A Study on Various Signature Detection Methods to handle Zero-Day Worms

Swathy Akshaya .M, Dr. Padmavathi .G*
Avinashilingam Institute for Home Science and Higher Education for Women
akshayakulandaivel@gmail.com , {padmavathi.ganapathi, Padmavathi.avinashilingam}@gmail.com

Abstract

Zero-day polymorphic worms increasingly threaten the Internet hosts and services. Not only can they exploit unknown vulnerabilities but can also change their own representations on each new infection or can encrypt their payloads using a different key per infection. They have many variations in the signatures of the same worm thus, making their fingerprinting very difficult. This paper presents a survey on the researches to detect modern zero-day malware in the form of polymorphic worms.

Keywords: Zero-day, Zero-day attacks, Zero-day malwares, Signature Detections, Worms.

Introduction

Zero-day vulnerability, also known as a computer zero-day, is a flaw in software, hardware or firmware that is unknown to the party or parties responsible for patching or otherwise fixing the flaw. Zero-Day (0day) susceptibility is undisclosed computer software that could be exploited to affect hardware applications, data or networks negatively. Zero-day (day zero or Zero-hour) vulnerabilities are ones that have been previously obscure and not uncovered openly but rather are exploited by attackers. Cyber criminals are increasing the success rate of attacks by finding and exploiting Zero-day vulnerabilities. In most of these cases, the information is not available until attacks have already taken place. As a result, attacks using Zero-day exploits are hard to identify and analyse.

Currently security is about deliberate, targeted, well-resources threats, such threats use zero-day exploits as their common weapon for sophistication and stealth. According to Symantec's Internet Threat Report of 2013 there is 42% increase in targeted attacks in 2012, where 31% of all targeted attacks aimed at businesses. Also, the zero-day vulnerabilities continue to trend upward from the last three years with 14 zero-day vulnerabilities reported in 2012. Multiple zer0-day attacks occur each year.

In 2016, there was a zero-day attack (CVE-2016-4117) that exploited a previously undiscovered flaw in Adobe Flash Player. Also in 2016, more than 100 organizations succumbed to a zero day bug (CVE2016-0167) that was exploited for an elevation of privilege attack targeting Microsoft Windows.

In 2017, zero-day vulnerability (CVE-2017-0199) was discovered in which a Microsoft Office document in rich text format was shown to be able to trigger the execution of a visual basic script containing PowerShell commands upon being opened. Another 2017 exploit (CVE-2017-0261) used encapsulated PostScript as a platform for initiating malware infections.

The modern zero-day includes multi-vulnerability scanning to identify potential targets, complex mutation to evade defenses, targeted exploitation that launches directed attacks against vulnerable hosts.

Zero-day worms and polymorphic worms present a great challenge for computer and network administrators as well as researchers. Even the Botnets use the same techniques as the worms do. Botnets mainly launch a worm to compromise the computer machines. The best way to defend against these bots is to prevent the initial bot infection by detecting and containing the worms at the right time.

Zero-Day Attacks

The most dangerous zero-day exploits ever seen in cyberspace are as follows.

Hydraq Trojan- known as Aurora attack aimed to steal information from several companies.

Stuxnet- known as malware of the century and discovered in June 2010. It infects Programmable Logic Controller (PLC) in SCADA systems, contains 3 zero-day exploits, 3 root kits, and has military grade encryption and used unknown injection and spreading malware technique.

Duqu- related to Stuxnet worm discovered in September 2011. Part of this malware is written in unknown high level programming language. It exploits zero-day Windows kernel vulnerabilities, uses stolen digital keys and is highly targeted.

Flame- modular computer malware discovered in 2012 exploits same zero-day vulnerabilities in Microsoft Windows as Stuxnet. It is one of the most sophisticated and complex malware ever encountered.

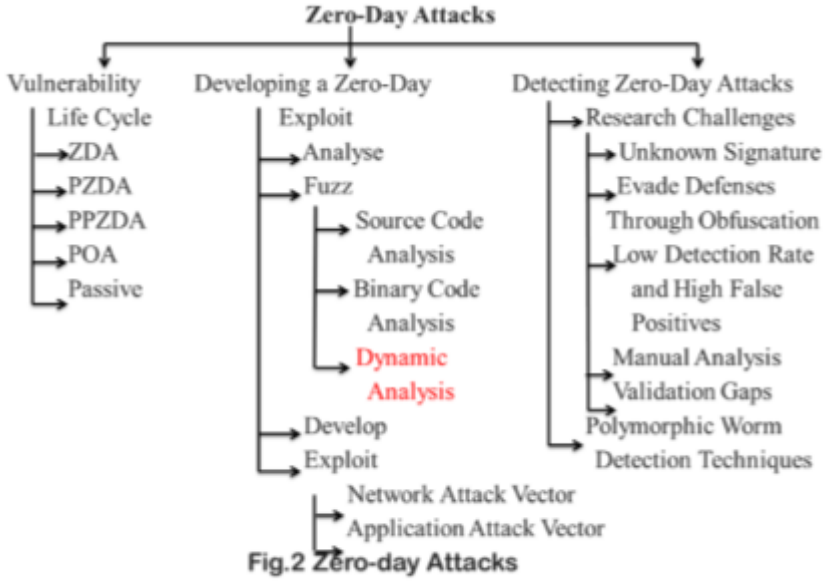

Fig.2 Zero-day Attacks

Figure 1: Zero-Day Attacks

Signature based Detection Techniques

Owing to simplicity and the ability to operate online in real-time it is preferred to employ worm detection systems which are network signature-based. The following diagram depicts the classification of zero-day polymorphic worms in different categories.

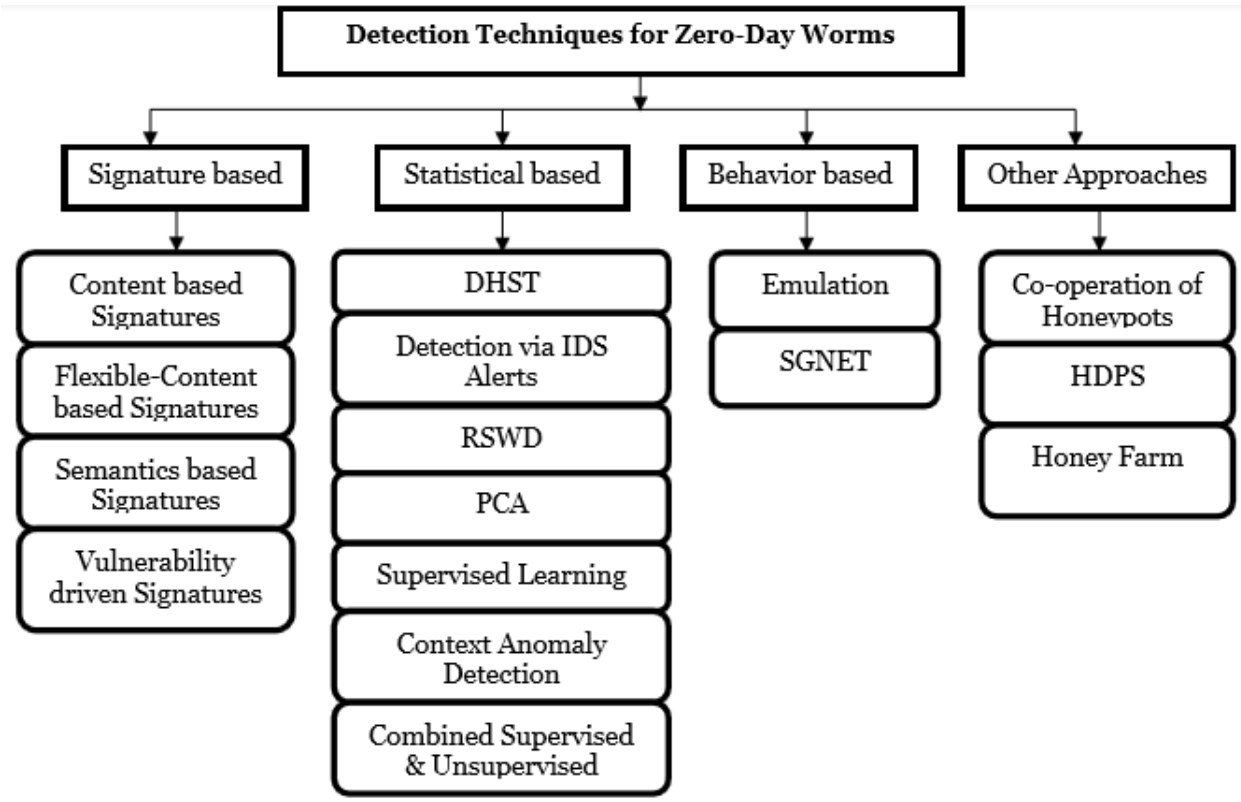

Figure 2: Detection Techniques for Zero-day Worms

A. Content-Based Signatures:

Content-based detection relies on using byte-pattern-based worm signatures to detect worm traffic. When the byte pattern of a given traffic flow matches the byte pattern defined by a worm signature, that traffic is identified as being worm traffic.

1) ***Polygraph:*** Polygraph is the first system designed to automatically generate signatures for polymorphic worms. It is based on the assumption that all the instances of the worm consist of multiple invariant substrings. Unlike Autograph and Early Bird, Polygraph uses multiple substrings to detect polymorphic worms which change the sequence of byte stream in every sample.

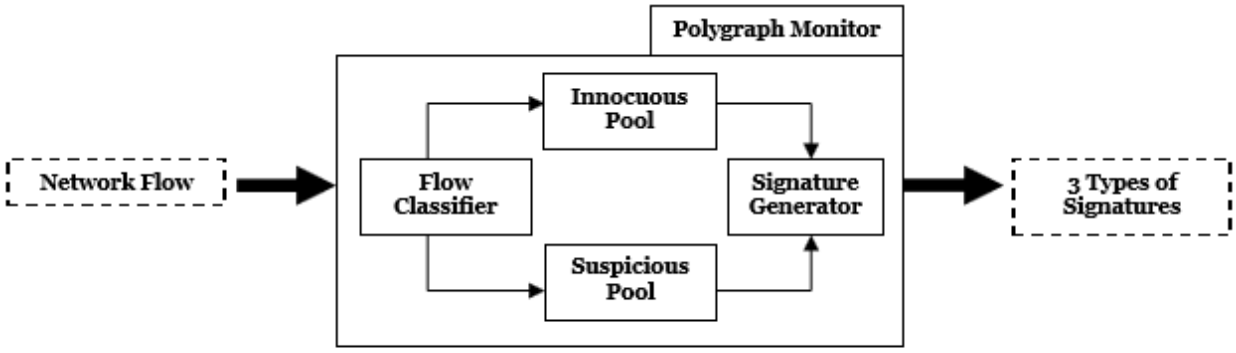

Figure 3: Architecture of Polygraph Monitor

2) ***Efficient content-based detection of zero-day worms:***
A novel method for detecting new worms based on identification of similar packet contents directed to multiple destination hosts.

3) *Hamsa:*
Hamsa is a network based automated signature generation system for zero-day polymorphic worms which is fast, noise tolerant and attack resilient. The content based signatures have fast signature matching algorithms and can be easily incorporated into firewalls or NIDSs.

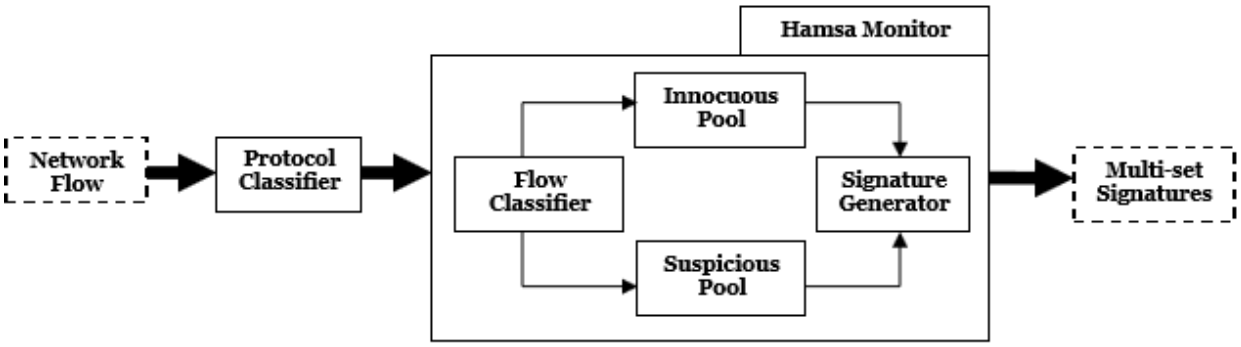

Figure 4: Architecture of Hamsa Monitor

4) ***Payload-based IDS:***
A new payload based IDS is introduced which, integrates header-based multidimensional flow clustering as front-end processing with content sifting (signature extraction) performed, separately, on each cluster in the subset of identified suspicious clusters. Front-end clustering improves the purity of the signature pools and also reduces complexity. A "suffix tree" approach is applied to signature extraction, gleaning both length and frequency information.

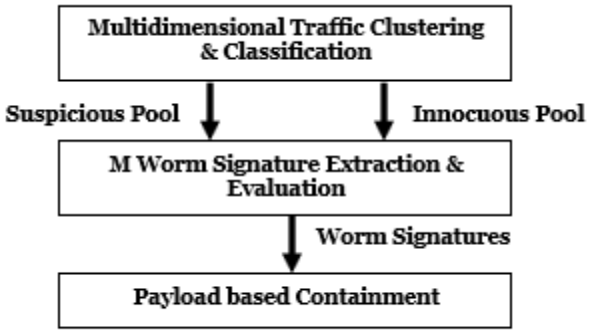

Figure 5: Structure of Payload-based Intrusion Detection System

5) ***Sweet Bait:*** It is a combination of network intrusion detection and prevention techniques. It employs different types of honey pot sensors, both high-interaction (Argos) and low-interaction (Sweet Spot) to recognize and capture suspicious traffic.

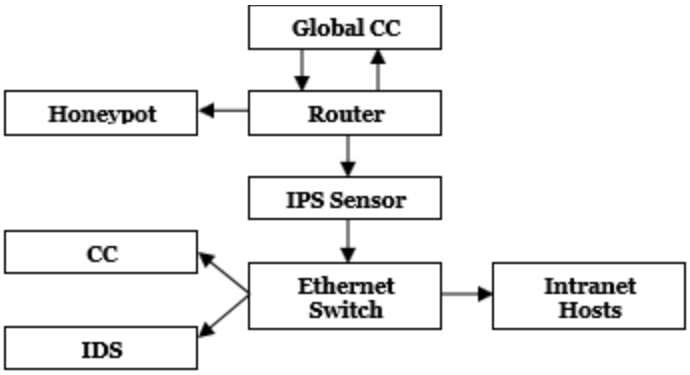

Figure 6: Architecture Overview of Sweet Bait

Sweet Bait automatically generates signatures for random IP address space scanning worms without any prior knowledge. And for the non-scanning worms Argos is used. Forensics shell code is inserted, replacing malevolent shell code, to gather useful information about the attack process.

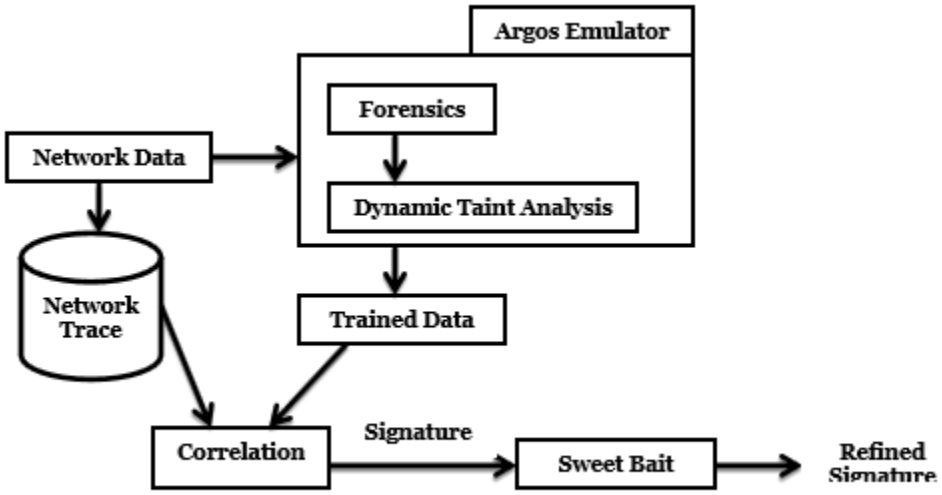

Figure 7: Argos: High-Level Overview

6) ***LISABETH:***
It is a network based automated signature generation system for polymorphic worms that uses invariant byte analysis of traffic content, as originally proposed in Polygraph and refined by Hamsa. LISABETH leverages on the hypothesis that every worm has its invariant set and that an attacker must insert in all worm samples all the invariants bytes.

7) ***Honey cyber:***
An automated system for signature generation for zero-day polymorphic worms, "Double-honey net" is proposed as a new detection method to identify zero-day worms and to isolate the attack traffic from innocuous traffic. The authors also introduced unlimited Honey net outbound connections to capture different payloads in every infection of the same worm.

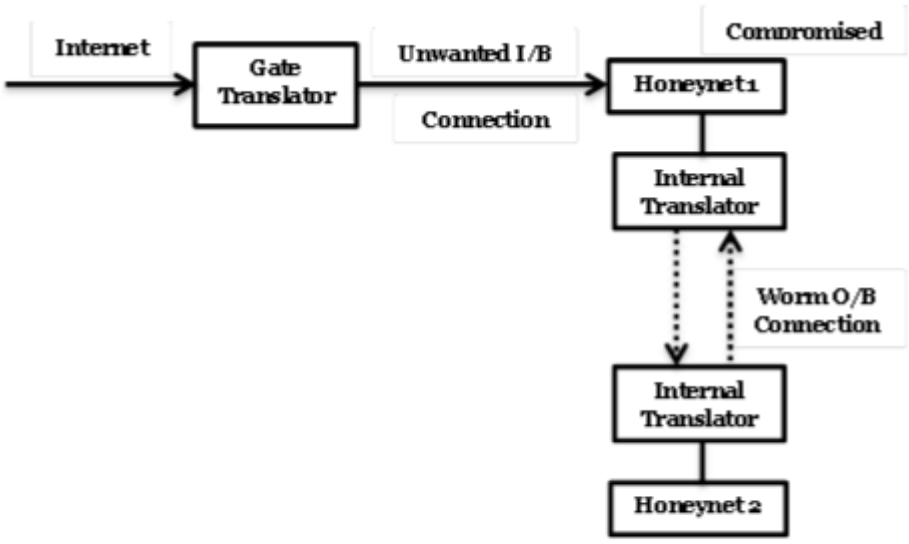

Figure 8: System Architecture of Honey cyber

8) Zero-day Attack Signature Management Infrastructure:

ZASMIN is an early detection system to detect unknown network attacks, the system adopted new technologies, composed of suspicious traffic monitoring, attack validation, polymorphic worm recognition and signature generation.

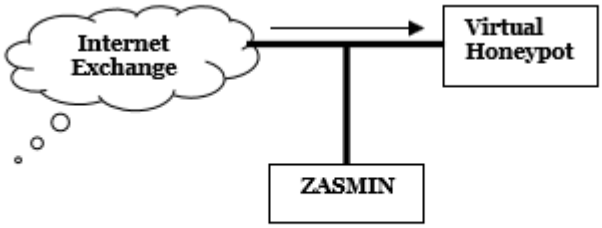

Figure 9: Honeynet Testbed in the National Internet Exchange Point

9) Bioinformatics:

Simplified Regular Expression (SRE) signature is proposed which is based on multiple sequence alignment; this system addresses the problem of generating accurate exploit-based signature for a single polymorphic worm.

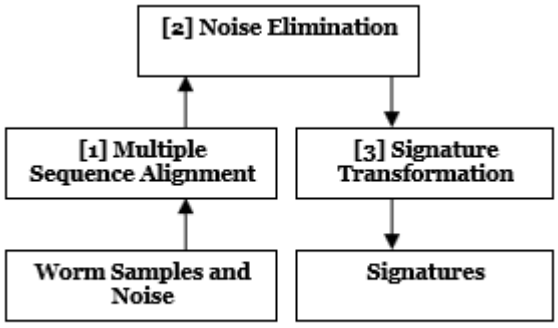

Figure 10: Bio-informatics Approach

10) Graph-based:

A graph based classification framework of content based polymorphic worm signatures. A new signature scheme, Conjunction of Combinational Motifs (CCM) is proposed to detect and create signatures for new versions of polymorphic worms.

Flexible content-based signatures:

These approaches work on a byte level and are flexible to match strings or substrings with incoming packets but, their signatures describe patterns of how malicious bytes are organized. Example techniques use byte-frequency distributions or regular expressions.

1) *PAYL:* An anomaly detection sensor detects and generates signatures for zero-day worms. It detects inbound anomalous loads, and correlates them with outgoing traffic on the same ports.
2) *Position Aware Distribution Signature (PADS):* This new signature has a byte frequency distribution instead of a fixed value for each position in the signature "string". PADS signature not only captures the static elements in the executable, but also captures the set of likely values for the variable elements.

Semantics-based signatures:

Instead of using repeated substring found in the network stream they either use the structure of the executable code present in the network stream or attack analysis information to generate Semantic signatures.

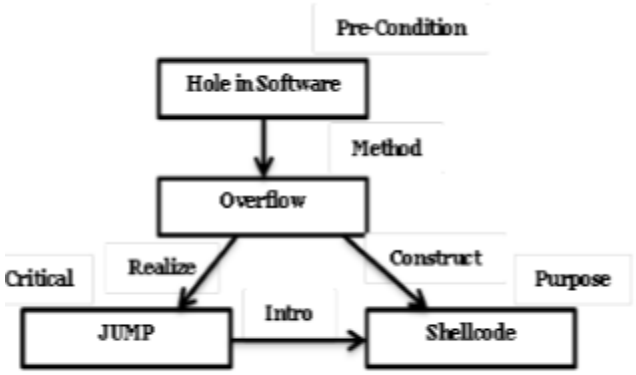

Fig.11 OSJUMP model

1) Structure of Executable Code Detection: An approach to generate signatures for detecting polymorphic worms, based on Control Flow Graph (CFG) of an executable code.

2) OSJUMP: A new worm attack model known as OSJUMP model detects online polymorphic worms by recognizing JUMP address using data-mining.

3) Sting: An end-to-end self-healing system automatically self-monitor its own execution behavior and detect errors or intrusions, self-diagnose the root cause of the error/intrusion, self-harden against further attacks and self-recover to a safe state.

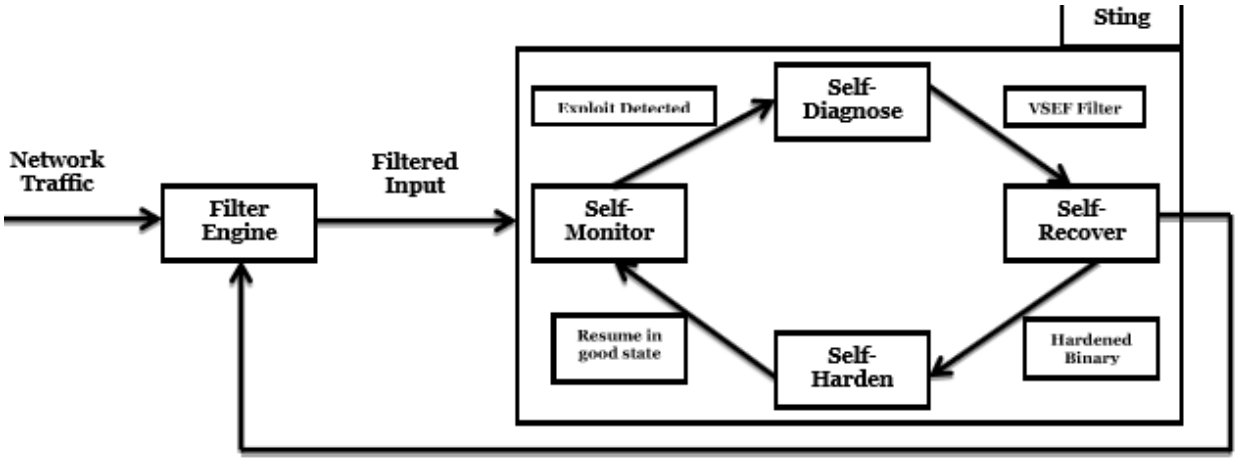

Figure 12: Sting Self-healing Architecture

Vulnerability-driven Signatures:

Vulnerability-driven signature captures the characteristics of the vulnerability worm exploits, it is inherent and hard to evade. LESG is the first network-based vulnerability-signature detection mechanism.

1) **Length Based Signature Generator (LESG)**: A network-based automatic worm signature generator that generates length-based signatures for buffer overflows worms. The designed system generates vulnerability-driven signatures at network level without any host-level analysis of worm execution or vulnerable programs.

Table.1: Comparative Analysis of Signature based Techniques

Parameters → / Techniques ↓	Robust	Efficient	Accurate	Computational Over-head	Noise Tolerant	Distributed/ Scalable	Response Time
Polygraph	Robust against reordering , insertion, deletion of worm bytes	If samples in pool ¿2 then 100% efficient, if ¡2 then 0%	Increase in FP and FN for noise beyond 80%	Uses greedy approach for signature generation to reduce computation cost	Generate quality signatures for noise flows <80%	-	-

Efficient-Content based	-	Detection delay causes 7%-14% of worm infection	Zero FP for substring length ¿ 150 bytes and cache size i 600 msec	Reduces computation overhead by using fingerprint selection	-	-	-
Hamsa	Allows to choose any bytes for the variant part of the worm	On-line detection speed is fast than CFG	Average and maximum FP rate is 0.09% and 0.7% respectively	Noise increases computation overhead	Generates quality signatures with 20% noise in suspicious pool	-	-
Payload IDS	-	Uses efficient tree computing to detect worms	Shows zero FP	Computationally efficient then Early bird and Polygraph	-	-	-
Sweetbait	Robust against polymorp hic worms	Produces 140,000 Alerts per second.	Zero FP	Modifying Aho-Corasick algorithm [44] increased overhead	-	Distributed Architecture. Scalability is left to future work	Take 2 mins to update signatures to NID & NIP
LISABETH	Resilient to attacks over Hamsa	Signature generation takes 20% less time than Hamsa	Average FP rate is 0.095%	Reduces overhead by avoiding redundant and useless signatures	Create correct signatures with low FP rate in presence of noise	-	-
Honey cyber	Captures all worm instances	Average detection rate is 90%	Zero FP and low FN	-	-	-	-
ZASMIN	Tries to find GetPC instructio n to detect polymorp hism	Improves detection rate by using parallel detectors to recognize malicious code	FP rate is 2.3%.	-	-	-	-
Bio-informatics	Robust against worms containing very short Invariant bytes.	Proposed efficient signature (SRE) for worm Matching.	Almost zero FP and zero FN	Gives better performance for worm samples > 4	Accurate when noise ratio is less than 57%	-	-
Graph based	Robust against new versions of known polymorp hic Worms.	Better detection time than Polygraph and STP [61]	Almost zero FP and zero FN	-	-	-	-
PAYL	Able to detect polymorp hic worms	Nearly 85% detection rate	0.1% FP rate	LCS and LCSeq introduces computational overhead	Even in presence of noise the actual worm content is separated	-	-
PADS	Able to detect polymorp hic worms	Proposed two algorithms to efficiently compute	FP ratio is nearly 0.0003	Signature matching causes high overhead	In presence of noise accuracy suffers	-	-

		PADS					
CFG	Robust to insertion, deletion and modification of worm code	Detection rate is 16.97%	Accuracy varies with threshold value; high threshold, low false alarm rate	Matching fingerprints is computationally expensive	-	-	-
OSJUMP	Detects polymorphic worms with same or different JUMP address	Improves 18% detection rate than Snort	Nearly zero FP with 0.00021% worst positive alarm rate	Data-mining produces 2-3 times less overhead than with string matching.	-	-	-
Sting	Detects polymorphic and metamorphic zero-day worms	Efficient due to the low overhead of the light-weight detectors	Zero FP rate	Few processes add 10% overhead	-	-	Response time for new attack is roughly 1 sec
LESG	Able to detect zero-day polymorphic worms	Implemented efficient length-based signature matching	FP rate is 0.15%	With noise and multiple worms computational complexity increases	With 90% noise the system has FN rate of 6.3% and FP rate of 0.14%	-	-

Statistical - Based Detection Techniques

A. Distributed Sequential Hypothesis Testing:
A distributed and decentralized detection technique is designed uses a statistical tool called (dSHT) to build a strong distributed worm detector from imperfect anomaly detectors that have high false positive rates. The two most important components of worms are defense- detection and response.

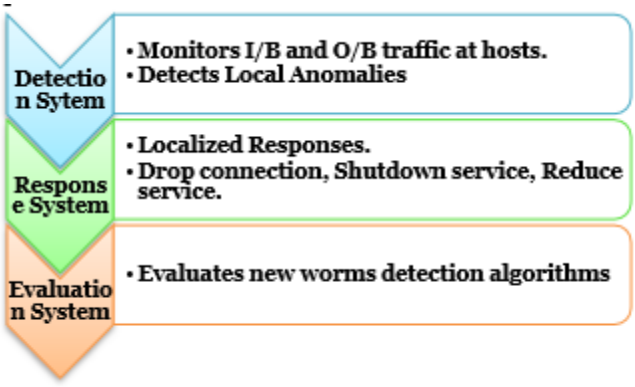

Figure 13: dSHT System Components

B. Detection via IDS Alerts:
The basic features of IDS alerts such as source address and port, destination address and port, detection time and signature name. To detect 0-day attacks from IDS alerts an unsupervised learning technique, One-class Support Vector Machine (SVM) along with Sequential Minimal Optimization (SMO) algorithm is applied.

C. Rough Set Worm Detection:
It provides a minimum set of filtering rules for network barrier equipment to block the worm spreading.

D. Principal Component Analysis:
It is a technique for detecting new attacks in low-interaction honey pot traffic. It is based on (PCA), a widely used multivariate statistical technique for reducing the dimensionality of variables and unveiling latent structures and detecting outliers in data sets.

E. Supervised Learning:
A machine learning framework is developed using eight different classifiers, namely Naive Bayes (NB) Algorithm, k-Nearest Neighbour (KNN) Algorithm, Sequential Minimal Optimization (SMO) Algorithm with 4 different kernels (SMO-Normalized PolyKernel, SMO-PolyKernel, SMO-Puk, and SMO-Radial Basis Function (RBF)), Back propagation Neural Networks Algorithm, and J48 decision tree.

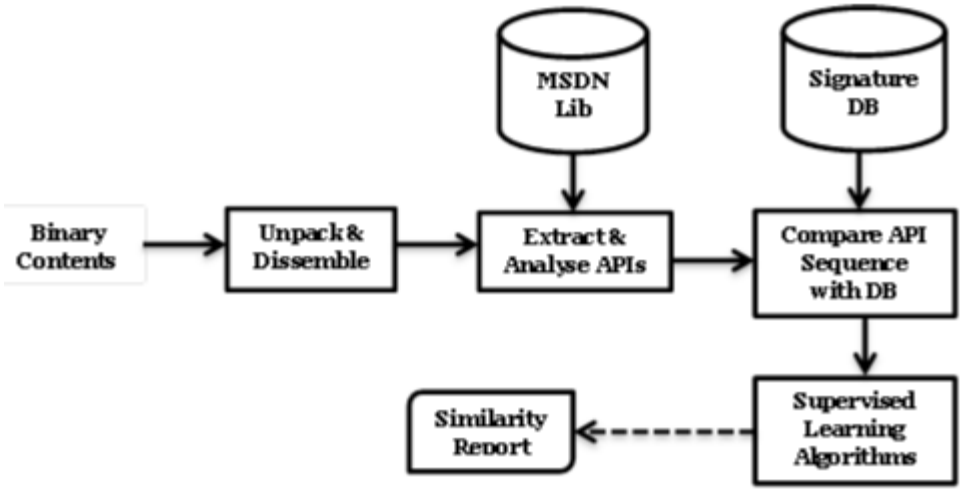

Figure 14: System Overview of Zero-day Malware Detection

F. Contextual Anomaly Detection:
A contextual misuse and anomaly detection prototype is used to detect zero-day attacks. The contextual misuse detection utilizes similarity with attack context profiles, and the anomaly detection technique identifies new types of attacks using the One Class Nearest Neighbour (1-NN) algorithm.

G. Combined Supervised & Unsupervised:
Two-level malware detection is proposed that utilizes supervised classification to detect known malware and unsupervised learning to detect new malware and known variants. A tree-based feature transformation is also introduced to overcome data imperfection issues and to detect the malware classes effectively.

Table.2: Comparative Analysis of Statistical based Techniques

Parameters / Techniques	Robust	Efficient	Accurate	Computational Over-head	Noise Tolerant	Distributed/ Scalable	Response Time
DSHT	Able to detect zero-day polymorphic worms	Detection rate of local IDSs is set to 0.9	FP rate of local IDSs is set to 0.1	Low computation due to distributed algorithms & cooperative systems	-	Distributed & decentralized system	Usually takes long time to claim a worm
Detection via IDS Alerts	-	Detection time is long and irregular.	High false alerts and detects 0-day attacks by adjusting parameters	-	Noise (port scan, sweep activities) misclassified as FP	-	-
RSWD	Able to detect zero-day polymorphic worms	Detection time varies from 15 to 17 seconds	FP rate lies between 0.35% and 1.67%	A real time clustering algorithm reduces computation time	-	-	-
	-	-	Depends on threshold	Average execution time	Internet noise (scans,	-	-

PCA based			value	lies b/w 0.000088 to 0.070073 seconds	backscatter) is filtered out		
Supervised Learning	-	Detection rate lies between 71% to 86%	FP rate ranges from 0.08 to 0.22	Modifications on 1-NN reduces computational time	-	-	1-NN takes time to detect zero-day attacks
Contextual Anomaly Detection	Robust against various types of obfuscation	99% detection rate	FP rate is nearly 0.025	-	-	-	-
Combined Supervised &Unsupervised	Robust against polymorphism	91% detection rate	0.1 FP rate	-	A tree-based kernel for classifiers handles noise	-	-

Behavior-Based Detection Techniques

A. Network-level emulation:

Detection method to scan network traffic streams for the presence of previously unknown polymorphic shell code. Their approach relies on a NIDS-embedded CPU emulator that executes every potential instruction sequence in the inspected traffic, aiming to identify the execution behavior of polymorphic shell code.

B. SGNET:

SGNET is a distributed framework to collect rich information and malware for on-going attacks and zero-day attacks. SGNET uses an automated script generation tool for honeyed to automatically generate approximations of the protocol behavior under the form of Finite State Machines (FSMs).

The honey pot acts as a proxy for the real host, allows building samples of network conversation for the new activity that are then used to refine the current FSM knowledge.

Table.3: Comparative Analysis of Behavior based Techniques

Parameters → / Techniques ↓	Robust	Efficient	Accurate	Computational Over-head	Noise Tolerant	Distributed/ Scalable	Response Time
Emulation	Robust to self-modifying & non self-contained polymorphic	-	Zero FP	Implemented performance optimizations to reduce execution overhead	-	-	-
SGNET	Able to detect polymorphic worms and executable payloads	Requires training for few days to detect new attack	Broken malware downloads were considered as FP	-	-	Distributed honeypot framework & can be scaled	Takes approx. 7 days to generate FSM

Other Detection Techniques

A. Cooperation of Intelligent Honey pots:

Two types of honey pots; Cooperation based active honey pot and Self-Protection type honey pot, are used to collect unknown malicious codes automatically while maintaining their concealment against malicious attackers.

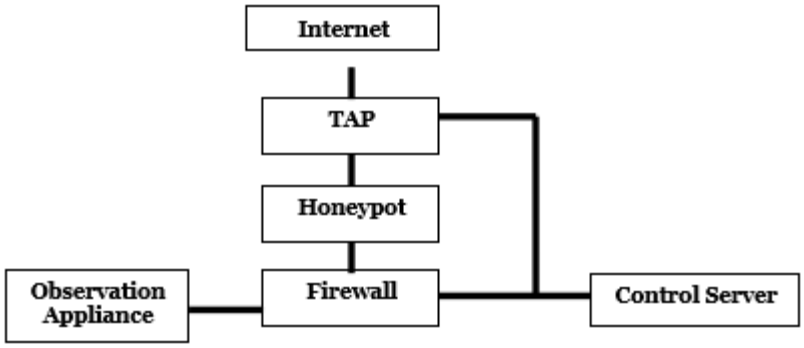

Figure 15: Cooperation-based Active Honey pot

B. Hybrid Detection for zero-day Polymorphic Shell codes (HDPS):
A hybrid detection approach for zero-day polymorphic shell codes, to detect return address and to filter mass innocent network flows. HDPS applies a sliding window to detect return address. It uses Markov Model to detect the existence and location of executable codes in suspicious flows and applies a sliding window to identify the executable code.

C. Honey farm:
A hybrid scheme that combines anomaly and signature detection with honey pots. This system takes advantage of existing detection approaches to develop an effective defense against Internet worms.

Table.4: Comparative Analysis of Other Detection Techniques

Parameters → / Techniques ↓	Robust	Efficient	Accurate	Computational Over-head	Noise Tolerant	Distributed/ Scalable	Response Time
Co-operation of Honeypots	Withstands even if the honeypot is compromised	Takes 4 to 5 minutes to download malicious code	Detected unknown attacks	-	-	-	-
HDPS	Robust against polymorphism, Metamorphism and other obfuscation.	Efficient	Nearly zero FP rate	Classify packets to reduce Over-head for later Analysis.	-	-	-
Honey Farm	-	Detection rate is 81%	False alarm rate is 4%	-	-	-	-

Conclusion

Zero-day attacks are like invisible arrows, they can't always be spotted on the way. Thus, a way must be figured out to remove the arrow after it is logged in back while limiting the damage. The following factors need to be considered to avoid pitfalls and to efficiently detect modern zero-day malware.

1. **Comprehensive real-time Network Analysis and Visibility (NAV):** Modern Zero-day attack is based on understanding the multidimensional analysis of various networks in real-time and to make auto corrective decisions.
2. **Code-level analysis:** Detection and investigation of zero-day malware are based on source code and binary code analysis. Binary analysis is more beneficial than source analysis.
3. **Detection rules:** The developed rule should be tested, modified again and again to reduce false positives and false negatives.
4. **VM-based analysis:** It provides effective isolation of critical environment with cost effective security devices.
5. **Data-level protection:** Authentication, Authorization, Access auditing and analysis are the three control areas should be more focused to avoid zero-day intrusions.
6. **Secure code:** It is emphasized on application security, better applications and security code.

References

RatinderKaur and Maninder Singh (2014) A Survey on Zero-Day Polymorphic Worm Detection Techniques, IEEE.

Ratinder Kaur and Maninder Singh (2016), Hybrid Real-time Zero-day Malware Analysis and Reporting System, I.J. Information Technology and Computer Science. Is. No. 4 Pg. 63-73

Georgios Portokalidis, Asia Slowinska, Herbert Bos (2006) Argos: an Emulator for Fingerprinting Zero-Day Attacks, ACM.

Dalla Preda, Mihai Christodorescu and Somesh Jha, Saumya Mila Debray (2007) A Semantics-Based Approach to Malware Detection. ACM

Manuel Egele, Theodoor Scholte, Engin Kirda, Christopher Kruegel (2012), A Survey on Automated Dynamic Malware-Analysis Techniques and Tools. ACM Computing Surveys, Vol. 44, No. 2, Article 6

P. Akritidis, K. Anagnostakis, E.P.Markatos (2005), Efficient Content-Based Detection of Zero-Day Worms. IEEE

ZhichunLi, MananSanghi, YanChen, Ming-YangKao, BrianChavez (2006) Hamsa: Fast Signature Generation for Zero-day Polymorphic Worms with Provable Attack Resilience. Proceedings of the 2006 IEEE Symposium on Security and Privacy (S&P'06), pg.1081-6011/06. IEEE

Razieh Eskandari, MehdiShajari, Asadallah Asadi (2015) Automatic Signature Generation for Polymorphic Worms by Combination of Token Extraction and Sequence Alignment Approaches. IKT 2015 7th International Conference on Information and Knowledge Technology. Pg. 978-1-4673-7485-9/15.IEEE

Brightstarlang Wanswett, Hemanta Kumar Kalita (2015) The Threat of Obfuscated Zero Day Polymorphic Malwares: An Analysis. International Conference on Computational Intelligence and Communication Networks, pg. 978-1-5090-0076-0/15, IEEE

Sai C. Pallaprolu, Rishi Sankineni, Muthukumar Thevar, George Karabatis, Jianwu Wang (2017) Zero-day Attack Identification in Streaming data using Semantics and Spark. IEEE 6th International Congress on Big Data, pg. 978-1-5386-1996-4/17, IEEE

Sanjeev Das, Yang Liu, Wei Zhang, Mahintham Chandramohan (2015) Semantics-based Online Malware Detection: Towards Efficient Real-time Protection against Malware. IEEE Transactions on Information Forensics and Security, pg. 1556-6013, IEEE

Gil Tahan, Chanan Glezer, Yuval Elovici, Lior Rokach (2010) Auto-Sign: an automatic signature generator for high-speed malware filtering devices. J Comput Virol, pg. 6:91–103, Springer