International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 3, Issue 4, July-August 2014 **ISSN 2278-6856**

MALWARE DETECTION APPROACH BASED ON ADVANCED CRYPTOGRAPHIC PROVENANCE VERIFICATION

Anirudha B. Vikhe [1] and Prema S. Desai [2]

[1]Pune University, SKNCOE,Pune

[2]Pune University, SKNCOE,Pune

Abstract

In this paper, the problem of fake network calls on outgoing data packets, initiated by malware with intent to disrupt regular system operations and affect the sensitive as well crucial information residing on host is tackle. The provenance of such data is achieved with new advanced cryptographic approach which provides assurance to OS and helps for identifying the malware attacks. The goal of this new technique is to maintained data integrity and improve the trustiness of such data in order achieve guaranteed assurance of correct origin of system data. Based on the advance verification technique used, new security property has been avail for improving data surety and assurance. For implementing new security model, two special cryptographic modules, sign and verify are employed for ensuring the correct origin or source of system data and prevent adversaries from tampering with intent to threat its integrity. The hardware component, trusted platform modules helps to improve security level by providing attestation for cryptographic keys and storage for critical information. With the utilization of signing and symmetric at both modules the secure verification process put forth. Sign module is responsible for signature generation of input data along with UMAC and unique signing key. Verification process takes at verify module for the data under consideration. The signature task is enhanced with use of advanced cryptographic algorithm which much more effective and fast as compare to AES. The implantation result show that ACA is secure and fast for provenance verification approach.

Keywords: Authentication, cryptography, forgery, integrity, keystroke events, malware detection, provenance, trusted computing.

1. INTRODUCTION

Millions of systems are affected by malware all over globe which disrupt the data of host. Specifically malware is computer contaminant and defined as program or software designed with intent to harm the user of system by affecting the sensitive information and gain access to system. The computer contaminants such as worms, backdoor, spyware, adware, virus creates numerous problems which make system operability problematic and hard for system user. Problems like system security failure, unauthorized access to user account, network slowdown, system shutdown, modifications to OS for hiding, loss of critical information, etc. The traditional approach for malware detection is completely based on signature scanning has been insufficient to prevent tampering of data against stealthy and secretive malware attacks. Most of the users are unaware for attack as malware such Trojan and rootkits hide their presence and make it difficult for detection system to catch them. Normally these stealthy contaminants reside on user's system and continuously interact with resources such as critical information and data present that particular infected host. The performance of system is slow down due to malware activities such as DOS attacks, command and control, spam, etc, by initiating unauthorized network calls by malware. These untrusted network calls make difficult for operating system to identify the original call initiated by authorize user. The objective is to distinguish such malware triggered network calls by improving the trustiness of data flow. To ensure the integrity of system and prevent adversaries, a new host based technique is utilized. The surety for origin or source of data is based on new property data provenance integrity which verify data against its source of origin and make it easy for OS to distinguish the suspicious network calls. The verification of user generated data is based on cryptographic provenance technique that utilizes advance cryptography algorithm and AES for provenance verification. The advanced provenance verification approach ensures trustiness of data and prevents malware from injecting fake key stokes and initiating unwanted network calls. The security of system is enhanced with use of hardware component trusted computing platform to make forgery difficult task. The challenge is to protect user generated data with use hardware based trusted chip TMP. The advance CPV enhances security of system against malicious traffic for maintaining data integrity. The advanced provenance verification approach focuses on ensuring correct origin of data along with the use of trusted computing platform at kernel level. The packets following regular path through system network stack are most common target for intruders. Usually malware traffic in inserted at network layer of host's network stack to corrupt legitimate traffic generated by user actions at application layer. Other problem is deactivation of transport layer by malware with intent to insert malware inputs. To tackle these problems at kernel level, two special cryptographic modules namely sign and verify acts as checkpoint against the data flowing along network layer of host. These checkpoints are responsible for maintaining data integrity by restricting data to flow through them. The

provenance of outgoing packets is attended with sign and verify module which improve data surety. The sign module acts as checkpoint at transport layer and is responsible for signature generation of packets arriving from user space. Verify module acts as checkpoint at network layer where data under goes verification process. Both the modules follow RSA key exchange for generating two special keys, signing key and symmetric key. Signing key works along with advanced cryptographic algorithm and generates signature for incoming data at transport layer. By providing encryption for the signature generated at sign module it is secretly shared with verify module. The task is accomplished with communication key for signature sharing. Verify module is responsible for verification of the received signature, which successfully detect malicious data. Hardware based Trusted computing platform provides prominent storage for cryptographic keys, sensitive and critical information, by restricting access to it with intent to improve root of trust. Security problems caused as result of intrusive activities is solved by TPM with hardware enhancement. TMP provides features such as protection for key material, system authentication, file sealing, storing changes for configuration in PCR, attestation, memory curtaining.

2. LITERATURE SURVEY

2.1 . Survey

Trusted platform module hardware chip was originated by efforts of Trusted Computing Group which comprises Intel, HP, IBM, Microsoft, Compact in 1999 with purpose to use hardware to enhance the level of trustiness and strengthen security of system or network. TPM acts as device with provision for credential storage, software integrity, secure storage of cryptographic information and device identity. Along with trusted software TPM utilize hardware that which offers resistance to malware attacks with enhance trusted infrastructure. With TPM authorization number of system security features includes data protection, chain of trust and device identity. It has pair of public/private key know as endorsement key which has been created at time manufacturing. The private key is internal to chip which not accessible to anyone other than manufacturer, hence cannot be read out. Endorsement key makes use of anonymous attestation protocol proves to be tamper resistant. Deian Stefan proposed a framework Telling hUman Bot Apart (TUBA) [4] for malware detection, this approach is based human malware differences with regard to characteristic behavior. The goal is to extract some features based on the human user and malware interaction to identify and detect bots. The system uses behavior characteristics such as user typing patterns, surfing patterns, click counts, etc. are used to distinguish true events that are generated by legitimate user. By identifying the difference between human user and automated bots, TUBA monitors the user's typing patterns prevents malware from injecting fake key strokes and verify its integrity. The TUBA client server architecture consists of trust agent sharing key secretly with trusted server, client generates keystrokes that are signed and attested by trust agent. Remote server verifies the signature for any fake events injected by malware. Deian Stefan proposes keystroke based malware detection technique [5] and use behavior feature from user for identifying infected system. The keystroke authentication based on TUBA was tested for forgery attacks against Gaussian bot and Noise bot. The two bot programs are capable of injecting fake key events on host. Keystroke based biometric authentication mechanism is used as user's typing pattern to distinguish human user from automated bots. TUBA protects integrity of host by detecting extrusion. Security of system is enhanced with use of TPM which proved to be secure environment and protect kernel integrity. The unknown key events are captured due to initial training stage during which the key events from trusted user are collected by server. For any distrustful event the system ask user to type the string and try to match the user typing pattern with result collected during training stage. If match the user is authorize else considered as unknown. Arati Baliga proposed effectual tool Gibraltar for detecting rootkits that are difficult to detect as they hide their presence at kernel level data structure. A tool Gibraltar [6] is used to identify the rootkits present at kernel level. It works on separate system monitors the target system. It works two phase training and enforcement phase. During training phase it enforces invariants on data structures of target system. Snapshot of data structure take by invariant generator are tested against the invariant. For any violation of invariants of target system is checked during enforcement phase. Integrity of kernel is at threat due presence rootkit which manipulate data and code, which is mostly violated with the hiding technique in order to avoid detection system. Arati baliga studied different kernel level attacks put forth new type of attacks based on [7] that affect operating system and can easily fool detection system. Various types of attacks such as wastage of resource, disabling pseudo number generator, disabling firewall of host, are have been described with impact on system functionality. The detailed study of new class of attacks that do not use any kind of hiding technique given by Arati baliga and also designed prototypes for some attacks where attacks have been categorized on the basis of hiding technique, modification to data both static and dynamic. New types of beak-in such as virus, spyware, logic bombs, Trojan horse, etc. propagates at faster rate and are difficult detect which affect system performance. With the automatic technique for detection newly emerged break-ins based on user intent is proposed by Weidong Cui. The malware detection technique BINDER [8] captures extrusions on the basis of following rules. User intent is based on three rules that checks for connection within same process, different process and as from parent. Violation of these rules is used for extrusion detection. BINDER architecture checks each process for started and finished time, by classifying the delay with each of process as new delay, old delay and previous delay. BINDER checks each process based on user intent and delay for connection. It is considered process that executed by malicious program are

standalone. Attacks such sending irrelevant emails, denial of service, click frauds where code runs to generate profit, etc, produce unwanted traffic on network. R. Gummadi proposes not a bot system [9] for identify and certify human generated traffic. Not a B comprises of at attester responsible for production attestations to user generated traffic by running at client system and verifier that is present for verification of attestation. The OS and applications are considered to be not trusted. On request attester sends attestations to verifier for verification. Verifier uses TPM for storage of signing key which ensures the secrecy. Working of attester to attest which request is based on three parameters. The best way is to ask user to attest or not by asking question that could be only answer by user. Second alternative is to observe the user key strokes and click patterns and on arrival of request use it as distinguishing feature between bots and user. Attester observes the keyboard and mouse for storing the user activity. Bots formed network know as botnets for attacking system and cause damage number of systems. Internet Relay Chat allows users to communicate with each other which are connected to server. IRC [10] is used by bots for command and control and allows attacker as means of communication with other bots which are part of botnet. J. Goebel used communication channel for detection of bots with technique n-grams analysis and scoring function. J. Goebel implemented a concept Rishi used to detect the IRC bots much efficiently. The communication channel used by attacker to control bots was used effectively Rishi.

2.2 . Motivation

There are number of problems that caused by malware by interacting with host and its resources. Malware attacks are growing at faster rate with devastating damage to system and the critical information present on host as traditional signature scanning approach proves to be insufficient. With the necessity to protect such critical and sensitive information against suspicious malware activities led to provenance verification of data. Also the demand for fast and secure approach for malware detection has led to use TPM that maintains data integrity constraints. Data generated at application layer by legitimate user activities must be kept unmodified and free malware fake keystroke injection. Cryptographic provenance approach ensure and prevents malware from initiating fake network call that challenge the system integrity, as well provides a secure storage for encryption key within the TMP. The need for trustiness of data is achieved with advanced cryptography that improves confidentiality of data which prevents unauthorized modification.

3. PROPOSED MODEL

3.1Model

The system comprises of three main modules input, sign, and verify module. The input module is responsible for accepting input from user which is passed to sign module for signature generation. Initially Sign and verify undergo key exchange and generation step for symmetric and communication keys. These key are used to prove the provenance of data and make system more secure against malware by providing verified data as output. Data considered to be suspected is handed by user to input module. At this stage there is no surety about incoming data at input module of system. This data is forwarded to sign module which has already undergone the key exchange and generation stage with verification module to produce signing and communication key. Sign module is responsible for signature generation and encrypts data with cryptographic algorithm. The decryption of data takes place at verify module which verify the data based on signature stored at hash table. The data packets which are not able to prove their provenance is easily captured at verify module. For secure and effective malware detection two special modules are used, sign and verify. Sign module is placed at transport layer and verify module at network layer so that packets from network would be authenticated for signature at the verify module. The outgoing packets normally flow through the system network stack. These packets flow from user space to kernel space. The traffic generated at user space flows in downward direction from application layer to physical layer. At network layer the packets have threat of tampering because most of malware attack took place at this point which try to avoid firewall of system. Proposed system helps to mitigate these types of attacks and improve the data integrity with secure approach.

- **Sign Module**

Sign module works in collaboration with transport layer by generating signature for all incoming packets from user space. These sign packets from sign module are sent to verify module for verification. For improving the security of outgoing packets advanced cryptographic algorithm used which provides encryption and decryption and two modules share cryptographic keys.

- **Verify Module**:

At verify module packets are checked based on their signature, if any packets are malware generated than those packets fail to prove the provenance and are detected for been malicious. With this approach malware cannot avoid the firewall system and improves the data assurance. In this technique three initial tasks are performed, Booting stage, Key generation stage, Verification stage.

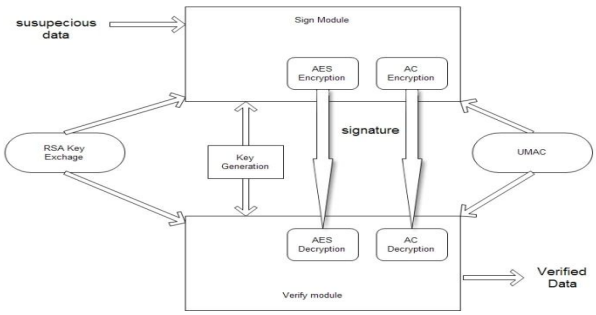

Fig. 1. Fig:-System architecture

The system will make use of trusted platform module and advanced cryptographic algorithm for security enhancements. The integrity of system is achieved by storing key within TMP chip. The proposed system provides protection for both application data and Kernel data. It also protects the outgoing packets from application layer to the layers beneath by maintain the integrity of such packets. Original data is protected with provenance verification by providing resistant against malware attacks. Integrity application data and kernel data is maintained where application data is result of user actions and kernel data is system generated data.

- **Major input to the System:**
  1) The input module is responsible for accepting input from user which is passed to sign module for signature generation.
  2)Signing key generated as resulted of XOR operation performed after exchanging four random numbers s0, s1, v0, v1, between sign module and verify module.
  3) Communication key generated as resulted of XOR operation performed after exchanging four random numbers s0, s1, v0, v1, between sign module and verify module.
- **Output from the System**:
  1) System generates alert to notify the user based on data given as input for verification. Following are the two types of alert generated:-
  2) If data inputted to system is coming from trusted source and is verified cryptographically.
  3) If data inputted to system malware generated or generated due to fake malware activities and is verified cryptographically.

3.2 Proposed Mathematics

❖ **Data Sets**

- Input:

IN= $\{in_i\}$ i Є [1, n]

- **Intermediate Results**

PK= $\{pk_i\}$ i Є [1, n]
PriK= $\{prik_i\}$ i Є [1, n]
SK= $\{sk_i\}$ i Є [1, n]
SIK= $\{sik_i\}$ i Є [1, n]

- **Output**

Alerts A; if data is affected
Then malicious data
Else, verified data

- Create Signature

Use UMAC algorithm to generate signature for each packet.
UMAC signature = HK(S) XOR F(nonce)
Where H = hash algorithm
K = Signing Key
S = Source
F = Pseudorandom number generator

- Signature Encryption with advance cryptography algorithm

CB1 = P (XOR) KB1
CB2= CB1 >>3
CB3 = CB2 (XOR) KB2
CB4 = CB3 (XOR) KB3

CB4 is the encrypted data

3.3 Create Signature Planning of Advanced CPV Model

Fig. 2 shows the overall idea of execution of Advanced CPV model. It shows the various operations that performed data source verification between sign module and verify module. Initially both of the modules undergo RSA key exchange by exchanging their public keys. Each of modules generates two random numbers which exchanged in encrypted manner. These four numbers are used to generate signing key and symmetric key by undergoing XOR operation. Following are steps for key generation described in detailed. The Sign module generates two random numbers a0 and a1, and encrypts a0 and a1 using the Verify module's public key sends them to verify. The Verify module receives and decrypts a0 and a1 with its private key. It then generates two random numbers b0 and b1. The Verify module encrypts b0 and b1 using the Sign module's public key. The Sign module decrypts them with its private key. Both the Sign and Verify modules have a0, a1, b0, and b1. They compute the signing key as a0 b0 and the symmetric key for their communication encryption as a1 b1 The signing key along with AES and ACA is used to generate signature for input data. Symmetric key is used for is used for data verification in verify module. The two algorithms use keys for encrypting the data and send securely to verify module for verification.

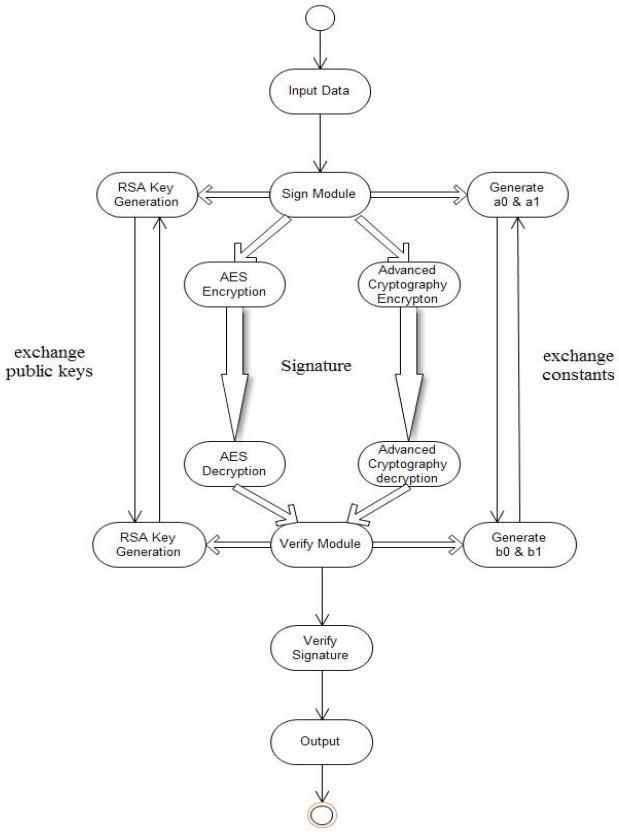

Fig. 2.Execution of the Model

3.4 Platform

The proposed system is implemented on the Windows 7 or windows 8. The platform for the implementation of this system is java.

4. Performance analysis

In order to verify the source of data cryptographically two algorithms have been used in data provenance verification i. e. Advanced Encryption Standard and Advanced Cryptographic Algorithm. Both of these algorithms have been successfully implemented at sign module for signature generation as well as verify module responsible for data verification process. The comparison of AES and ACA is done on basis of following parameters to detect malware attacks and presence of suspicious data. The result table fig 3 below shows comparison parameters in detailed.

TABLE1: SYMBOL TABLE

Symbol	Representation
In	Set of input from user
PK	Set of public keys
PriK	Set of private keys
SK	Symmetric key
SIK	Signing Key
P	Plain text
CB	Cipher block
KB	Key block
A	Set of alerts

presence of suspicious data. The result table fig 3 below shows comparison parameters in detailed.

Sr.no	Parameters for comparison	AES	ACA
1	Input file	146kb	146kb
2	Encryption time	69ms	4ms
3	Decryption time	360ms	290ms
4	Key size	16bytes	64bytes

Fig 3. Result table.

Input

The specifies the input module to browse particular file from host system considered for verification and pass it to sign module as well as verify for signature generation. In this scenario file of size 146 Kb is considered as input to system.

Observations

The factors considered for comparison are classified into two types:-Dynamic: It includes Encryption and decryption time which depend on particular file considered. Static: It focuses on key for AES and ACA which is 16 bytes and 64 bytes respectively. Following is the result obtained for given scenario:-

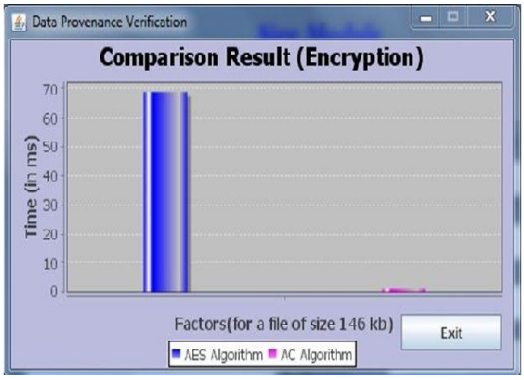

Fig 4. Encryption time

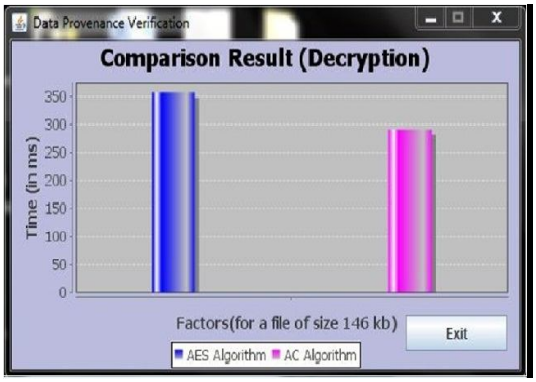

Fig. 5. Decryption time.

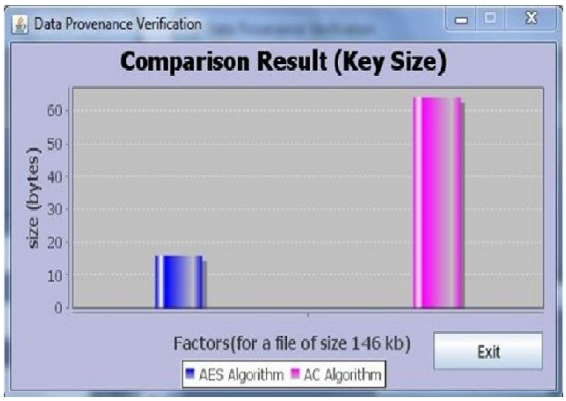

Fig 6. Key size

1) ACA requires considerably less time for encryption in comparison to AES for encrypting file of size 146kb.
2) For decrypting of same file with size of 146kb ACA requires than as in comparison with AES.
3) Key size for ACA is 64 bytes proves too hard for attacker to get combination as in comparison to 16 bytes used with AES.
4) The results generated are system generated graphs showing encryption and decryption times in milliseconds.

5. Conclusion

In this paper, the proposed idea of implementing Advanced Cryptography Algorithm in cryptographic provenance verification technique is used for host-based

malware detection. The effective advance CPV approach for verification of the data present on user system and provides protection computing resources of host. General approach for improving the assurance of system data and properties of a host is presented, which is used to prevent and identify malware activities. CPV's application in identifying stealthy malware activities of a host, in particular how to distinguish malicious/unauthorized data flow from legitimate one on a computer that may be compromised. Advanced CVP approach gives satisfactory guarantee of data with trusted computing platform which includes the most trustiness in the provenance of data integrity, confidentiality and availability. The security of systemis enhanced with the use of advance cryptographic method which provides encryption and decryption for data. This Advanced CCP model can be enhanced to work on the provenance verification formed by combinations advance cryptography.

Acknowledgement

For proposing this model referred the IEEE Transaction paper under the title "Mining Discriminative Patterns for Classifying Trajectories on Road Networks", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 5, MAY 2011. Also, the preliminary version of this model is accepted and published by International Journal of Computer Applications 81(13):14-16, November 2013 under the title "Classifying Trajectories on Road Network using Neural Network".

References

[1.] Deepak S Gaikwad and Usha A Jogalekar. Article: Classifying Trajectories on Road Network using Neural Network. International Journal of Computer Applications 81(13):14-16, November 2013. Published by Foundation of Computer Science, New York, USA.

[2.] Anirudha Vikhe, P. S. Desai, "Data Provenance Verification for Secure Host Using Advanced Cryptographic Algorithm", INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS, VOL 88-NO.11, FEB. 2014

[3.] Huijun Xiong, Chehai Wu, Deian Stefan, Danfeng Yao Member, Data-Provenance Verification For Secure Hosts , IEEE Transactions On Dependable and secure computing vol.9 no.2 year 2012.

[4.] Vishwagupta, Gajendra Singh, Ravindra Gupta Advance cryptography algorithm for improving data security, International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)Volume 2, Issue 1, January 2012.

[5.] D. Stefan, C. Wu, D. Yao and G. XU. Ensuring host integrity with cryptographic provenance verification. In CCSS 09, poster, November 10-12, Chicago, IL, USA, 2009.

A.D. Stefan and D. Yao. Keystroke dynamics authentication against synthetic forgeries. In Proceedings of the International Conference on Collaborative Computing: Networking, Applications and Work sharing (Collaborate Com), November 2010D.Stefan, C. Wu, D. Yao and G. XU. A Cryptographic Provenance Verification Approach for Host-Based Malware Detection.

[7] A.Baliga, V. Ganapathy, and L. Iftode. Automatic inference and enforcement of kernel data structure invariants. In 24th Annual Computer Security Applications Conference (ACSAC) 2008.

[8] B.Baliga, P. Kamat, and L. Iftode. Lurking in the shadows: Identifying systemic threats to kernel data. In IEEE Symposium on Security and Privacy, pages 246–251. IEEE Computer Society, 2007.

[9] W. Cui, R. H. Katz, and W. Tian Tan. Design and implementation of an extrusion-based break-in detector for personal computers. In ACSAC, pages 361–370. IEEE Computer Society, 2005.

[10] R. Gummadi, H. Balakrishnan, P. Maniatis, and S. Ratnasamy. Not-a-Bot: Improving service availability in the face of botnet attacks. In Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NDSI), 2009.

[11] J. Goebel and T. Holz. Rishi: Identify bot contaminated hosts by IRC nickname evaluation. In Proceedings of the First USENIX Workshop on Hot Topics in Understanding Botnets, April 2007.

[12] S. W. Smith. Trusted Computing Platforms: Design and Applications. New York: Springer, 2005.

[13] Schneier and N. Ferguson. Practical cryptography, 2003.

[14] J. M. McCune, A. Perrig, and M. K. Reiter. Safe passage for passwords and other sensitive data. In NDSS. The Internet Society, 2009.

[15] M. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. My botnet is bigger than yours (maybe, better than yours). In Proceedings of the First USENIX Workshop on Hot Topics in Understanding Botnets, April2007.

AUTHOR

Anirudha Vikhe is a PG Fellow of 2012 batch in Computer Network of Smt. Kashibai Navale College of Engineering, Vadgaon (Bk), Pune-41(MH), India. This author received Bachelor of Engineering degree in Information Technology from Smt. Kashibai Navale College of Engineering, Vadgaon (Bk), Pune-41(MH). He is currently doing stipendiary internship at the Smt. Kashibai Navale College of Engineering, Vadgaon (Bk), Pune-41. His research interest includes data security and integrity. He has one journal and a proceeding conference paper on his research work of PG project as "Data Provenance Verification for Secure Host Using Advanced Cryptographic Algorithm", INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS, VOL 88-NO.11, FEB. 2014.

Prema Desai is an Asst. Professor in Smt. Kashibai Navale College of Engineering, Vadgaon (Bk), Pune-41(MH), India