International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 4, 2014. ISSN: 2348-6600.
Reference ID: IJCS-048. PAGE NO: 273-277.

Effective Mechanism for Blocking DNS Based Misbehaviours and Web Content Modification

K.Priyadharshini
ME/CSE ANNA UNIVERITY
Bharathiyar Institute of Engineering for women
Deviyakurichi, Attur(dt), salem.
kdharsini@live.com

Abstract— **Domain Name System (DNS) queries for botnet command and control provides a distributed infrastructure for storing, updating, and disseminating data that conveniently fits the need for a large-scale command and control system. The HTTP protocol is for the end-to-end communication between a client and a server. DNS provides not only a means of communication between computers, but also systematic mechanisms for naming, locating, distributing, and caching resources without tolerance. These features of DNS may be utilized to fullfill more effective command-and-control system than what HTTP servers may provide. The DNS server then responds with the appropriate data using the agreed upon semantics. We identify several groups of features that allow Disclosure to reliably distinguish C&C channels from benign traffic using Net Flow records to reduce Disclosure's false positive rate, we incorporate a number of external reputation scores into our system's detection procedure. We provide an extensive evaluation of Disclosure over two large, real-world networks. Our evaluation demonstrates that Disclosure is able to perform real-time detection of botnet C&C channels over datasets on the order of billions of flows per day. The DNS server is one of the primary and most vulnerable infrastructure components through which communications service providers suffer Denial of Service and Distributed Denial of Service attacks. Attackers, in particular botnet controllers, use stealthy messaging systems to set up large-scale command and control for web content defacing.**

***Index Terms*— Network security, DNS security, botnet detection, and command and control**

I. INTRODUCTION

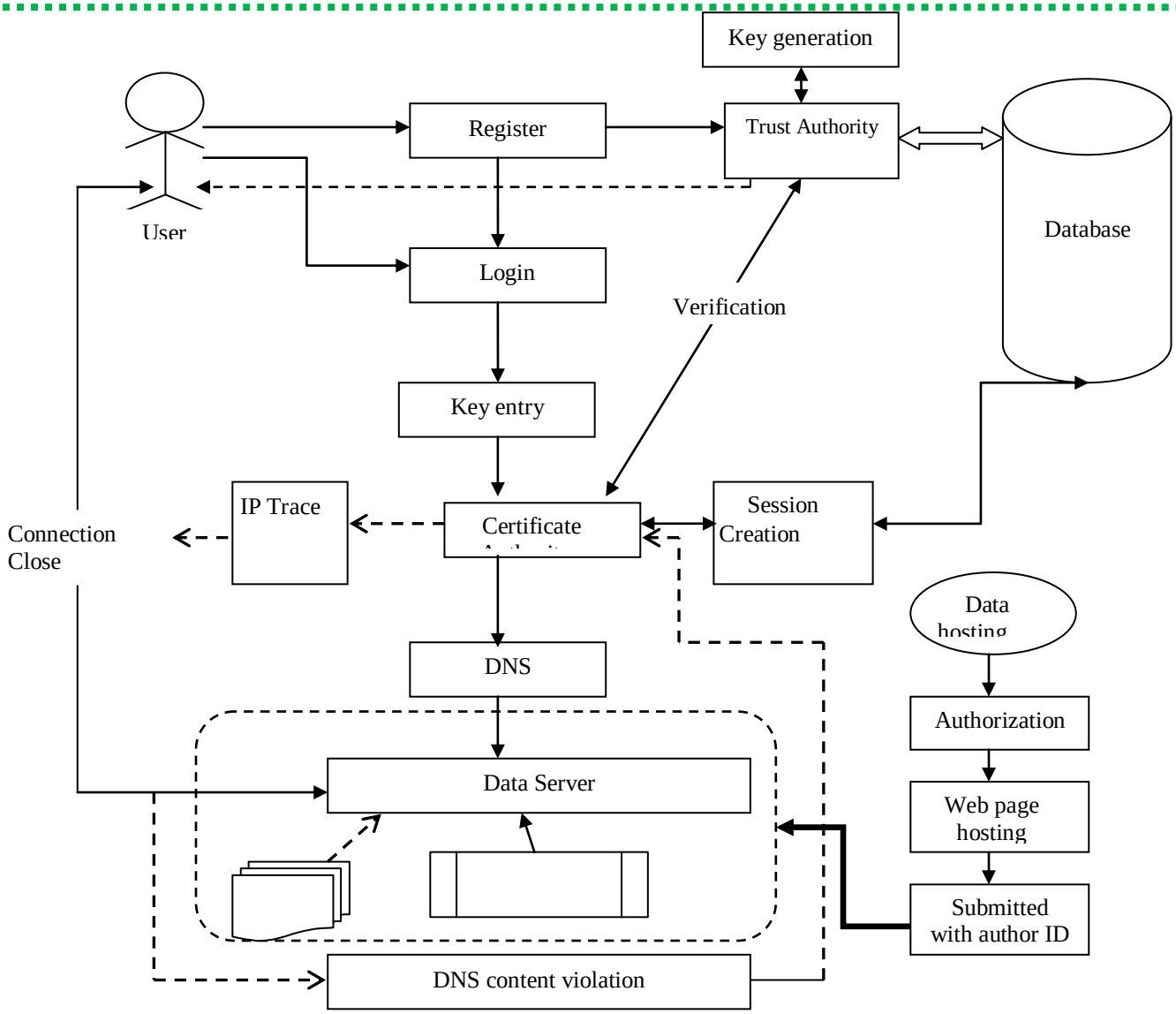
BOTNET command-and-control (C&C) channel refers to the protocol used by bots and botmaster (i.e., botnet controller) to communicate to each other, e.g., for bots to receive new attack commands and updates from botmaster to submit stolen data. Botnet operators constantly explore new stealthy communication mechanisms to evade detection. HTTP-based command and control is difficult to distinguish from legitimate web traffic. we systematically investigate the feasibility of solely using Domain Name System (DNS) queries for botnet command and control. DNS provides a distributed infrastructure for storing, updating, and disseminating data that conveniently fits the need for a large-scale command and control system. The HTTP protocol is for the end-to-end communication between a client and a server. The HTTP protocol is for the end-to-end communication between a client and a server. In comparison, DNS provides not only a means of communication between computers, but also systematic mechanisms for naming, locating, distributing, and caching resources with fault tolerance. These features of DNS may be utilized to fullfill a more effective command-and-control system than what HTTP servers may provide. These features of DNS may be utilized to fullfill a more effective command-and-control system than what HTTP servers may provide. The DNS server then responds with the appropriate data using the agreed upon semantics. To reduce Disclosure's false positive rate, we incorporate a number of external reputation scores into our system's detection procedure. Finally, we provide an extensive evaluation of Disclosure over two large, real-world networks. Our evaluation demonstrates that Disclosure is able to perform real-time detection of botnet C&C channels over datasets on the order of billions of flows per day.

Over the years the internet has expanded to enormous proportions, with increasing numbers of hosts and availability of high-speed connections. This expansion has also lead to an increase in the number of attacks on hosts.

The Domain Name System (DNS) is one of the important parts of the internet. Without it most people would not be able to connect to their favorite website. It is not hard to imagine that DNS servers have also been the targets of attacks. These attacks are possible because of exploits in the DNS protocol or bugs in the DNS software. There is also a group of attacks that overload a host with packets taking up massive amount of bandwidth and processing power in the hope of making the DNS server unavailable for genuine users.

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 4, 2014. ISSN: 2348-6600.
Reference ID: IJCS-048. PAGE NO: 273-277.



These attacks have recently become more diverse, as attention has shifted from attacking the front end to exploiting vulnerabilities of the web applications in order to corrupt the back-end database system. A plethora of Intrusion Detection Systems (IDSs) currently examine network packets individually within both the webserver and the database system. However, there is very little work being performed on multitiered Anomaly Detection (AD) systems that generate models of network behavior for both web and database network interactions.

In such multitierd architectures, the back-end database server is often protected behind a firewall while the webservers are remotely accessible over the Internet. Unfortunately, though they are protected from direct remote

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 4, 2014. ISSN: 2348-6600.
Reference ID: IJCS-048. PAGE NO: 273-277.

attacks, the back-end systems are susceptible to attacks that use web requests as a means to exploit the back end. To protect multitiered web services, Intrusion detection systems have been widely used to detect known attacks by matching misused traffic patterns or signatures.

Botnet command-and-control (C&C) channel refers to the protocol used by bots and botmaster to communicate to each other, e.g., for bots to receive new attack commands and updates from botmaster, or to submit stolen data.

A C&C channel for a botnet needs to be reliable, redundant, noncentralized, and easily disguised as legitimate traffic. Many botnet operators used the Internet Relay Chat protocol (IRC) or HTTP servers to pass information. Botnet operators constantly explore new stealthy communication mechanisms to evade detection. HTTP-based command and control is difficult to distinguish from legitimate web traffic. Web-Delivered services and applications have increased in both popularity and complexity over the past few years.

Daily tasks, such as banking, trav el, and social networking, are all done via the web. Such services typically employ a web server front end that runs the application user interface logic, as well as a back-end server that consists of a database or file server. Due to their ubiquitous use for personal and/or corporate data, web services have always been the target of attacks.

A class of IDS that leverages machine learning can also detect unknown attacks by identifying abnormal network traffic that deviates from the so-called "normal" behavior previously profiled during the IDS training phase. Individually, the web IDS and the database IDS can detect abnormal network traffic sent to either of them. However, we found that these IDSs cannot detect cases wherein normal traffic is used to attack the web server and the database server.

If an attacker with non admin privileges can log in to a web server using normal-user access credentials, he/she can find a way to issue a privileged database query by exploiting vulnerabilities in the web server. Neither the web IDS nor the database IDS would detect this type of attack since the web IDS would merely see typical user login traffic and the database IDS would see only the normal traffic of a privileged user.

II. EXISTING SYSTEM

Decentralized nature of DNS with a series of redundant servers potentially provides an effective channel for covert communication of a large distributed system, including botnets. To protect DNS based attack, Intrusion detection systems have been widely used to detect known attacks by matching misused traffic patterns or signatures. A class of IDS that leverages machine learning can also detect unknown attacks by identifying abnormal network traffic that deviates from the so-called "normal" behavior previously profiled during the IDS training phase. Individually, the web IDS and the database IDS can detect abnormal network traffic sent to either of them. However, we found that these IDS cannot detect cases wherein normal traffic is used to attack the webserver and the database server.

A. Disadvantages of Existing System:

- Traditional Intrusion Detection System(IDS) are not much familiar with the Multitier web application attacks not in case of DNS attacks.
- Fire wall protection is the security in most web applications and has following limitations.
- Those Systems concentrate separately by either in Front End or Back End Server attacks.
- Attacks are detected after the data corruption will occurs
- Server (Back End) data will get modifications frequently
- Normal users also affected due to modification of web content
- Data Corruption and Denial Of Service(DOS)
- Not much effective in providing security

III. PROPOSED SYSTEM

B. User Authentication

This is the basic module for every client communication by an ANONYMOUS Network. The user (Client) should register them to Trust Authority before communicate with DNS server. During the registration process, Trust Authority can directly extract information like IP, HOST details of the Client. The IP and Host information helps during the verification of Certificate Authority.

C. Key Generation And Verification

Trusted Authority generates a Random Key for every new client and stores the value with proper HOST and IP details respectively. Once the key generation successfully occurred, client can communicate the Certificate Authority for further process. Next step is the process of verifying key which is deposit to the Certificate Authority by client. In this verification process Authorities communicate each other and

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 4, 2014. ISSN: 2348-6600.
Reference ID: IJCS-048. PAGE NO: 273-277.

compare keys, IP and HOST details. If any negative value occurs then communication will be terminate.

D. Session Passing With Content Violation Check

After the verification process, client key will renovate into Session Value by using Blind Signature Method. Session nothing but state information maintained for every connection with client. All session data stored on database of server. This method will help to transmit signature based session for server communication. Before the session generation, Certificate Authority checks with Content Violation Table. This table store blocked client sessions. This process helps to block users who were misbehaved in early session.

E. Process Controller

This module is the controller of the entire process. There are two ADMIN in this work

- Trust Authority
- Certificate Authority

These two are the initial Security Level Authorities (ADMIN), who are decides to whether the user should communicate with server or not. One will make the first level of security and can enter, update and checks the detail of users like IP, HOST, name etc... And another one checks the details of user key, verifies them with Blacklist and stores the IP. The server is like the web server which will gives the valid response to the user requests respectively and marks the users into blacklist if they misbehaved.

F. Reports

This is the final module which gives the detail reports about the users to both Trust and Certificate Authority and also for the Server too. Some of the reports are

- Blacklisted users
- Normal users
- Total users
- IP Address

1) Blacklist IP Address

IP stands for "Internet Protocol," which assigns a numeric address to every computer and router on the Internet. These addresses are attached to every Internet service request, such as for a Web page from a Web server, or to connect to other Internet resources.

IP address blacklisting is a method of protecting Web and other Internet servers from malicious attacks. This is accomplished by setting rules within server software or hardware routers regarding what traffic will be considered an attack, and then preventing the computers creating that traffic from connecting

IV.PERFORMANCE EVALUTION

Piggybacking case studies. We select four hosts from our data set to simulate the piggybacking behaviour rs on them and evaluate the mean TTC. The four hosts are the first, 50th, 100th, and 200th most active hosts according to their total traffic volume during the 2-month period. Fig. 3 plots how the mean TTC changes with the minimum TTC in a piggybacking query strategy, with the maximum TTC set to infinity. The results show that bot's communication efficiency is higher on more active hosts as expected. A maximum TTC (e.g., 48 hours) may be set to ensure periodic communication of the bot. Mean TTC grows with minimum TTC and is almost always greater than minimum TTC.

V.CONCLUSION

Multi safeguard Mechanism (MSM), an IDS system that models the network behavior of user sessions through the network level, to resolve DNS based websites defacing. This system helps servers can control misbehaving users, thereby blocking users without compromising their anonymity. To hunt out attacks those independent Intrusion Detection Systems would not be able to identify.

Prevent the web content attackers of DNS Trace and Blocks Misbehaving users. Multi safeguard Mechanism (MSM) used to provide security against Website Defacing, performed by user who works in Anonymous Network. It can be extended to any Network Infrastructure which aims to provide flexibility.

REFERENCES

[1] Andreas Wespi, Herve Debar and Marc Dacier and (1999) "Towards a taxonomy of intrusion-detection systems" ,Computer Networks 805–822.

[2] Angelos Stavrou, AnupK.Ghosh,SushilJajodiaandYih Huang,(2008)"Efficiently Tracking Application Interactions using Lightweight Virtualization"ACM 978-1-60558-298-6/08/10.

[3] Balduzzi .M,Bilge.L, Kirda.E, and Kruegel.C (2011), "Exposure: Finding Malicious Domains Using Passive DNS Analysis," Proc. 18th Ann. Network and Distributed System Security Symp. (NDSS).

[4] Bos.H, Dietrich.C.J, Freiling.F.C, Rossow.C, Pohlmann.N and van Steen.M (2011), "On Botnets

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 4, 2014. ISSN: 2348-6600.
Reference ID: IJCS-048. PAGE NO: 273-277.

that Use DNS for Command and Control," Proc. European Conf. Computer Network Defense.

[5] Butler.P, Xu.K and Yao.D (2011),"Quantitatively Analyzing Stealthy Communication Channels," Proc. Ninth Int'l Conf. Applied Cryptography and Network Security (ACNS '11), pp. 238-254.

[6] Dagon.D, "Botnet Detection and Response, the Network Is the Infection (2005)," Proc. Domain Name System Operations Analysis and Research Center Workshop.

[7] Monrose.F, Provos.N, Rajab.M.A and Terzis.A (2008), "Peeking through the Cloud: DNS-Based Estimation and Its Applications," Proc. Sixth Int'l Conf. Applied Cryptography and Network Security (ACNS),

[8] Moskowitz.I.S, Newman.R.E, Syverson.P and Serjantov.A(2003), "Metrics for Traffic Analysis Prevention," Proc. Privacy Enhancing Technologies Workshop (PET '03), pp. 48-65.

[9] Nick Mathewson, Paul Syversonand Roger Dingle dine (2004) "Tor: The Second-Generation Onion Router",Proceedings of the13th USENIX Security Symposium.

[10] Shang.H and Willis.C.E (2006), "Piggybacking Related Domain Names to Improve DNS Performance, "Computer Networks,vol.50,no.11,pp. 1733-1748.

Published by SK Research Group of Companies (SKRGC).