archive ]
User	ACME\djackson
Machine Name	EPOINT-WIN7X86
Machine IP (Corporate)	192.168.0.13

**Matches (matches found in 1 component):**

C:\Users\djackson\Desktop\Customer credit card info.xls (10 Matches):

- ...ID FIRST LAST AMEX: 344058488426266
- EXP DATE PHONE ZIP 30000 NATALIE WALDMAN AMEX: 342955624318368
- Apr-10 (592) 427-8964 89427-8964 30001...
- 49627-1525 30026 Beth MCDERMOTT AMEX: 372135898797783 Jul-12 (962) 282-7475
- 14282-7475 30032...74 90478-9374 30041
- LILA AUDETTE AMEX: 347279493269015 Jul-12 (318) 807-4810 16807-4810
- 30042...25952-8251 30045 DAREN SCHIAVONE AMEX: 345796298727014 Jul-12 (674) 902-3953 05902-3953 30046...
- 43213-7677 30049 YOLANDA TAYLOR AMEX: 342650299263839 Jul-12 (827) 426-6088
- 48426-6088 30050...73187-4819 30053 CYNTHIA MORRELL AMEX: 342781835011463 Nov-09 (644) 870-9142
- 41870-9142 30054... 70181-1642 30057 ETHEL FIGUEROA AMEX: 342337649030528 Nov-09 (337) 288-6963 82288-6963
- 30063...15747-2901 30073 APRIL DICKERSON AMEX: 348771682068975 Dec-09 (525) 722-0402 60722-0402 30080... 73187-4819
- 30089 RUTH MCDERMOTT AMEX:

**Attributes:**

**Data Insight:**

- Most Active
- Data User 2
- Data User 3

**LDAP:**

Employee ID	012345
Whole Name	David Jackson
Telephone Number	415-829-5071
Employee Email	djackson@acme.com
Manager Name	Cindy Delarosa
Manager Email	cdelarosa@acme.com

The bottom status bar shows the URL: `https://localhost/ProtectManager/EndpointIncidentDetail.do?value(variable_1)=incident.id&value(operator_1)=in&value(operand_1)=3843&value(state_menuID)=saved.7101#`.

А это пример превью скана бумажного документа или PDF-документа. Красными рамками выделены распознанные фрагменты с приватными данными. Очевидно, DLP-системы содержат OCR-модуль для таких целей.

Reference SSN:

333-22-4567

## Contact Details on File

Email	jane.smith@example.org
Phone	858-333-1111

## Auto Loan Details

Make	Honda Accord	Model	LX	Year	2014
Rate	2.9%	Term	60 mo		

## Loan Terms

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Orci porta non pulvinar neque laoreet. Vitae congue mauris rhoncus aenean vel elit scelerisque mauris pellentesque. Amet nisl suscipit adipiscing bibendum est. Mattis rhoncus urna neque viverra justo nec ultrices dui sapien. Sed ullamcorper morbi tincidunt ornare massa eget egestas purus viverra. Arcu felis bibendum ut tristique et egestas. Viverra aliquet eget sit amet tellus. Eu mi bibendum neque egestas congue quisque

## 4. API

### 4.1. Контракт

Ниже представлено приблизительное представление о том, как должен выглядеть контракт адаптера DLP-системы. Сорян, это псевдокод, но вроде и так понятно.

Прикладного программиста прежде всего интересует API для анализа содержимого (простого текста, документов MS Office или PDF). Интерфейс `IDataLossPreventor` позволяет проанализировать переданное содержимое и получить в ответ коллекцию правил, которые сработали на этом содержимом.

```

/// <summary>
/// .
/// </summary>
public interface IDataLossPreventor
{
    /// <summary>
    /// .
    /// , .
    /// </summary>
    /// <remarks>
    /// , DLP- .
    /// </remarks>
    /// <param name="applicationID"> , DLP-.</param>
    /// <param name="content"> .</param>
    /// <returns> .</returns>
    IEnumerable<Policy> Inspect(string applicationID, Content content);
}

```

Метод Inspect интерфейса IDataLossPreventor возвращает коллекцию экземпляров класса Policy:

```

/// <summary>
/// .
/// </summary>
public sealed class Policy
{
    /// <summary>
    /// , DLP-.
    /// </summary>
    public string Name { get; set; }

    /// <summary>
    /// DLP-.
    /// </summary>
    public string Message { get; set; }

    /// <summary>
    /// , .
    /// </summary>
    public Similarity Similarity { get; set; }

    /// <summary>
    /// , .
    /// </summary>
    public Similarity Tolerance { get; set; }

    /// <summary>
    /// , , .
    /// </summary>
    public string ContentID { get; set; }

    /// <summary>
    /// , .
    /// </summary>
    public IEnumerable<Location> Locations { get; set; }
}

```

На при вызове метода Inspect передается экземпляр класса Content, представляющий содержимое анализируемого документа, а так же (желательно) тип этого содержимого.

```

    /// <summary>
    /// .
    /// </summary>
    public sealed class Content
    {
        /// <summary>
        /// .
        /// </summary>
        public string ContentID { get; set; }

        /// <summary>
        /// .
        /// </summary>
        public Stream Payload { get; set; }

        /// <summary>
        /// .
        /// DLP-.
        /// </summary>
        public ContentType Type { get; set; }
    }

```

Типичные модульные тесты на этот интерфейс могут выглядеть следующим образом:

```

[TestFixture]
public sealed class DataLossPreventorTests
{
    private readonly IDataLossPreventor sut;

    ...

    [Test]
    public void When_Content_is_safe_Then_Rules_will_not_be_returned()
    {
        // Arrange
        var safeContent = new Content("Absolutely safe text", ContentType.Text);

        // Act
        var ruleSet = sut.Inspect("appID", safeContent );

        // Assert
        ruleSet.Should().BeEmpty();
    }

    [Test]
    public void When_content_contains_unsafe_data_a_collection_of_Rules_that_were_triggered_is_returned()
    {
        // Arrange
        var unsafeContent = new Content("Text with secret data such as #123-45-67", ContentType.Text);

        var expected = new Policy[]
        {
            new Policy("WORK-PHONE", " ", Similarity.Likely, content.ContentID)
                .AddLocation(new TextLocation(left: 30, right: 40))
        };

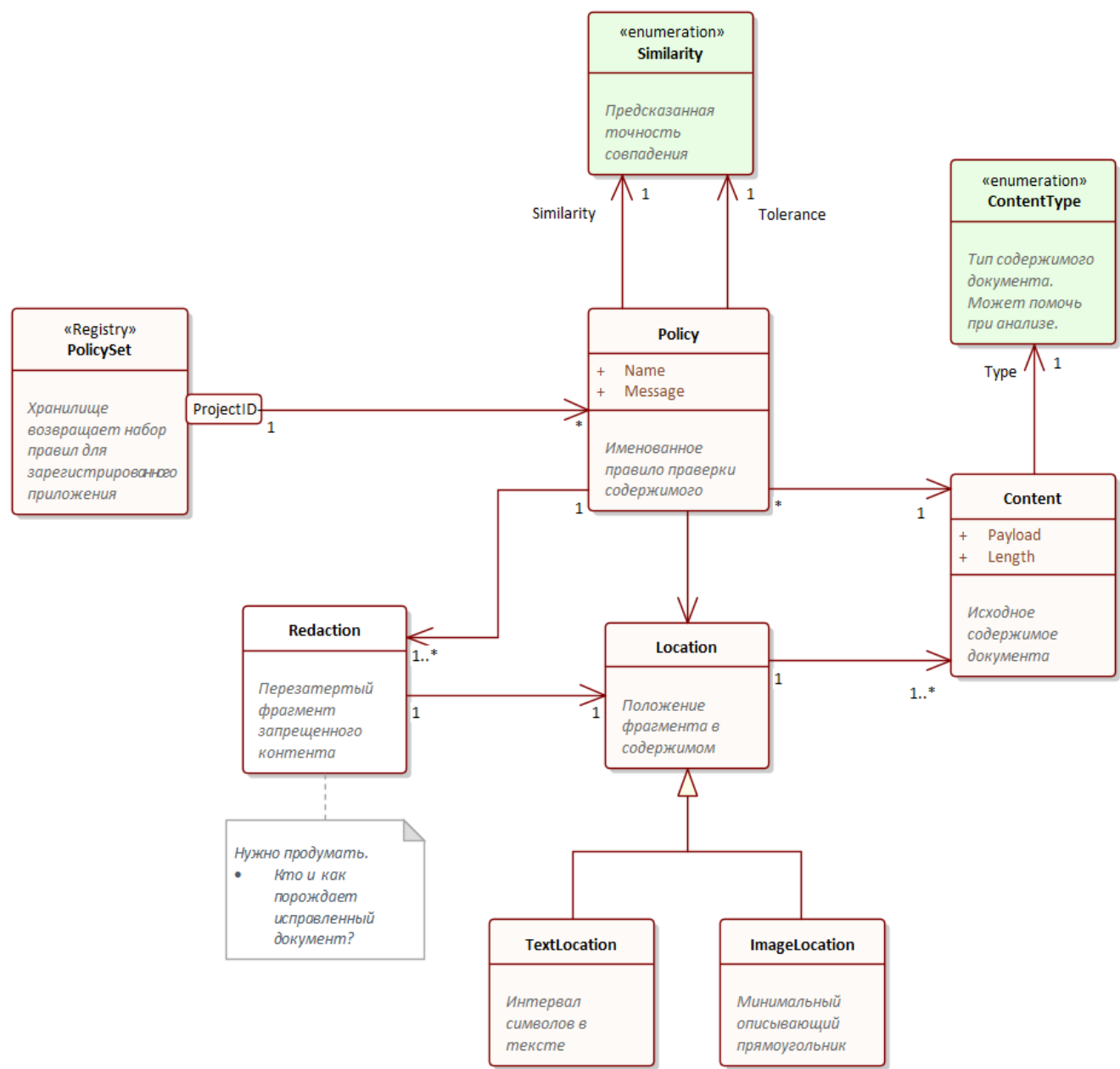
        // Act
        var actual = sut.Inspect("appID", unsafeContent);

        // Assert
        actual.Should().BeEquivalentTo(expected);
    }
}

```

Очевидно, что второй тестовый метод разумнее сделать параметрическим.

4.2. Аналитическая модель ответа DLP-системы.



5. Предполагаемая реализация

Имеет смысл построить компонент DataLossPreventer как цепочку декораторов над адаптером реальной DLP-системы SearchInform (или другой, если эта будет заменена ГПН).

В декораторах удобно будет реализовать функционал обработки ошибок (а точнее - сокрытием ошибок реальной DLP-системы), восстановления после сбоев (паттерн Retry), регулировку нагрузки (то есть реализацию паттерна CircuitBreaker) и всего, что требуется для безопасной работы с внешними ресурсами.

