CYBER AWARENESS IMPROVEMENT USING ARTIFICIAL INTELLIGENCE

Amandeep Kaur[1], Asst Prof. Ms Deepti Choudhary[2]
[1]M.Tech(CSE) - III semester (OBJECT ORIENTED SOFTWARE ENGINEERING).
[2]Department of Computer Science & Engineering, Kalinga University, Naya Raipur (C.G).

***ABSTRACT:** As this is the 21st century, the term "globalization" has become popular. Simply put, globalization is the process of international integration. Under this, as nations, people have lots of factors that must be definitely considered. Among those important areas, information technology (IT) is something that is being evolved day by day. As a negative aspect of IT, with the help of technological advancements, criminals are using cyberspace to commit numerous cyber-crimes. Since people are connected to the cyber space with their own devices, they are all vulnerable to intrusions and other various kinds of threats. Basic protection methods, such as internet security suits, are not just enough to protect the data and devices. Introducing effective and highly advanced cyber defense systems has become essential. As of today, with the technology, the globe is moving towards the artificial intelligence (AI). AI plays a major role in technology and has been involved with many technological aspects as well. Creating cyber defense systems, using intelligent agents has become a trend by today. Basically, an intelligent agent is a software component which can be emerged in an environment, take decisions, and has the ability of noticing and representing. The purpose of this study is to introduce a sophisticated cyber-crime defense system which involves intelligent agents that are based on artificial intelligence.*

I. INTRODUCTION

This research paper mainly focuses on how to combat cybercrimes, and also it demonstrates how intelligent and effective the tool "agent" that can be used in detection and prevention of cyber-attacks. Cyber-attacks tend to have a huge impact on the IT industry when it comes to data theft, many societies across the world have components or systems which depend on web applications. As web applications are used increasingly on basic and critical activities they have become a very vulnerable and a popular target for security attacks. It can be noticed that the increase of cyber-attacks are very high in today's cyberspace. Any action that bypasses the security mechanisms of the targeted system using a computer and a network can be defined as a cybercrime. In a cybercrime the computer might be used as an intruder or it can be the target. Most of the existing studies focus on models based on estimation and control theories. In these models, future process loads and traffics are estimated to find how to reach a desired state and their efficiency is ensured to optimize certain criteria. However, such methods and their subsequent models are generally in the continuous and sequential control task traffic mode which is not the focus of the current experiment. In this experiment, it is assumed that as soon as the process load/traffic manager starts learning and acting, it progresses towards the desired status at a constant rate, without optimization involvement. Therefore, the prime interest is whether a high index/load process thread can be considered to be non-mask able at any point of time, even if it causes a dangerous situation, or if it can be simply ignored. This aspect of cyber security falls into the category of multitasking and load management control. This experiment considers the load management routine as a single task and it uses a selective task scheduling approach, which allows for a learning system based on algorithmic comparison and pre-emptive task selection.

II. IDENTIFY, RESEARCH AND COLLECT IDEA

Cyber attacking, it is a common word used in the present world. Daily thousands of computer networks or computer systems get attacked by an unknown systems or hackers in order to damage or destroy the system. In order to prevent and detect such attacks many systems are being developed. A background study was done in order to identify the available technologies, mechanisms etc.

A. Intrusion Detection System:

An Intrusion Detection System or IDS is a network security technology originally built for spotting vulnerabilities that exploit against a targeted application or a computer system. It is the process of monitoring the events occurring in a computer system or in a network and analyzing them for possible incidents indications, which are violations or impending threats of destruction of computer security strategies, suitably used policies, or common security practices. An ID system gathers and analyzes information from various sources within a computer or a network to identify possible security breakings, which include both intrusions and attacks from the outsiders the organization and does not use them properly or attacks within the organization. Particular intruders can be pin pointed and shown through an algorithm. Intrusion detection system only can identify intrusions, and it cannot prevent the system from attacks. It should be fast enough to identify the intruders (external or internal intruders) as soon as the attack is going on. In IDSs efficiency is a more important feature. Intrusion Detection System (IDS) technologies are not very effective as there are several limitations, such as performance, scalability and flexibility. Intrusion Prevention System (IPS) is a new approach to defense networking systems.

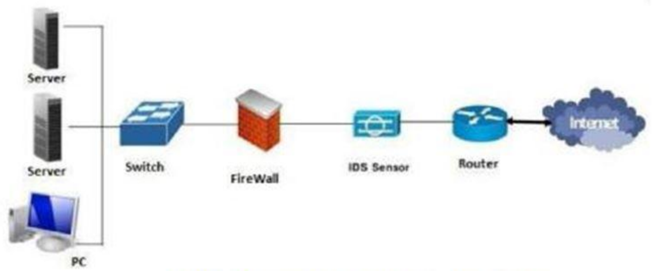


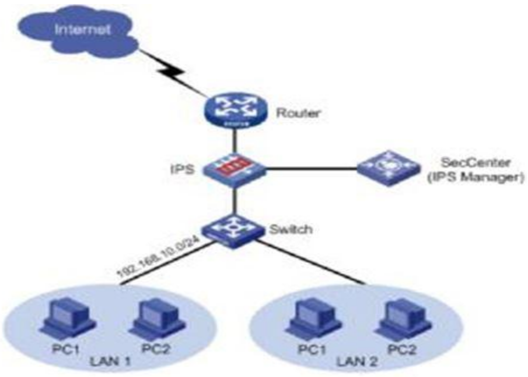
Figure 01 indicates how AN ID is placed in a system.

B. Intrusion Prevention System:

Intrusion prevention systems or IPS, also known as intrusion detection and prevention systems or IDPS, are network security appliances that monitor networks and system activities for malicious activities. The IPS often lies directly behind the firewall and provides a complementary or integral layer of analysis that selects for dangerous contents. Intrusion prevention is a preemptive approach in network security which is used to identify potential threats and respond to them swiftly. Like an intrusion detection system (IDS), an intrusion prevention system (IPS) checks and controls network traffic. However, because an exploit may be carried out quickly after the attacker gains access, intrusion prevention systems also have the ability to take immediate actions, it's about a bunch of rules created by the network administrator. As an example, IPS might drop a packet that it determines to be malicious and block all further traffic from that IP address or port [9]. Legitimate traffic, meanwhile, it should be sent forward to the recipient with no sudden interruption or delay of service. Unlike its predecessor the Intrusion Detection System (IDS) is known to be a passive system that scans traffic and alerts back the threats the IPS is placed intact with (in the direct communication path between source and destination), automated actions will be taken on entire traffic flows that enter the network by actively analyzing them. Specifically, these actions include:

- Dropping the malicious packets;
- Sending an alarm to the administrator;
- Blocking traffic from the source address;
- Resetting the connection.

The IPS should work properly, as it one of the main frontline components used to avoid the degrading of network performance. It must also work fast because exploits could be caused in real-time. The IPS must also spot and react precisely, so it can eliminate threats and false positives. Figure 02 indicates how an IPS is placed in a networking environment.

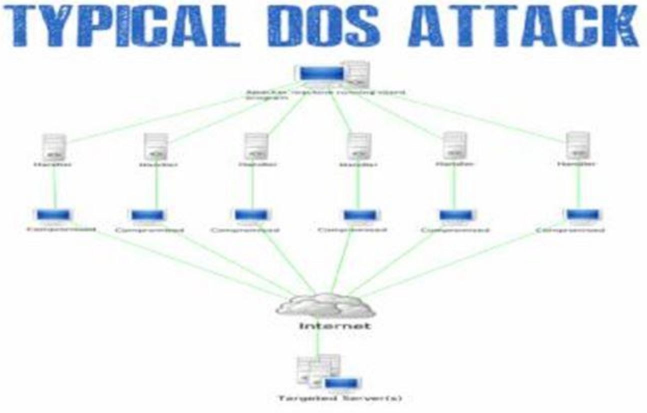


C. Cyber Security System / Cyber Attack Detection Systems (CADS): -

Cyber Attack Detection Systems (CADS) and its generic framework perform well for all the classes. This is based on Generalized Discriminate Analysis algorithm (GDA) for feature decrement of the cyber-attack datasets and a collective approach of classifiers for classification of cyber-attacks [1] [10]. Cyber Attack Detection System is having improved detection accuracy for all the classes of attacks.

D. Detects denial-of-service (DOS) attacks: -

A DOS attack is an attack type which is used to make a computer or a network resource unavailable to the users, such as to temporarily or permanently interrupt or suspend services of a host connected to a network. By targeting user's computer and its network connection, or the computers and network of the sites the user is trying to use, an attacker may be able to prevent the user from accessing email, websites, online accounts (banking, etc.), or other products and services that reside on the affected computer. The most common type of DOS attack is a situation where an attacker floods a network with information. When the user types an URL for a particular website into web browser, he is sending a request to that site's server to view the page. Only a certain amount of requests will be processed by the server at a time, therefore requests will not be processed if an attacker swamps the desired server with. This is known as a "Denial of Service" because the user will not be able to access that site. There are many other types of cyber-attacks such as brute force attacks, browser attacks, shellshock attacks, SSL attacks, backdoor attacks and dotnet attacks[3] [6] [7]. Figure 03 illustrates the mechanism of a Typical DOS attack.



E. Agent Based / Artificial Agent: -

An entity that can be activated, autonomous and has the capability of formulating inner judgment can identify as an agent. An agent is a software program that gives assistance to user to complete some tasks or activities. Agents in a multi-agent system (MAS) must be able to cooperate and work together with every user of the system [4] [8]. Therefore, a common language is requisite for the purpose of communication, an Agent Communication Language, or ACL can be used for this. Intelligent agents are software components which have special features of intelligent behavior such as pro-activeness, understanding of an agent communication language [2] [3]. They may also possess

Copyright 2017.All rights reserved.

features such as mobility, adaptability and collaboration. Multi-agent system is a system which consists of multiple agents interacting with each other to learn or exchange experience [4] [8]. Consequently more complete operational picture of the cyber space can be provided by these multi-agent tools.

F. Algorithms: -

An algorithm can be identified as a procedure or a formula which helps in solving a problem. A computer program can be viewed as an implementation of an algorithm. In mathematics and computer science, an algorithm usually means a procedure that helps to solve a recurrent problem. New approaches can be made by combining set of algorithms in order to detect and defeat cyber-attacks [5] [9]. Combining Fuzzy logic and Genetic Algorithm (GA) for identify intrusions has being developed since there is an essentiality of a high security approach to safe and confident communication of information between different organizations [7]. In creating new approaches FUZZY LOGIC algorithm and GENETIC algorithm are being used. Genetic Algorithm is an optimization algorithm that helps in finding appropriate fuzzy rules. Fuzzy rule is a machine learning algorithm. Fuzzy logic along with genetic based approach gives more powerful performance.

G. Data sharing between agents: -

Agents share its data with other agents in the system. In sharing data, the system has used wide varieties of sharing schemes such as, centralized data reporting on one side and decentralized sharing on the other. This article present a theoretical concept and framework based on peer-to-peer computing in order to integrate a multi-agent system. But this is sharing results in a scalability bottleneck due to the high volumes of incoming data; these systems often have slow performance or slow reaction.

H. Data mining: -

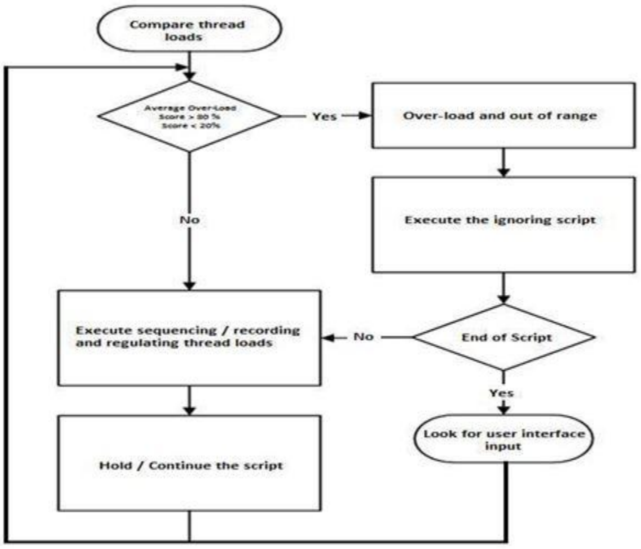
Data mining /data or knowledge discovery is the process of analyzing data from different perspectives and transforming it to useful information. It allows users to analyze data from many different dimensions, categorize it, and summarize the identified relationships. Typically, data mining is the process of identifying correlations or patterns among fields in large relational databases. Data mining concept can be used to analyze a multi-agent based approach in Intrusion detection. Analyzing previous cyber attacking details using data mining techniques predictions regarding the future attacks can be done.

III. ALGORYTHM PRINCIPLE

The analogy of SHOWAN (is Kurdish for shepherd) is introduced to describe a multitask initiative used to capture upon multiple concurrent threads. A functional representation of multitasking is the shepherd which has several tasks in a queue, and makes sure all of them are engaged to secure the flock against Each thread is designed with ability to be broken into smaller pieces in recursive periods. The goal is to use all the available processing power to enhance the performance of your application. The margin is an indication of CPU/Process load factor, which normalizes other threads' activity and prevents the result of reaching zero. It is selected to avoid 100% load as this is too high an index. However, for every time period that a high priority task passes the high/low margin, a penalty of a factor of 20% of the task's weight is deducted [2]. The choice of the penalty factor is valuable, as too low a factor could be flagged as it is at the zero indexes for a significant period of time. The optimal solution to the model is a sequence of high index threads (attacks) that has to be attended to or managed at every time period throughout the planned zone in order to maximize the objective function.

IV. THE METHOD

For this experiment, a sample data transfer process was selected under the DSDV routing protocol test without mention of a firewall. The exclusion of a firewall is due to the fact that an attack (Passive/Active) can be initiated by an entity inside the security perimeter. The core of this experiment is a program which implements the DSDV routing protocol. It has been used to transfer over a path between two nodes as well as handle changes in the routing table, including the link break. The program includes three threads, one for sending messages, one for receiving messages and one for main program, which creates other two threads. The sending thread will check the batch files and update the routing table of the node, transfer it into a piece of message and broadcast it. It will also increase the sequence number of this node every few seconds. The receiving thread will always wait for messages from other nodes, and update the routing table once such a message received. A lock mechanism is used in this program to make sure only one thread would update the routing table in one time. DSDV starts out high and then drops over the first 100 numbers, because the connections are starting up over the first 100 s, so the load on the network is constantly increasing over that time by automatic scripts, which check the transients and passages throughout the process. (Figure-1) We assumed that the route congestion is presumably not so severe. So that DSDV can track routes to all destinations and DSDV nodes are not dropping packets in the RTR level. The reason is that the route tuning practices to make a congested network run better and to bring the network slightly out of congestion collapse such as using shorter IFQs parameters is not the purpose of this experiment. Before start the simulation, we create four template TCL scripts to be used by our batch file to manually simulate attack scenarios. Batch file runs the simulation based on the test scenario varying speed and pause time. Batch files also copy the scenario based on the manually initiated threads, to run the AWK and recreate the LOG and finally trace and record batches will conclude the decision process to ignore or end the script. This whole process will give DSDV enough time to continuously update the entire routing table periodically and as per request, which creates a slight delay in delivery, but the end to end delay does not change to increase in the number of attacks and it only may increase number of hops.


Compare thread loads
Average Over-Load Score > 80 % Score < 20%
Yes
Over-load and out of range
Execute the ignoring script
No
Execute sequencing / recording and regulating thread loads
No
End of Script
Yes
Look for user interface input
Hold / Continue the script

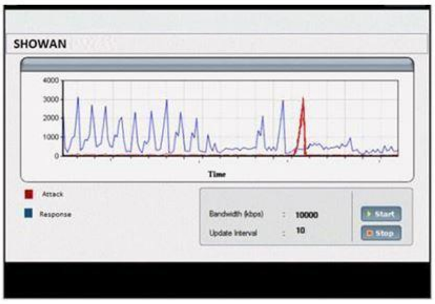
In addition, ten threads at different load and priority levels were initiated. [3] A low-fidelity multitasking software environment with graphical user interaction named SHOWAN was developed and used. In this environment, a user identified attack could be simulated and the behavior of the transfer process (task acceptance and ignorance rates from their satisfactory status) was measured. Task related factors such as task priority, penalty coefficient, and the duration of the experiment could be manipulated. We have used AWK scripting language and Java code for extracting data from trace files. The analysis is done for generating various performance metrics like packet delivery fraction, average end-to-end delay, packet loss, packet delay, routing overhead and route acquisition time.

V. THE SEQUENCE

The sequence of generating word placement and rhyming model starts from sampling and extracting data from each channel under attack, which creates pool of valid alternatives. This first step in sequence is to sample and expand data in the input prose with meaning-equivalent alternatives. The next phase is to measure the compared samples and collect, since different data strings have different meter constraints [20]. For example, a passive cyber-attack initiated by an entity inside security perimeter should follow a specific prose, which represents a natural flow of samples. An example cyber security initiated by an entity outside of security perimeter may follow iambic pentameter constraints, implying a rhythm that words establish in a line. This method introduces a combination of 10 weak/strong sample pairs per line which creates a string under the required constraints using choices from the first phase. Providing the pre-calculated measure from previously recorded cyber-attacks (Original-attack) simplifies the look-up task, in which, we collect rhyme information of each sample in the original-attack or expanded sample set. The junction of the rhyming sets across different data sets is taken to find rhyming pairs of samples. Using the expanded word set greatly improves our chances of finding valid matching rhyme pairs. This experiment annotates the risk categories according to scores. There are 5 score classifiers in the project corresponding to:

- Threads passing higher margin scores that might contain high risk
- Threads passing higher margin scores that sure contain high risk
- Threads that stay in safe operation area
- Threads passing lower margin scores that sure contain high risk
- Threads passing lower margin scores that might contain high risk

Rhyming and placement method requires to use known category along with the others, otherwise AI script classifier would be able to identify only scores with every given threads. But in real world there are other types of threads that do not fall into either of the above presented categories, so it is required to meddle with AI classifier thresholds. The graphical software user interface mimics threads load and priority indexes, which were shown as graphical monitors. (Figure-2)

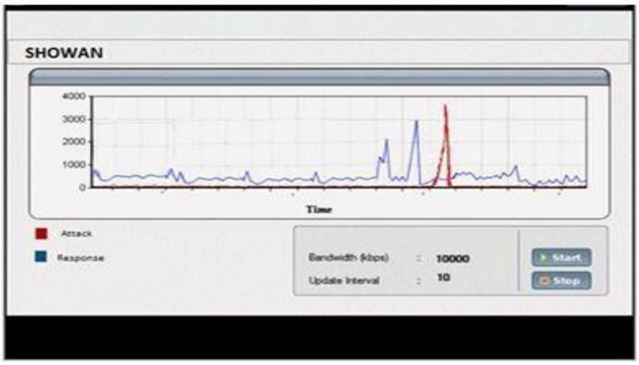


Bandwidth (kbps) : 10000
Update Interval : 10

Figure 2: Initial Reaction to a High Risk Activity Captured By SHOWAN

As the process runs, the load index indicators drop, demonstrating how the system resource usage becomes normalized over time. The software updated the status of the threads every tenth of a second. Therefore, the process could, theoretically, react to a sudden load unbalancing of any We also assume only one routing protocol is necessary for confirming no more than one connection to the event. Furthermore, the software calculates changes in the measures of an unsuccessful attack as the duration of the attack minimizes [1]. The software computes a numerical score for each thread, depending on how normal the process can keep the average status of the threads over time, how long threads were indexed in the 'out of the range zone', and whether or not the thread with a higher index value crashed.

Figure: Normalized Response to a High Risk Activity Captured By SHOWAN



Bandwidth (kbps) : 10000
Update Interval : 10

The software recorded which thread was more affected at any point in time. The process behavior in response to the simulated attack was then replayed on the computer and studied in order to understand the process stream line around denial of service (DOS) or R2L [10]. This helped to discover how the process handled the surprises against familiar threads. Figure 3 shows a sample of the graphs generated.

VI. EXPERIMENTAL PROCEDURE

The procedure involved running the software multiple times and under different scenarios in order for the examinee process cycles to learn and capture a normal range of scores within nominated strategies. It does not model all the details and characteristics of an attack but obtains simple models where multiple attack scenarios can be observed. The scenarios differ from each other in terms of index and weighted value. The length of the test for each scenario is ten to fifteen minutes, during which the graph for each process/thread to scenarios is recorded. In all cycles, scenarios started with an initial index of 20%, and penalties of 30% of the index value were deducted for every time that they neared upper and higher limits. The primary intent of the scenarios was to discover how the cognitive act of the computer’s process can be improved by learning the normal process stream line while effectively ignoring the irregular activities. Study of the process behavior also indicates how close to failure a complex process reached when the complexity of the attack increased.

VII. CYBER-AWARENESS IMPROVEMENT USING ARTIFICIAL INTELLIGENCE TECHNIQUES

We used the random waypoint model where each thread starts the transmission simulation by remaining stationary for minimum pause times. It then selects a random destination and moves toward that, assuming at maximum speed. Upon reaching the destination node, the thread pauses again, selects another destination, and proceeds there as previously explained, repeating this behavior for the duration of the simulation. It has been considered to use the advantage of multicore processing, in order to speed-up of the script execution, but using multiple processors in parallel computing is limited by the time needed for the sequential fraction of the program.

To really take advantage of a multi-core system, the script should split up the main processing into multiple threads as well and requires an algorithm, capable to be parallelized. This is only possible to a point however, depending on the algorithm. A simple explanation would be:

For a

= + + +

To be parallelized in a most simple way (Would be more complex under a CPU thread management system):

=+

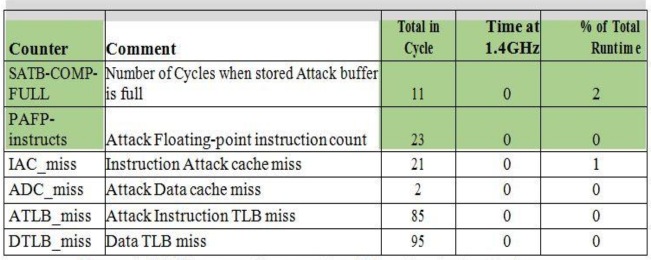
And:

= +

So:

= +

0 and 1 could be calculated in parallel threads. However, to calculate X, we need the results of both threads. So most part of the algorithm execution must be in parallel, but another part is implicitly sequential. It depends on results from earlier part of calculations, so there is no way to run this calculation in parallel with other dependent calculations. The peak performance of the processor can be calculated by multiplying the frequency by the number of cores and the number of instructions that can be issued for each core. Assuming that the single core processors runs at 1.4GHz, then the processor can sustain 1.4*1 = 1.4 billion instructions per second. Because multiple threads are sharing a single core, the question of whether the single core is fully loaded or not becomes interesting. For example, suppose that the core is fully loaded. That means each thread should be getting its fair share of the available instruction slots. So each thread should be able to issue an instruction every few cycles. In this case it becomes interesting to ask whether running fewer threads on the core would improve the latency of the application. Or while CPU is fully loaded, were the new attacks would be properly handled. Alternatively, if not all the threads on a particular core are busy, it is interesting to ask whether the core has sufficient instruction issue capacity to handle additional threads. In this experiment, assumption has been made that utilization of single core CPU is already optimized and manual experimental attacks are not to determine whether fewer number of threads would help CPU performance to benefit from fewer or more virtual processors being assigned work. We have also estimated the CPU resource consumption as following sample, in order to be used during cyber-attacks analysis, (Figure-4):



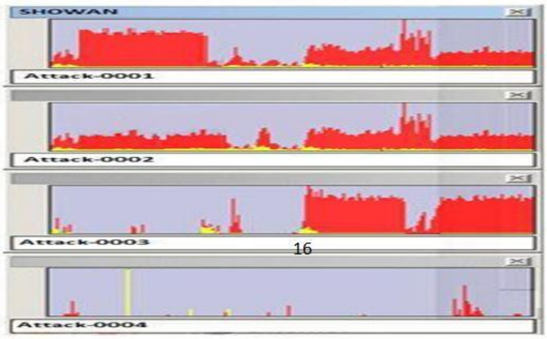
Counter	Comment	Total in Cycle	Time at 1.4GHz	% of Total Runtim e
SATB-COMP-FULL	Number of Cycles when stored Attack buffer is full	11	0	2
PAFP-instructs	Attack Floating-point instruction count	23	0	0
IAC_miss	Instruction Attack cache miss	21	0	1
ADC_miss	Attack Data cache miss	2	0	0
ATLB_miss	Attack Instruction TLB miss	85	0	0
DTLB_miss	Data TLB miss	95	0	0

Figure 4: CPU Resource Consumption Estimation during Cycles

VIII. THE RESULTS AND CONCLUSION

The results of the experiments were compared through graphical indicators for two types of attack (Locked Sequence and Synchronized), during qualitative comparison. It indicates the best, worst, and mean performance reaction of the examined thread. [6]

For quantitative comparison, the process overreacted to the high index and high load threads in early implementation phases. It was not able to disengage the synchronized threads and attempted to handle too many, causing poor performance and out of margin penalty. This effectively reduced switching time to zero. Even with manual manipulation assistance, none of the threads were able to beat the high margin. After a number of successful attempts, the process has improved and learned the normal operation load/priority indexing routine. As a result of the learning relaxation, the maximum best score range was scaled within 56% to 85% of the near-optimal for all different attacks. The overreaction caused the process to ignore threads with least priority index or the lower value threads had a much better performance [14]. During the process, some standard rhyming detection method has been called, while looking up the manually developed multitasking attacks. The purpose is to increase machine learning capabilities and help computer system administrators to learn the attack symptoms from machine-learning prospective. The methods, presented with an interface showing the original samples and manipulating elements. An algorithm dynamically compares the composed manipulating elements and finally let the administrator personnel to provide some alternative combination and understand the safe operating margins. For the process to learn good behavior, selection, and the number of threads to which, it must react is much more important than the specific thread's moment-to moment index changes



Idea behind this experiment is that, while two threads performing fundamentally similar operations on separate logical processors, the equal load observation will likely see little performance difference and gain lose. Technically for a multi-thread process to be a benefit; the two threads coexisting on a physical CPU must perform a variety of operations to allow the processor to make better use of latency and other similar factors. The test bed used in this experiment is by no means to benchmark processor architecture or design. The goal is to stress the processor, using different processor resources, and try to gain detail insights into the effects of unbalanced and non-rhythmic multi-processing. De-noising is also an important part of post result processing. For example several small pieces of a captured result is manipulated and stained by noise points and direct de-noising may leave scratch and traces. Using over-scaling to 350% normalizes and transforms the characteristics of the signal as it appears in figure -6. Every round of optimization, results decompose and expand into more sub-sections by SHOWAN, of which corresponds to a smoothed version, and the other corresponds to details versions.

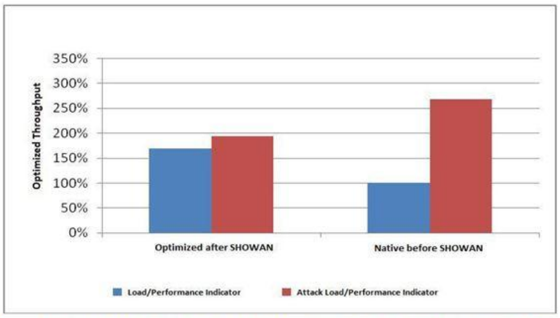


Figure: Comparison of Optimized vs. Non-Optimized Load/Performance Metrics

While the process is on idle, SHOWAN still runs in the background managing and analyzing the major initiatives taken by a process against higher index tasks on behalf of process threads, involving resources and performance in internal and external environments in previous scenarios. This idle processing improves the process learning curve. Once the thread is flagged by the process, it will handle much faster and efficiently in subsequent attempts. This means, strategic task management plays a major role and is more important than tactical task management. This experimental study is not similar to performance monitor counters, which are basically hardware registers that measure events occurring in the processor. They normally can be used to help find performance bottlenecks by identifying an excessive amount of events of a particular type. For example, a high number of conditional branch instructions may indicate a section of logic that, if rearranged, might lower the number of branches required. Even though performance counter can bring these issues to light, it is up to user to match them to application code and decide how they will help you improve application performance. The present experiment provides a graphical representation of each thread's state at a particular time in the run time. Each thread state is color-coded to help identifying what each thread is doing in conjunction with all of the others. Threads that go through multiple state changes are easily identifiable through the changing colors in the track pane. Figure 10-2 shows fifteen threads being tracked. A cyber-attack assumes to start as a multi-thread or hyper-thread operation, which may synchronously activates some processor's thread and do not duplicate all available resources.

REFERENCES

[1] International Journal of Scientific and Research Publications, Volume 6, Issue 4, April 2016.

[2] 2011 3rd International Conference on Cyber Conflict C. Czosseck, E. Tyugu, T. Wingfield (Eds.)Tallinn, Estonia, 2011 © CCD COE Publications.

[3] Cyber security and Artificial Intelligence from Fixing the Plumbing to Smart Water.

[4] Enn Tyugu, "Artificial Intelligence in Cyber Defense", 3rd International Conference on Cyber Conflict, Tallinn, 2011. Development, Concepts and Doctrine Centre (DCDC), UK Ministry of Defence,

[5] Strategic Trends Programmer: Global Strategic Trends – Out to 2040, 4th ed., January 2010.

[6] Defense Advanced Research Projects Agency (DARPA), United States, "DARPA Urban Challenge",http://archive.darpa.mil/grandchallenge/ , November 2007.

[7] Alessandro Guarino, "Autonomous cyber weapons no longer science-fiction",Engineering and Technology Magazine, Vol 8 Issue 8, http://eandt.theiet.org/magazine/2013/08/intelligent-weapons-are-coming.cfm, 12 August 2013.