Machine Learning for Intrusion Detection

Prof. Dr. Klaus-Robert Müller,
Pavel Laskov, Ph.D. and Konrad Rieck

Vorlesung "Maschinelles Lernen – Theorie und Anwendung"
Technische Universität Berlin

May 6, 2008

Outline

Zero Day Hole In Google Desktop

Posted by Zonk on Fri Jun 01, 2007 04:47 PM

40by40 writes

> "A Web application security specialist has figured out a way to launch man-in-the-middle attacks against a computer with a fully patched Google Desktop installed. With knowledge of the Google Desktop security model (a combination of one-time tokens, iFrames and JavaScript), hacker

Hackers target home users for cash

'It's war out there' says Symantec report

Consumers are now on the main target of malicious hackers intent on enriching themselves through the misery of others. Vulnerabilities in desktop applications and the increased use of stealth techniques are on the rise among members of the digital underground, according to the latest edition of Symantec's Internet Security Threat Report.

Meldung vom 15.08.2003 16:55

heise Security

BSI richtet Fax-Abruf für Opfer von W32.Blaster ein

vorlesen

Opfer des Wurms *Lovsan* beziehungsweise *W32.Blaster* können sich jetzt auch per Fax Ratschläge für Erste-Hilfe-Maßnahmen holen. Wie das Bundesamt für Sicherheit in der Informationstechnik (BSI) in Bonn mitteilt, wurde dazu ein

tagesschau.de

Hacker knacken Millionen Kreditkarten-Konten

Der Skandal um gestohlene Kreditkarten-Daten in den USA weitet sich aus. Nach jüngsten Angaben stahlen Hacker vermutlich die Daten von mehr als acht Millionen Kreditkarten. Zunächst war von fünf Millionen Kreditkarten der Anbieter Visa und MasterCard die Rede gewesen. Medienberichten zufolge sind auch Kreditkarten von American Express betroffen.

The Register®

FBI logs its millionth zombie address

Federal law enforcement agents targeting botnets recently recorded a grim milestone, identifying the millionth potential zombie victim, the FBI said Wednesday.

BBC NEWS

Estonia hit by 'Moscow cyber war'

Estonia says the country's websites have been under heavy attack for the past three weeks, blaming Russia for playing a part in the cyber warfare.

Estonia says many state websites hav been affected

Many of the attacks have come from Russia and are being hosted by Russian state computer servers, Tallinn says.

Security – Who cares?

The Internet as a risk factor

- Omnipresence of security threats and attacks
- Severe economic damage due to Internet crime
- Emergence of new criminal "industries"

For example: A careless user may fall victim to ...

- Credit card, password and identity theft
- Remote control of his system, e.g. for sending Spam
- Involvement in crime as a "stepping stone"

Computer Security

Protection of resources on computer systems

Principle goals of security

1. *Confidentiality* of resources
2. *Integrity* of resources
3. *Availability* of resources

For example: You write a love-letter to your friend

- Only the recipient should read the letter → Confidentiality
- Your message should not be tampered → Integrity
- The target mailbox should not be blocked → Availibility

Security Measures

Active security: Prevention and protection

- Encryption of communication (Confidentiality)
- User and message authentication (Integrity)
- Redundancy and distribution of data (Availibility)

Reactive Security: Detection and response

- Anti-virus scanners and malware removal tools
- Intrusion detection and response systems
- Incident management and computer forensics

Intrusion Detection Systems (IDS)

Attack

Attempt to comprise the confidentiality, integrity or availibility

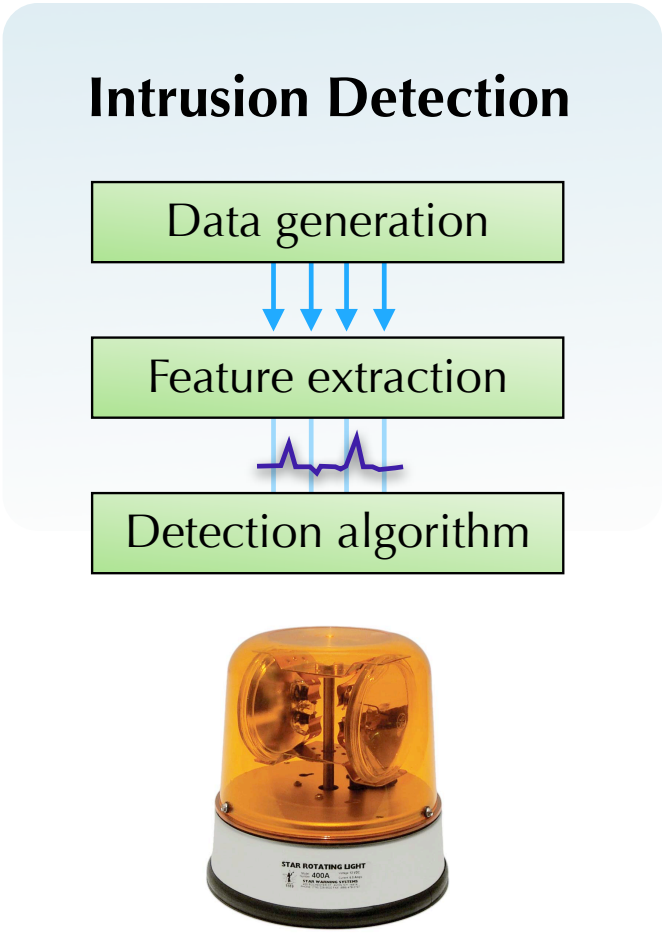
Intrusion detection system (IDS)

System monitoring a stream of events for attacks

Differentiation of IDS by ...

- Event source (e.g. host, network, application)
- Analysis type (e.g. rules, heuristics, learning)
- Response (e.g. blocking, sandboxing)

Intrusion Detection in Detail



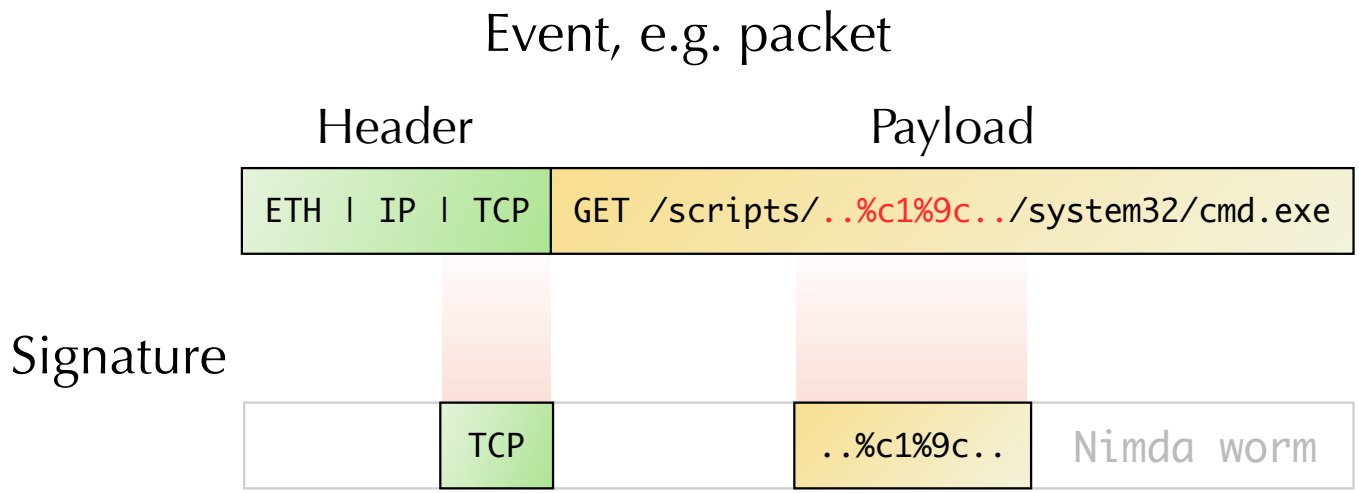
1. Data generation
   Monitoring a stream of events, e.g. network packets
2. Feature extraction
   Extraction of features from events, e.g. strings of network packets
3. Detection algorithm
   Classification or anomaly detection on extracted features

Classic Intrusion Detection

Identification of attacks using signatures (detection rules)

Life-cycle of an attack signature

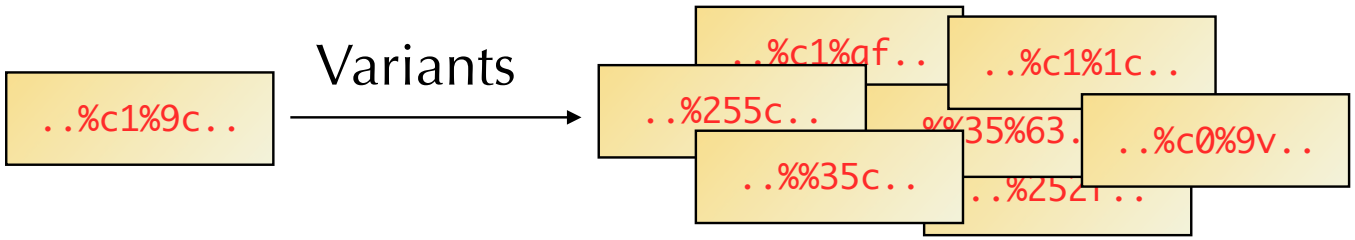
- Security expert analysis new attack and develops signature
- Intrusion detection systems are updated with new signature
- Signature identifies attack in the wild



Drawbacks of Signatures

Classic intrusion detection may fail in the future:

- Human analyses expensive and time-consuming
- Further delay to distribution of new signatures
- Unable to scale with increasing amount of attacks
- Ineffective against attack variants and polymorphism



→ **Need for automatic and adaptive detection technology**

Machine Learning for IDS

Extend intrusion detection with machine learning

1. Extraction of features suitable for learning
2. Application of robust learning methods

⇒ High cost of labels (up to several hours per incident)

- Capability to learn on unlabeled or one-class data
- Compensation of attacks in training data

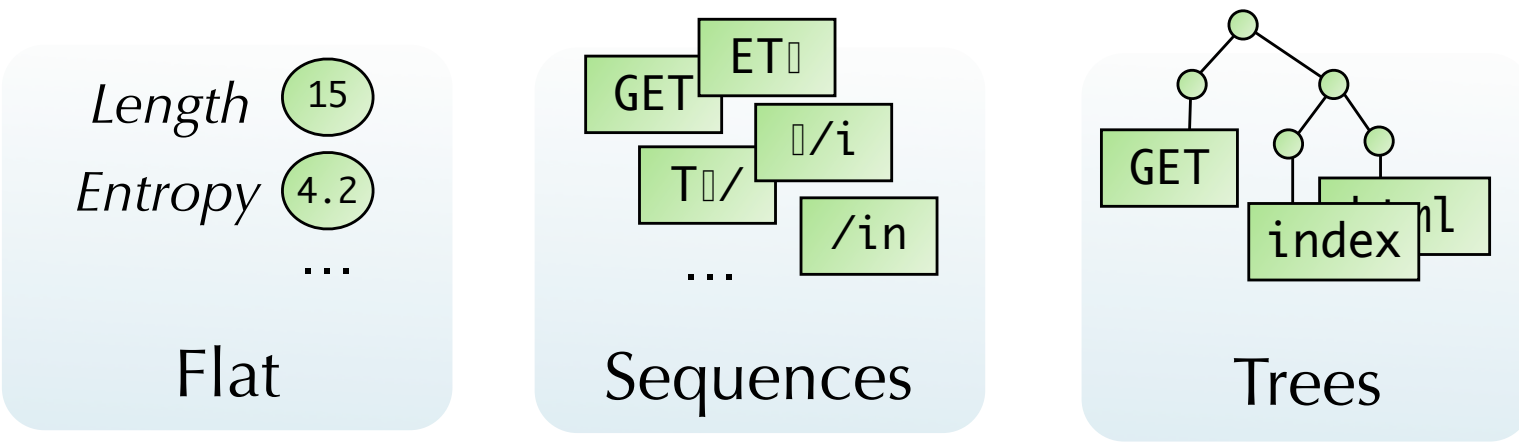
⇒ Huge amount of data (e.g. 1 GBit/s on a network link)

- Efficient feature extraction
- Efficient learning algorithms

Features for Intrusion Detection

Feature Extraction

Event x



Features $\Phi(x)$

Flat Feature Vectors

Embedding of events using set of numerical features

Feature map

A function $\phi : \mathcal{X} \to \mathbb{R}^n$ mapping events $\mathcal{X}$ to $\mathbb{R}^n$ given by

$$x \mapsto \begin{pmatrix} \phi_1(x) \\ \vdots \\ \phi_n(x) \end{pmatrix} \quad \begin{matrix} \text{feature 1} \\ \vdots \\ \text{feature } n \end{matrix}$$

Incorporation of categorical features via function ψ

$$\psi : C \to \mathbb{R}^{|C|}, \quad \psi(c) = \underbrace{(0, \ldots, 1, \ldots, 0)}_{1 \text{ at position } j} \text{ if } c = j\text{-th category}$$

Flat Feature Vectors

Data-dependent normalization of each feature $\phi_i(x)$ using mean μ_i and variance σ_i of feature.

$$\hat{\phi}_i(x) = \frac{|\phi_i(x) - \mu_i|}{\sigma_i}$$

Example: KDDCup 1999 data set (obsolete!)

Source	Features	Type
Connection properties	`duration, service, src_bytes, dest_bytes`	`int, bool, string`
Content features	`logged_in, root_shell, num_shells`	`int, bool`
Window features	`host_count, srv_count, error_rate`	`int, float`

Sequential Features

Embedding of sequential events, e.g. network packets

- Event x is sequence of symbols from alphabet $\mathcal{A}$
- Characterize x using a language $L \subseteq \mathcal{A}^*$
- Feature space spanned by frequencies of words $w \in L$

Feature map

A function $\phi : \mathcal{A}^* \to \mathbb{R}^{|L|}$ mapping sequences to $\mathbb{R}^{|L|}$ given by

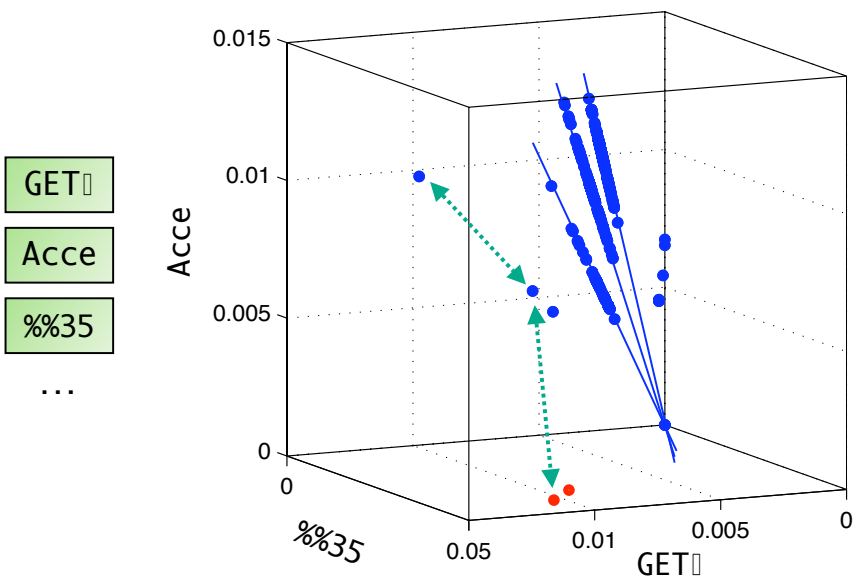
$$x \mapsto (\#_w(x))_{w \in L}$$

where $\#_w(x)$ returns the frequency of w in sequence x.

Sequential Features

Language $L = \mathcal{A}^n$ (N-grams)
$\Rightarrow$ Independent of attack and protocol characteristics

Example: 4-grams extracted from HTTP traffic



Attacks in lower front: presence of attack term "%%35" and absence of "Accept" keyword

Linear structures caused by HTTP pipelining

Tree Features

Embedding of tree events, e.g. *parsed* network packets

- Event x is parse tree of grammar G
- Characterize parse tree x using contained subtrees
- Binary feature space spanned by subtrees $t \in \mathcal{T}$

Feature map

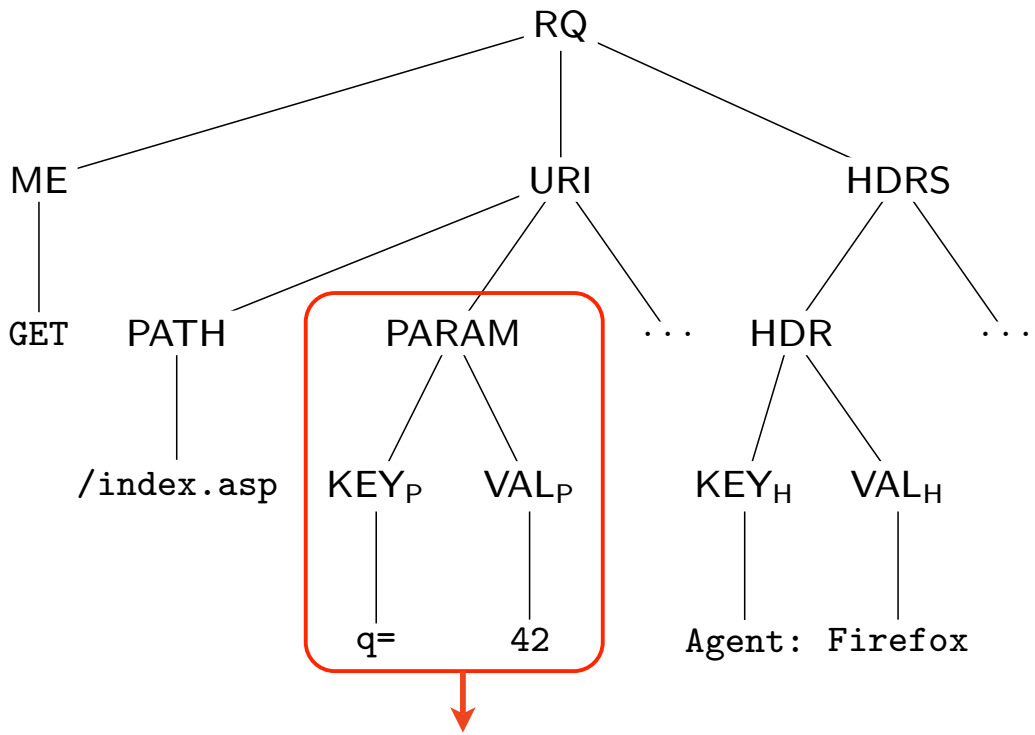
A function $\phi : \mathcal{T} \to \mathbb{R}^{|\mathcal{T}|}$ mapping trees to $\mathbb{R}^{|\mathcal{T}|}$ given by

$$x \mapsto (\mathbb{I}_t(x))_{t \in \mathcal{T}}$$

where $\mathbb{I}_t(x)$ indicates if t is a subtree of parse tree x.

Tree Features

Example: Extraction of parse trees for the HTTP protocol



$$\phi(x) = \underbrace{(0, 0, 1, 0, 0, 0, 0, 0, 0 \ldots)}_{\mathcal{O}(2^n) \text{ dimensions}}$$

From Features to Kernels

Incorporation of event data into learning methods via kernels

- Kernel functions for vectorial data
- Kernel functions for structured data ($\rightarrow$ previous lecture)

Examples:

- Linear kernel

$$k(x, y) = \langle \phi(x), \phi(y) \rangle$$

- Gaussian kernel

$$k(x, y) = \exp\left(-\frac{||\phi(x) - \phi(y)||^2}{\gamma}\right)$$

Efficient implementation using specialized data structures

Learning for Intrusion Detection

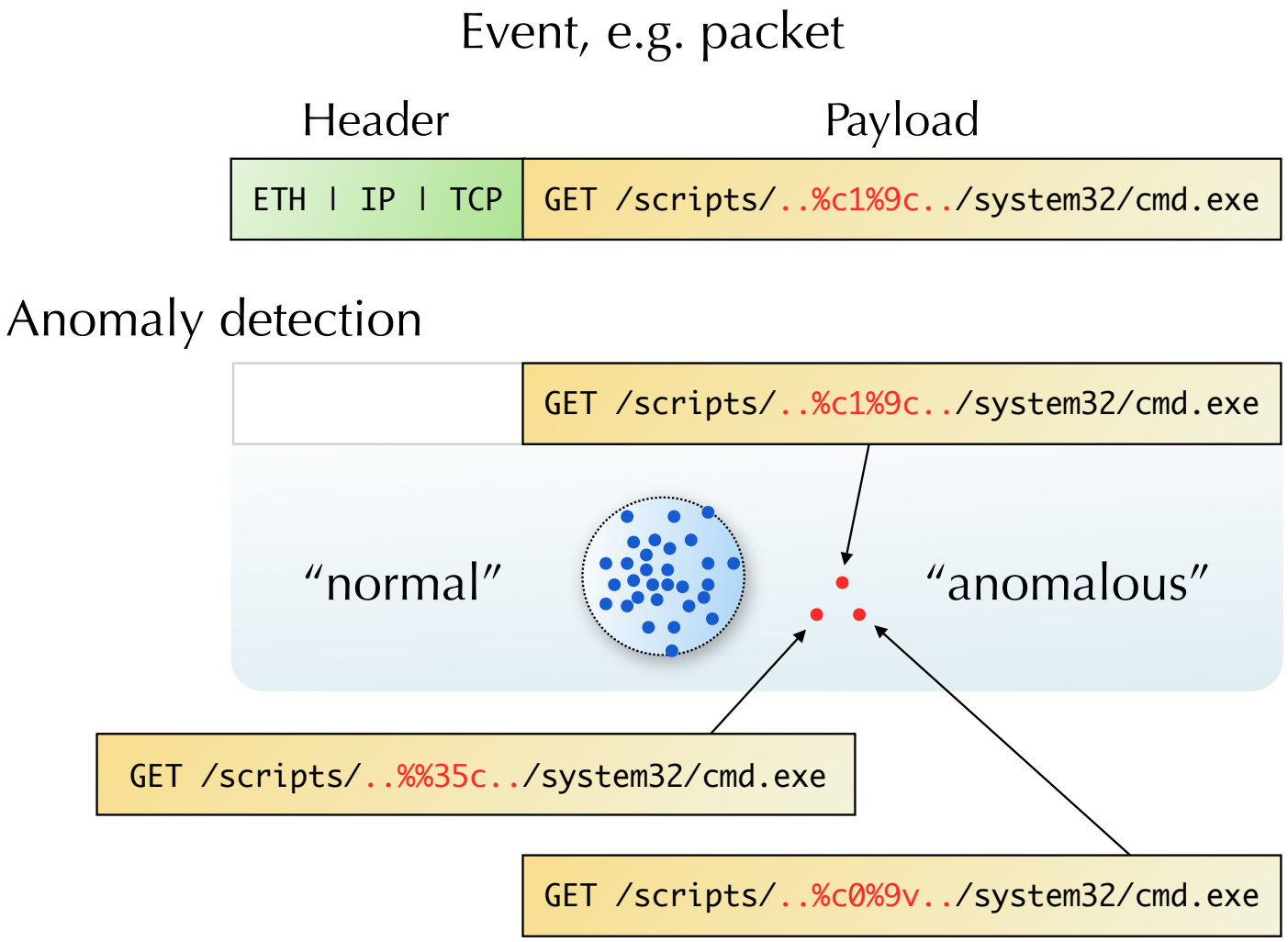
Learning Intrusion Detection?

Setup for learning

- No knowledge about future attacks → *One-class learning*
  Train detector on normal data only
  Assumption: attacks deviate from normal data
- Labeling real data expensive → *Unsupervised learning*
  Train on unlabeled "normal" data
  Assumption: low ratio of attacks in data
- Efficient computation → *Simple decision surfaces*
  Kernels for non-linear mappings to feature space

⇒ **Unsupervised Anomaly Detection**

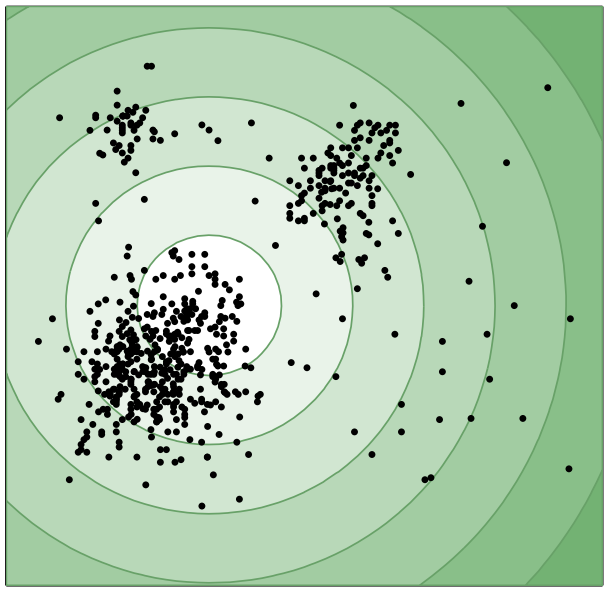
Unsupervised Anomaly Detection



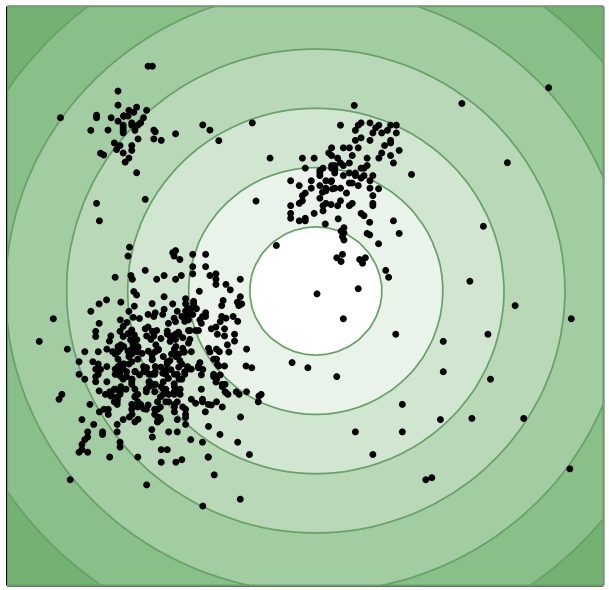
Hyperspheres for Anomaly Detection

Concept: Model data using hypersphere in feature space

Deviation of normality = Distance from center of hypersphere



(a) Center of Mass



(b) One-Class SVM

(a) Center of Mass

Model normality of data using center of mass

- For data $\{x_1, \ldots x_n\}$ center of mass $\bar{\mu}$

$$\bar{\mu} = \frac{1}{n} \sum_{i=1}^{n} \phi(x_i)$$

- Anomaly score $a(z)$ of new point z

$$\begin{aligned} a(z) &= ||\phi(z) - \bar{\mu}||^2 \\ &= k(z,z) - \frac{2}{n} \sum_{i=1}^{n} k(z, x_i) + \frac{1}{n^2} \sum_{i,j=1} k(x_i, x_j) \end{aligned}$$

(b) One-class SVM

Model data using hypersphere with minimum volume

- Determine smallest hypersphere with center μ^* and radius r

$$\begin{aligned} \min_{\mu,r} \quad & r^2 \\ \text{subject to} \quad & ||\phi(x_i) - \mu||^2 \leq r^2 \\ & \text{for } i = 1, \ldots, n \end{aligned}$$

- "Soften" margin of hypersphere using slack variables ξ_i

$$\begin{aligned} \min_{\mu,r,\xi} \quad & r^2 + C \sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & ||\phi(x_i) - \mu||^2 \leq r^2 + \xi_i \\ & \xi_i \geq 0 \quad \text{for } i = 1, \ldots, n \end{aligned}$$

One-Class SVM in Dual

Dual optimization problem given by

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i k(x_i, x_i) - \sum_{i,j=1}^{n} \alpha_i \alpha_j k(x_i, x_j)$$

$$\text{subject to} \quad \sum_{i=1}^{n} \alpha_i = 1 \quad \text{and} \quad 0 \leq \alpha_i \leq C \quad \text{for } i = 1, \ldots, n$$

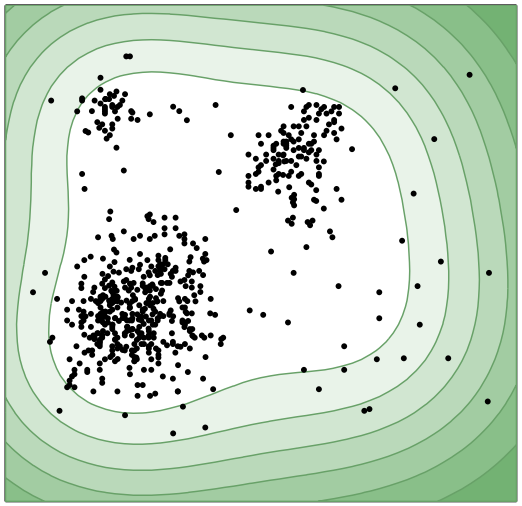
Formulation of anomaly value $a(z)$

$$\begin{aligned} a(z) &= ||\phi(z) - \mu^*|| \\ &= k(z,z) - 2\sum_{i=1}^{n} \alpha_i k(x_i, z) + \sum_{i,j=1}^{n} \alpha_i \alpha_j k(x_i, x_j). \end{aligned}$$

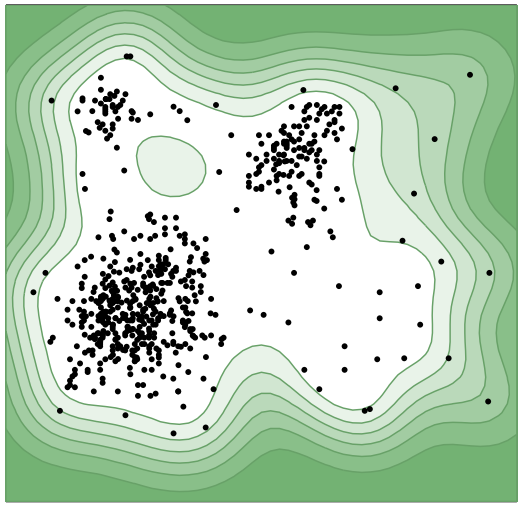
One-class SVM with Gaussian Kernel

Non-linear mapping of hypersphere using Gaussian Kernel

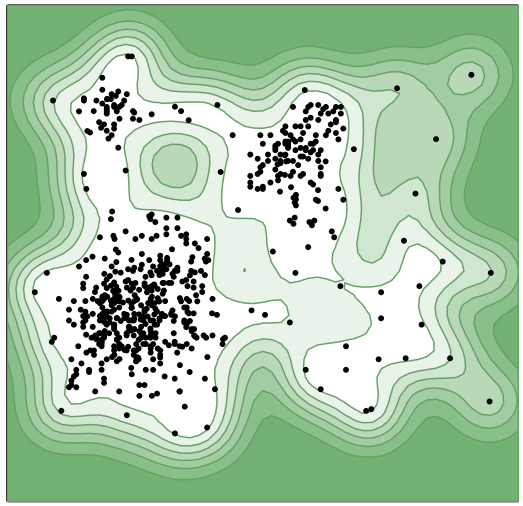
$$k(x, y) = \exp\left(-\frac{||\phi(x) - \phi(y)||^2}{\gamma}\right)$$



(c) $\gamma = 0.1$



(d) $\gamma = 0.01$

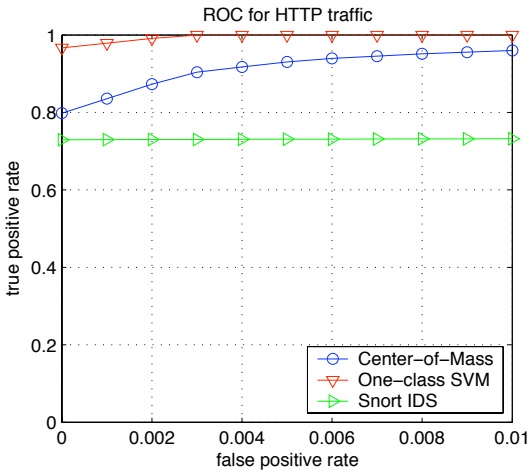


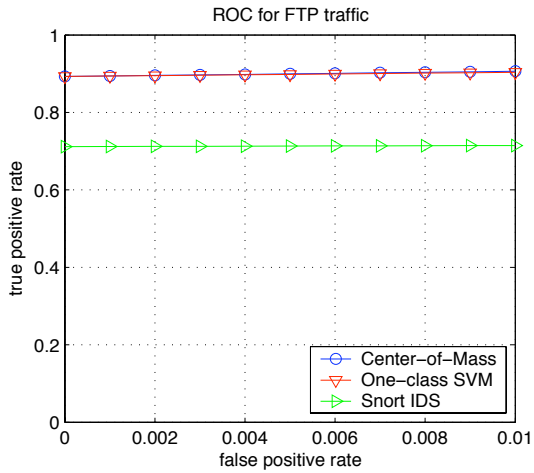
(e) $\gamma = 0.005$

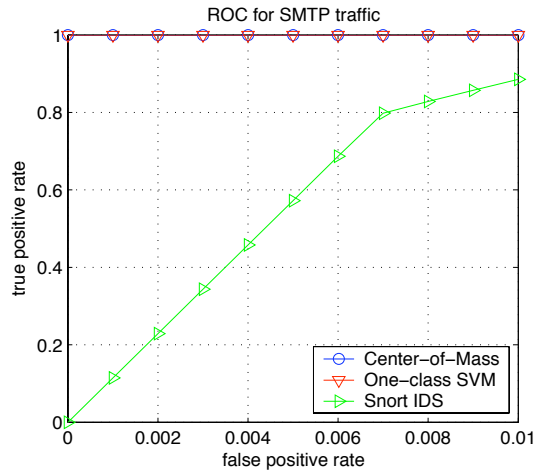
Results and Perspectives

Experiments with Network Intrusions

- Real network traffic generated by IDA members
- Attacks injected by security expert using Metasploit
- Setup: N-grams with Center-of-Mass and One-Class SVM

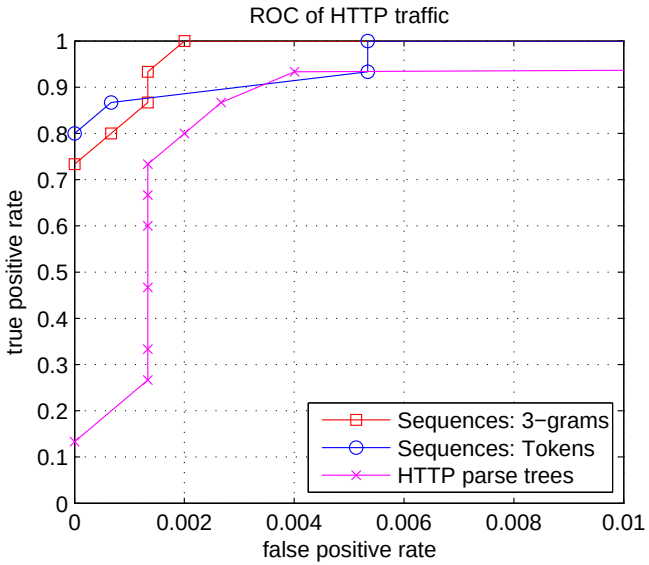






- High detection accuracy ($> 80\%$) with no false-positives
- Anomaly detection outperforms classic IDS “Snort”

Comparison of Structured Features



Structured features

- 3-grams from HTTP requests
- Tokens from HTTP requests
- Parse trees from HTTP requests

- Sequence features: High accuracy at low false-positive rates
  $\Rightarrow$ extracted subsequences reflect typical attack patterns
- Tree features: Moderate accuarcy due to false-positives
  $\Rightarrow$ structural but benign anomalies in HTTP traffic

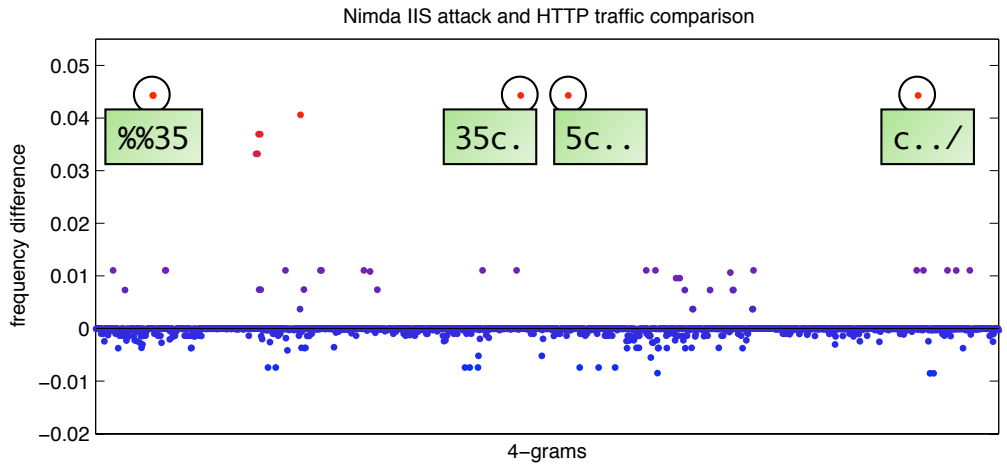
Explaination of Detection

Explaination using contribution of features to anomaly score

Network attack (Nimda Worm)

```
GET /scripts/..%%35c../system32/cmd.exe
```

Frequency differences of 4-grams



Attack pattern "%%35c" deviates from normal HTTP traffic

Conclusions and Outlook

Machine learning for intrusion detection

- Extension of classic signature-based detection methods
- Incorporation of flat, sequential and tree features
- Unsupervised anomaly detection using hyperspheres
  $\rightarrow$ high detection accuary with few false-positives

Perspectives: Further application of learning to security

- Real-time intrusion detection and response
- Malware analysis – Learning behavior and communication
- Protecting future communication (VoIP attacks & spam)

References

Eskin, E., Arnold, A., Prerau, M., Portnoy, L., and Stolfo, S. (2002).

A geometric framework for unsupervised anomaly detection: detecting intrusions in unlabeled data.

In *Applications of Data Mining in Computer Security*. Kluwer.

Rieck, K. and Laskov, P. (2007).

Language models for detection of unknown attacks in network traffic.

Journal in Computer Virology, 2(4):243–256.

Shawe-Taylor, J. and Cristianini, N. (2004).

Kernel methods for pattern analysis.

Cambridge University Press.