ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 9, Issue 4, pp: (18-22), Month: October - December 2021, Available at: www.researchpublish.com

Towards modern authentication in a large-scale compute Installation

[1]Nabil Nabulsi, [2]Abdulaziz Hamidi

Saudi Aramco, Dhahran, Saudi Arabia

Abstract: **One of the many challenges in large cloud or compute installations specifically for High Performance Computing (HPC), is implementing a scalable, centrally managed authentication solution that supports simultaneous bursts of authentication requests or resolution. A modern solution like identity Policy and Audit (IPA), proven as a viable solution for traditional IT workloads, however for large compute installations it is more challenging. IPA is a cost-effective solution based on FreeIPA, providing a central repository for storing user identity, access policies, and auditing. Compared to a solution like Sun's Network Information Service (NIS), IPA has strong features in general but also contains some shortcomings such as complexity of configuration, administration, and recovery. This paper presents a case study for migrating to IPA solution for large computing installation of +12,0000 nodes deployed across two data centers. It uncovers some inherited limitations of out-of-the-box IPA deployment for HPC workloads and ways to work around these limitations to achieve an efficient, reliable, and scalable authentication solution.**

Keywords: **HPC, Scalable Authentication, IPA, Linux.**

I. INTRODUCTION

NIS Versus IPA

In general, Linux allows native administration of user credentials and details like user ID and group ID locally using tiny databases or plain files. Although this works well for individual machines, it becomes less desirable when maintaining a larger group of servers such as HPC, if each machine is supposed to be identical in behavior. The chance of having machines out of sync is linear as the number of machines increases in the environment. A centrally managed solution to manage user credentials is necessary to maintain IDs and details eliminates out-of-sync problem.

NIS or Network Information Service developed by Sun Microsystems [1] utilized delegation, like DNS (Domain Name System). It allowed for independent distributed nodes for authentication called NIS slaves. These slaves or replicas receive updates from one master through a push mechanism to stay in-sync. Slaves could also request a transfer of all data using pull to get in-sync on demand. Scalability is achieved through multi-tiering of slaves with an unknown maximum number of clients. Joining an NIS domain was easy, authorization is based on IP address where data and communication are not encrypted. Another limitation of NIS was a limit of 16 groups a user could be member off [2]. Some of these and limitations got improved on in NIS+ [3]. Managing data in general was achieved through either native Remote Procedure Call (RPC) commands like yppasswd and ypchsh or editing local files on the master and rendering or updating the binary maps (native format) [4]. Though it was not the most secure approach, it was easy to configure, administer, recover and finally easy to troubleshoot.

IPA is an Identity Management system (IdM) bundled by RedHat based on FreeIPA combining Lightweight Directory Access Protocol (LDAP), Kerberos, DNS, Private Key Infrastructure (PKI) [5]. Different to the NIS approach, IPA utilizes a meshed network for maximum availability and service continuity. It uses replication policies to ensure data integrity across IPA servers and to maintain access during link or connection outages. There is no single master as the case in NIS and as such all servers can push updates to the other replicas [6]. Since IPA is incorporating technology like LDAP which proven to handle an equal size of HPC nodes using less back-end servers as compared to one using NIS.

However, the meshed network approach would not allow for infinite scaling. An IPA server is limited to handle a maximum of four replica agreements [7]. With a total of 60 replicas in a single domain [8], the theoretical number of clients IPA could support is north of 120,000 [9]. Different to NIS, an HPC node requires authorization before joining an IPA Kerberos world or Realm, called enrolment where Kerberos plays a vital part. Besides a known history of Kerberos vulnerabilities and current ones [10] timely patching is of high importance and machine's date and time synchronized is of key importance [11] to successful user authentication. Host based or central certificates offer a trusted and encrypted host-to-host connection, for servers and clients. Since LDAP is used as the central repository storage, the core of the product, the data is stored in and retrieved from a database using schemas. Some schemas allow storage of binary data such as passport photos. All this offers flexibility but also raises the difficulty threshold in terms of system administration. Though some tools provide an easier way to manage data [12], it is common practice to use native IPA commands, tools, or applications for any LDAP data updates.

II. A CURIOUS CASE MIGRATING FROM NIS TO IPA

In large computing installation, a general rule of thumb dictates using at least two NIS slaves per cluster. For larger clusters, say 1000+ nodes it is often four NIS slaves. In the end, a ratio of 300 nodes to one NIS slave is established. How many nodes a single IPA server can support is not known and can only be discovered through trial and error. Our testing reveals that a ratio of 1,500 nodes per IPA server is reasonable assuming the service is available. Therefore, for large HPC deployment of 12,000 HPC nodes, eight IPA back-end servers can do the job easily. This leads to a substantial reduction in authentication servers compared to NIS from 40 to only 8. In addition, IPA inherit client-side caching making it more efficient. Each IPA server is configured as a multi-master with multi-replication agreements in a mesh topology. This allows for data integrity and accessibility even during single site isolation.

III. BUILDING IPA BACK-END AND MIGRATING THE DATA

When planning IPA back-end, there are three factors to consider than the number of servers to use. First and foremost, is it capable of handling large number of simultaneous authentication requests. Second, does it meet service availability and continuity requirement. Finally, and most importantly, is it scalable.

Building a cluster of dedicated IPA backend servers' specifically for HPC workload is one way of handling large number of simultaneous authentications. The dedicated backend servers provide segregation and optimization of network traffic between HPC clients and IPA servers. Other clients including servers and workstations traffic can use other IPA servers thus reducing the load and improving authentication overall times. In addition, with dedicated IPA servers, optimizing, debugging, and fine-tuning IPA services become much easier undertake.

Authentication service availability and business continuity can be address by deploying adequate number of IPA backend servers. It must be available during regular maintenance, upgrades, and patching activities. For HPC nodes of 12,000 nodes that is deployed in two sites, for example, IPA backend should be designed to handle all 12,000 clients from single site. This dictate using equal number of IPA servers in each site. The IPA meshed replication topology here become of vital importance. Building replication agreements carefully is key to ensure replication data flows is consistent and preventing split brain scenarios. It is always better to prevent split brain scenarios than dealing with expensive recovery of an IPA server. Note that replication interruptions are not usually an issue as data is synchronized seamlessly after IPA servers are brought back online.

Network bandwidth and open-network flow is at most importance for a scalable back-end design. Implementing standard of 10G network for example on all HPC IPA backend servers is recommended. High network and matching bandwidth avoid potential bottleneck and accelerate clients' load-balancing delivering same user's experience using different backend server and allow for better scalability.

As IPA back-end design is complete and ready, one task remains is to migrate NIS data to IPA. Open-source IPA-community scripts, developed for that purpose were helpful, though require some modification to make them comply to our environment. Object trees like “Service OU” were not defined in the default LDAP schemas and had to be manually created. Also, IPA existing naming conventions had to be revised to avoid conflict and to prevent potential problems. Think of underscores in hostnames as one example. In addition, information had to be extracted from the NIS maps and converted to ASCII. The ASCII data was imported into the IPA backend, or factually the LDAP database. To guarantee data consistency during transition a verification process had to be developed. Both NIS and IPA back-ends were being updated simultaneously throughout the migration process. Once HPC clients were migrated, NIS data would no longer need to be updated.

IV. CHALLENGES WHILE MIGRATING THE COMPUTE ENVIRONMENT.

Large HPC sites are comprised of hardware coming from various manufacturers, running custom or in-house built applications and as such dictate a mixed operating system configuration (OS) in heterogenous environment. This implies cluster likely have dissimilar OS levels and packages releases and more importantly patch levels. This becomes a challenge if using IPA native clients (SSSD) as it requires maintaining and distributing separate IPA packages repository with all the dependencies for HPC environment. Mixing packages from a more up to date OS with a current installed OS is a delicate operation prone to problems. Maintaining and applying updates would also impose a major overhead and potentially leading to instability of the environment.

Another known challenge with IPA is diskless HPC compute nodes. Since clients are added to IPA realm through enrolment process by writing data to a local disk, diskless nodes are unable to retain enrolment data after reboots. This dictates diskless nodes are removed from the IPA realm and added again after each reboot. This becomes an expensive and cumbersome operations for large computational environment. However, IPA allows for restoring client’s state using a keytab file [13], this option is not reliable and fails during testing when the keytab file was few days older making de-enrolment and re-enrolment the only option available.

Reduced entropy or the lack of it at times of bulk enrolments challenge is another known problem [14]. Due to the way private keys are generated, IPA servers could run out of randomness or entropy. Though this recovers by itself, it causes delays and slows down enrolments, normally only witnessed for large number of disk full clients. Unfortunately, with the keytab backup mechanism and re-enrolments not working in a reliable fashion for diskless nodes, it can compound and trigger this potential problem even when trying to regulate the number of diskless nodes to boot at a one time.

In a scenario of migrating from legacy NIS using users, groups, and netgroups, a limitation in IPA was discovered causing slowness accessing compute nodes through secure shell (SSH). Although this by itself is not a problem for Linux servers normally, it caused running jobs on HPC node start-up to timeout and fail. Testing revealed a correlation between nested netgroups and how IPA stores and consequently retrieve information from LDAP server. Although not all users are affected by this limitation, it still important to think about as nested netgroups obstacle must be solved for moving forward.

V. THE LDAP FACTOR

In aid to overcome encountered challenges with IPA and in parallel effort to continue with the migration, another path was explored. Linux offers PAM plugins and libraries to allow LDAP authentication and resolution [15]. The IPA backend naturally allows clients connecting natively through the IPA protocol, but also LDAP and LDAPS. As such, the IPA meshed server backend required no alteration. Clients were talking secure LDAP to the IPA servers. Though IPA native client (SSSD) include advance features and allows elaborate logging and debugging these logs can be quite difficult to interrupt. On the other hand, the LDAP approach with NSCD offers a much lower difficulty threshold. The debugging capabilities of NSCD brought the mentioned SSH login slowness to a resolution. Scoping certain search queries, especially for netgroups, nearly eliminated the delay for public key based SSH logins.

TABLE I: Measured public key based SSH Login timings

Public key SSH login	**NIS**	**IPA**	**LDAP+NSCD**
Simple User	170ms	300-1400ms	110ms
User with nested netgroups	1000-1600ms	600-10,000ms	700ms
User in many groups (> 13)	1000-1200ms	900-1200ms	800ms
Root	260ms	300-1400ms	200ms

Though some of the HPC compute running using IPA, most compute nodes were migrated to native LDAP in combination with NSCD [16] to offload the backend by caching recently used data. The remaining IPA compute nodes rendered problems weeks later causing SSH logins to fail due to a Kerberos bug [17]. These nodes were then migrated to use LDAP as well.

VI. CONSIDERATIONS AND RECOMMENDATIONS

Back-End Tuning

At peak times during testing, back-end servers did see an increased system load level. In some cases, IPA servers were crashing due to high load. The instability of IPA services was resolved by applying what is called NUMA node tuning. The Linux kernel allows managing process affinity using numactl [18]. Placing the process and their respective memory segments within the same NUMA node can drastically improve performance. In case of IPA, we observed a significantly reduction in SSH login times and responsiveness when utilizing NUMA node bindings for LDAP process.

Another optimization was applied by striping down IPA servers unneeded services. CA certificates and DNS removal from IPA services provided more optimized stack and eliminated additional resources overhead.

VII. CONCLUSION AND FUTURE WORK

Migrating to a modern authentication solution like IPA for large HPC computing installation is not as easy and straight-forward as people think. Still, with proper planning and a lot of customization, it is proven to be viable. Considerations such as optimal number of IPA servers, segregation of clients' network traffic, back-end tuning, building proper replication topology, and most importantly selecting the right protocol are important factors for a successful migration.

The maturity and reliability of IPA has improved over the years but using out of the box implementation strategy is not recommended. Using IPA native clients' for HPC large computing settings is tricky and, in most cases, does not work. LDAP with NSCD clients' side caching combined with client-side auto load-balancing offer a working solution. It allows for granular mapping and scoping that contributes thus reducing the time spend traversing and collecting necessary data from the LDAP repository eliminating login delays prevalent in HPC workload. Although currently LDAP/NSCD are the way to go for HPC, future work involving native IPA clients is still needed.

REFERENCES

[1] WikiPedia, "Network Information Service," [Online]. Available: https://en.wikipedia.org/wiki/Network_Information_Service.

[2] LdapWiki, "NIS Limitations," [Online]. Available: https://ldapwiki.com/wiki/NIS%20Limitations

[3] O. C. a. i. affiliates, "How NIS+ Differs From NIS," [Online]. Available: https://docs.oracle.com/cd/E19683-01/816-2074/6m8esseln/index.html.

[4] U. o. A. Canada, "Network Information Service Overview," [Online]. Available: https://sites.ualberta.ca/dept/chemeng/AIX-43/share/man/info/C/a_doc_lib/aixbman/commadmn/nis_intro.htm.

[5] FreeIPA, "User Documentation," [Online]. Available: https://www.freeipa.org/page/Documentation#By_Component.

[6] RedHat, "Managing Replication Agreements Between IdM Servers," [Online]. Available: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/identity_management_guide/ipa-replica-manage.

[7] FreeIPA, "Deployment Recommendations," [Online]. Available: https://www.freeipa.org/page/Deployment_Recommendations#Multi-site_deployment_awareness.

[8] RedHat, "Deployment Considerations for Replicas," [Online]. Available: https://access.redhat.com/ documentation/en-us/red_hat_enterprise_linux/7/html/linux_domain_identity_authentication_and_policy_guide/replica-considerations.

[9] RedHat, "Planning the replica topology," [Online]. Available: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/planning_identity_management/planning-the-replica-topology_planning-dns-and-host-names.

[10] M. Corporation, "CVE Details," [Online]. Available: https://www.cvedetails.com/vulnerability-list/vendor_id-42/product_id-61/MIT-Kerberos.html.

[11] T. Henderson, "Your Death By Kerberos," [Online]. Available: https://smartbear.com/blog/test-and-monitor/your-death-by-kerberos.

[12] D. Lichteblau, "ldapvi," [Online]. Available: http://www.lichteblau.com/ldapvi/.

[13] FreeIPA, "Forced client re-enrollment," [Online]. Available: https://www.freeipa.org/page/V3/Forced_client_re-enrollment

[14] FreeIPA, "Releases," [Online]. Available: https://www.freeipa.org/page/Releases/4.0.0.

[15] A. d. Jong, "LDAP authentication with nss-pam-ldapd," [Online]. Available: https://arthurdejong.org/nss-pam-ldapd/setup.

[16] ldapwiki, "NSCD," [Online]. Available: https://ldapwiki.com/wiki/NSCD.

[17] R. Bugzilla, "Bug 1414302 - gssproxy does not managing its credential cache properly," [Online]. Available: https://bugzilla.redhat.com/show_bug.cgi?id=1414302.

[18] A. Kleen, "die.net," [Online]. Available: https://linux.die.net/man/8/numactl.