Identity-Based Internet Protocol Network

G. Nakamoto, R. Durst, C. Growney, J. Andresen, J. Ma, N. Trivedi, R. Quang, and D. Pisano, *MITRE Corporation*

Abstract— **The Identity-Based Internet Protocol (IBIP) Network project is experimenting with a new enterprise oriented network architecture using standard IP version 6 protocol to encode user and host identity (ID) information into the IP address. Our motivation is to increase our security posture by leveraging identity, reducing our threat exposure, enhancing situational understanding of our environment, and simplifying network operations. Our current implementation plan uses credentials from the Common Access Card (CAC) to establish a 40-bit user ID and credentials stored on the computer's Trusted Platform Module (TPM) to establish a 40-bit host ID. The remaining part of the IP address can be a standard (/48) network prefix or support a (/32) prefix and a 16-bit group tag. A registration process (built on top of an 802.1x security framework) then occurs between the host and a registration server (which is currently an enhanced RADIUS server). The IBIP registration server then validates the credentials and automatically configures the edge router, fronting the host, with appropriate access privileges so that no IP address spoofing (or impersonation) is permitted. Hosts that are client machines do not have their IP addresses advertised across the network - basically making them unreachable or hidden from reconnaissance initiated by other clients. Servers have their IP addresses advertised as usual. A unique IPv6 extension header was conceived to enable return traffic to hidden clients. This approach will also provide support for approved peer-to-peer applications which may have hidden clients at both ends (voice-over-IP phones, for example). All infrastructure devices (routers, switches, DNS, DHCP, and other designated servers) are also not directly accessible by end user machines. For servers, the user ID is replaced with a service ID which can be used to identify and enforce policies on what the server is permitted to do. For example, if the server policy is to function only as a web server, access control implemented on the edge router in front of that server would only permit web transactions from entering the network. Attempts to use other non-approved applications such as telnet or ssh can be explicitly blocked or monitored and reported. These access controls are created and deployed from the IBIP registration server without human intervention, reducing the likelihood of human error while simplifying configuration and training. All policy violations are also reported via syslog messaging (using existing infrastructure devices) which enhance situational understanding. In summary, this network architecture hides a majority of the machines and infrastructure devices from unapproved access, enforces strong ubiquitous authentication for both host and user, enables enforceable authorization policies, simplifies the configuration of routers, and provides improved situational understanding**.

***Index Terms*— Attribution, identity, IPv6, next generation networking, policy-based networking.**

I. Introduction

The Identity-Based Internet Protocol (IBIP) Networking project has developed a new network architecture using standard IPv6 protocol features to encode user and host identity (ID) information into the IP address. Our motivation is to increase our security posture by leveraging identity, reducing our threat exposure, enhancing situational understanding of our environment, and simplifying network operations. This architecture *hides* a majority of the machines from unsolicited access (enables a “need-to-know” concept for access), enforces strong ubiquitous *authentication* for both host and user (prevents impersonation), and enables enforceable *authorization* policies for servers (prevents non-approved applications from accessing the network). The architecture results in a policy-driven network configuration, in which all but initial, basic router settings are fully automated. Situational understanding is significantly improved as a result of comparing unfolding events to authorized policies. Any action that is not authorized becomes a policy violation and can be monitored and permitted (permissive mode) or monitored and blocked (restrictive mode at higher threat conditions). The high level concepts for this IBIP network architecture came from a six-month duration DARPA funded initiative entitled “Alternative Network Architecture Analysis” [1]. Since then, MITRE has, for the past two years, continued evolving the concept and developed a prototype network that spans between Bedford, MA and McLean, VA.

II. Research Objectives

The objective of our research is to determine whether we can, for an enterprise of the size of the US Army (roughly ~2 million), provide a significant enhancement to network security by making three fundamental changes to the network architecture while preserving the investment in existing equipment:

The first change is to use *identity* (of hosts and users) as the basis for addressing in the network instead of the current practice of using network topological addresses. This identity information is supported by strong authentication mechanisms to establish access control and authorization policies for the network. We initially chose to work with IPv6, with its 128-

Manuscript received April 6, 2012. This work was supported in part by the U.S. Army Contract No. W15P7-12-C-F600.

G. Nakamoto is with the MITRE Corporation, Bedford, MA, USA 01730 (781-271-3032; fax: 781-271-2423; email nakamoto@mitre.org)

R. Durst is with the MITRE Corporation, McLean, VA, USA 22102 (703-983-7535; fax: 703-983-7142; email durst@mitre.org)

All other authors are also with the MITRE Corporation.

bit address, instead of IPv4, which has a 32-bit address, in order to carry this identity information with each network data packet. However, in FY12, we developed a concept to use IPv4 as well (see section 4).

The second change seeks to reduce the *threat surface* in the network by limiting access to end-user systems in the network while permitting full visibility of server systems. This limited access to end-user systems must be mitigated to support return traffic from a server back to the client end-user system, *authorized* peer-to-peer communications, such as Voice over IP (VoIP) telephones, Help Desk functions (remote desktop connection), and certain background services such as asset inventory or patch management service. We also wish to make the infrastructure devices *inaccessible* to the typical user to further provide protection of the infrastructure as well as create a detection capability for potentially malicious behavior (direct attempts at accessing infrastructure devices as a policy violation). This change basically hides all client machines and infrastructure devices from network reconnaissance while being able to quickly detect when such attempts to access these hidden assets occur.

The third change is to dramatically improve the network configuration control and situational understanding, and in doing so, to improve Network Operations. By establishing and implementing a set of *permissible-use policies*, Network Operations personnel may become informed of exceptions to these policies to identify and address emerging issues in the network before it can affect mission operations. *Permissible use*, in this context, defines what a system (client or server) is permitted to do on the network. For example, if a web server is authorized to function only as a web server (advertising ports 80/http and 443/https, for example), it is not authorized to run a trivial file transfer service (or tftp – using port 69, for example). Consequently, the server is also not authorized to run *malware* that uses other ports. Any attempt to execute such application is clearly detected (and blocked) with existing infrastructure equipment and requires no additional sensors. To be practical, however, these policies and reconfiguration of the infrastructure must be automated (not involve user intervention) in order to simplify network operations and monitoring. Figure 1 illustrates these three key objectives.

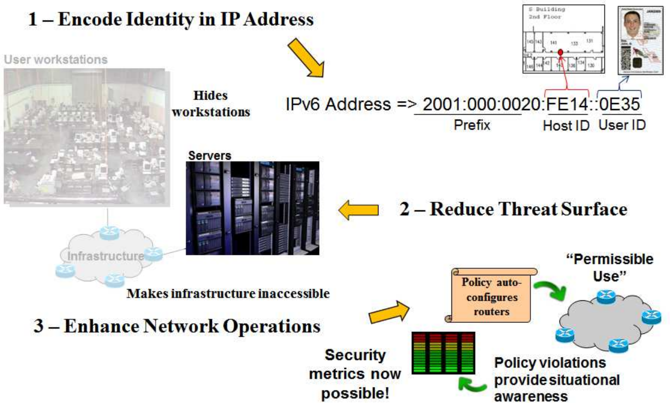
1 – Encode Identity in IP Address
User workstations
Hides workstations
IPv6 Address => 2001:000:0020:FE14::0E35
Prefix
Host ID
User ID
S Building
2nd Floor
Servers
2 – Reduce Threat Surface
Infrastructure
Makes infrastructure inaccessible
"Permissible Use"
Policy auto-configures routers
3 – Enhance Network Operations
Security metrics now possible!
Policy violations provide situational awareness

Figure 1. Three Key Objectives of IBIP

III. Technical Approach

To carry out the above objectives, IBIP introduces three key enhancements to the existing network. The *first enhancement* is the introduction of a registration server that provides the authentication and network access service. The *second enhancement* is associated with reducing the threat surface by hiding client machines, implementing organizational segregation, and anti-spoofing. The *third enhancement* that IBIP introduces is additional network operations information which results in improved situational understanding.

A. *Authentication and Network Access*

One of the key tenets behind the IBIP architecture is the incorporation of user and host identities into the IP address. Some of these ideas came from exploring extensions to the host identity protocol work [2] and concepts to prevent source address spoofing [3]. From a practical viewpoint, we envisioned that a person's Common Access Card (CAC) could form the basis of the user's identity and the private key held in the computer's Trusted Platform Module (TPM) could form the basis for the host's identity. Other authentication mechanisms are possible as long as traceback to and authenticity of the credentials could be assured.

Our initial approach was to use the existing and standardized 802.1x security framework with dual certificates (one representing the host and the other representing the user). A commercially available software application called a supplicant is used to initiate the 802.1x security handshake with the local Ethernet switch to gain network access. The supplicant uses the host's certificate to gain access to the 802.1x-aware Ethernet switch. The supplicant on the host presents the user- and host-credentials to the switch, which then initiates a secure exchange with our registration server to authenticate the credentials. Successful authentication provides notification to the switch to modify the switch port configuration from the "not connected" (or mitigation network) position to the normal network access virtual local area network (VLAN). Our specific instance of the registration server uses the standard RADIUS protocol to carry out the actual authentication service (as part of the normal 802.1x framework). Failure to authenticate either the host or the user results in no network connectivity (or constrained to the mitigation network). After access to the switch port is authorized (to the "normal" network), the registration server calculates the IPv6 address of the host using the combination of the user and host credentials and issues the address to the Dynamic Host Configuration Protocol (DHCP) server. This IPv6 address is bound to the host's credentials (further processed as a 128-bit universally-unique identifier or UUID) such that only that host can retrieve this IPv6 address from the DHCP server. The IPv6 address is fundamentally subdivided into three parts: the network prefix (32 bits), the host identity (48 bits), and the user identity (48 bits). Figure 2 shows such an implementation. Another variant reduces the 48 bits of the identity field to 40 bits and combines the resultant 16 bits to represent a group tag. This group tag can then be used to identify a higher level organizational abstraction for which group access policies can be created and enforced.

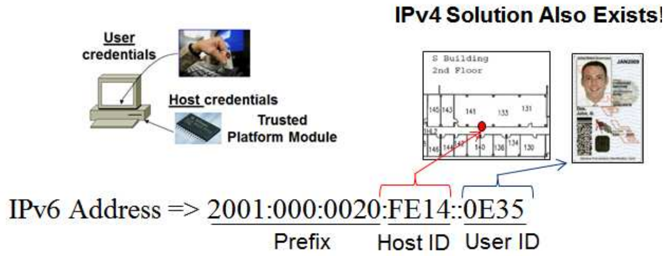

Figure 2. Example of Using Credentials to Create IP Address

Within the enterprise network, the address is treated as a "/128" address (in other words, only one host per network). While the host is obtaining its IPv6 address from the DHCP server, the registration server consults the policy database for that user-host combination in order to *automatically* configure the switch, a proxy router, and edge router supporting that host with new configuration information such as ingress[1] filters (derived from the role of that host and user). A *proxy router* was inserted between the edge switch and the edge router to carry out routing, filtering, and IP address manipulation not supported in existing routers. Its use enables the continued use of all existing infrastructure equipment with no modifications. However, it is envisioned that its functionality can be (one day) migrated to existing routers or switches as value added features. In our current implementation, the switch is also pre-configured as a private VLAN such that every port on that switch (except for one) is completely segregated from the other ports[2]. The remaining port is configured as a "promiscuous" port (which carries all traffic from the other private VLAN ports) and is connected to the proxy router. The Ethernet switch is also auto-configured to lock down the MAC address to the physical port of the switch. The proxy locks the MAC to the IP address. Any change to the above results in the port transitioning back to a "not connected" state or the proxy dropping the packet and logging the event as a violation. This network admission control process is illustrated in figure 3.

A. *Reducing the Threat Surface*

The second key IBIP tenet seeks to reduce the threat surface in the network by limiting access to end-user systems in the network while permitting full visibility of servers. In effect, we try to "hide" our client workstations as well as all infrastructure devices such as routers, switches, and network operations servers. "Off by Default" concepts [4] inspired this line of thinking. After all, if you can't see it, you can't attack it. However, more dynamic means to implement changes in policy were necessary and the idea of using bloom filters in a dynamic environment was not acceptable. Instead, we chose to have client workstations specifically register as clients that are treated in a special way. All "hidden" clients have source IP addresses that are *not routable* within the network and, as such, cannot be accessed by another machine (client or server) without a "need-to-know" (even if one knows the IP address). We currently use the low order bit of the group tag to encode if this IP address is to be routed or not. The proxy then enforces this encoding which is also validated during the registration process when the address is created. A client cannot, therefore, scan (discovery technique) or communicate with another client in general. Any attempts by a host to access hidden client (without a "need to know") or override policy by modifying the address is clearly identified as a policy violation, syslog'ed, and alerted to network operations. This "need to know" may be temporary such as when a server needs to know how to communicate back with the client (that initiated the communication) or semi-permanent such as VoIP phones (where each VoIP phone is considered a hidden client). Server IP addresses are advertised and clients can send a packet to the server. The next logical question is how does the server communicate back to the client if it is unreachable? When a hidden client sends a packet to a server, our proxy router detects that the source is a hidden client and processes the packet by adding a new IBIP IPv6 *extension header*. The IBIP extension header uses the standard IPv6 extension header option to preserve the client's source IP address in the extension header while it replaces the source IP address with that of the proxy's IP address (which is routable within the infrastructure). In other words, the proxy replaces the hidden source's IP address with its own IP address and "stores" the client's IP address in the extension header. The resulting packet has fully routable source and destination addresses and can support asymmetric network routing paths. An example of this process is shown in figure 4.

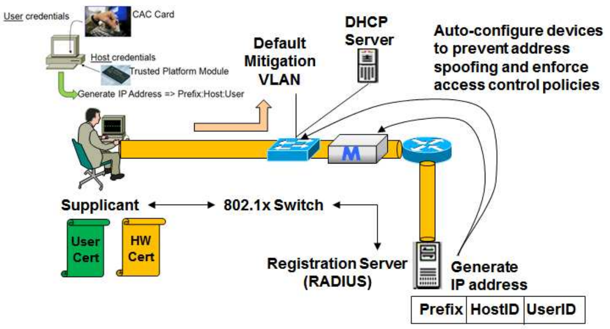

Figure 3. IBIP Network Admission Control

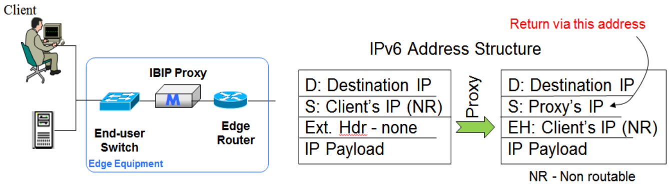

Figure 4. IBIP "Hidden Client" Header Extension

[1] Throughout this document, the term *ingress* will mean *toward the network* (away from the client or server).

[2] The use of Private VLANs is specific to the Cisco switches we are currently using, and are a matter of convenience. Other techniques exist that are applicable to other switches. As part of a transition into operational use, the equipment database that associates the switch port with specific vendor equipment must be augmented with configuration routines appropriate for that equipment. The Cisco Private VLAN configuration routines will be assigned to a library of functions appropriate for Cisco switches, while other routines will be developed and associated with other switches. To the extent possible, we use capabilities that are standard across many different kinds of equipment (for example, the use of 802.1x), however, in certain cases, the standardization hasn't caught up with the functionality. In those cases, we accommodate that while maintaining the auto-configuration properties of IBIP by having these device-specific configuration libraries.

As the packet arrives at the edge proxy near the destination, this “destination” proxy router detects that the packet has an IBIP extension header and processes the packet to reconstitute it back to its original state (less normal decrementing statistics). At the same time, that proxy maintains a short-lived cache of the “return via” proxy address to hidden client address association (representing the “need to know”) to support the return traffic from the server to the client. When the server sends the packet using the hidden client’s IP address, the proxy checks its “return via” cache. If it finds an entry for that client’s address (associated with that server’s address), then it will replace the client’s address with the proxy’s “return via” address and place the client’s address in an extension header. If it doesn’t find such a cache entry (no initial contact or entry has timed out), then the proxy discards the packet due to being “unreachable” (and NetOps can be notified). When the packet arrives at the proxy that fronts the client (e.g., the return destination), the proxy detects the existence of the IBIP extension header and processes the packet to re-instantiate it back to its original state (as sent by the server). This process maintains the data integrity of the packet between source and destination (enabling end-to-end IPsec, for example).

In order to accommodate authorized peer-to-peer (P2P) applications such as voice-over-IP and other collaboration tools, such workstations (or handsets) can be registered to be authorized P2P applications. Such a registration process will permit the proxy to obtain the necessary information to populate its “return via” data cache without having received a packet from the other source. The same model will work for Help Desk operations and background service applications such as asset inventory or patch management.

In addition to hiding client machines, IBIP addressing also has a powerful feature that permits implementation of an organizational or group structure that can be used for creating ingress or egress policies. These policies, in turn, can create logical network segmentation and separation. While the original IBIP IPv6 address was segmented into three sections: 32 bit network prefix, 48 bit host identity, and 48 bit user/service identity, we are experimenting with subdividing the host/user identity such that 16 bits (8 bits from each identity field) of the 96 (combined) bits of the identity fields can be used for what we are calling a *group tag*. In fact, the use of the low order bit of the group tag has become a key means for supporting the identification of “unroutable” IP addresses (and thus, has now become the norm). This approach still leaves 40 bits for identity (over 1 trillion values for users and another trillion values for hosts – which should be sufficient for most enterprises). Using this approach we can use the remaining 15 bits of the group tag to provide another means to identity the host-user as part of some higher level group. For example, all infrastructure devices (switch, routers, proxy, registration server, DHCP server, etc.) can be designated as *infrastructure* (as a succinct and different *group*). A policy can then be created that prevents access from any non-infrastructure device to any infrastructure-addressed device basically making the infrastructure inaccessible to the end user. Even if router *credentials were stolen* and some insider (or adversary on an inside machine) had the usernames and passwords of routers within the infrastructure, that insider (or hacker) *would not be able to access such infrastructure devices*. Any attempt to do so will be clearly flagged and alerted to NetOps.

B. Network Configuration Control and Situational Understanding

IBIP hides clients while leaving servers visible. A natural consequence of this is that malicious intruders will put more focus on attacking the servers. This will be especially true if malware is on the client (which may be difficult to avoid), or if an insider is trying to retrieve information for exfiltration, or if a bad actor has managed to remotely log in and is trying to establish additional footholds. For client machines, source IP filtering on the ingress direction is relatively straightforward from a policy viewpoint. Server side ingress filtering can become complicated and if implemented manually, it can also become a significant source of vulnerabilities introduced by human configuration mistakes. As such, this approach has not been pursued with much success in the past. IBIP’s use of identification, registration, and anti-spoofing are enabling features that allow us to reconsider this approach. Experimentation with IBIP’s automated network configuration capabilities indicate that these potentially complex configurations (derived from policies a user can understand) can be implemented automatically without human intervention. *A key challenge is whether we can accurately define policies for what a server is permitted to do without ambiguity and gain security benefits without increasing complexity.* To address this challenge, we commissioned a policy-oriented penetration test in which a set of systems (that had been brought together as part of a larger scale resiliency experiment) was targeted as an environment for such policy analysis. Our goal was to determine if such policies could be succinctly defined such that the operational use of the system was not impacted while effectively thwarting the attempts of a highly skilled insider-based network penetration team. *The initial testing indicates that these policies can be defined and that such policies do prevent network reconnaissance and unauthorized applications from accessing the network.* In addition, unauthorized activities are clearly identifiable with minimum false alarms.

The success of the policy-based network configuration definition in our initial penetration testing is significant, in that it validates another key implementation concept in the IBIP approach: that policies can be defined and applied in a hierarchical manner, with a subordinate organization inheriting the policies of its superior organizations and implementing supplementary policies as needed. It also greatly simplifies a technique known in the Army as “Task Organization,” in which a unit from one organization is assigned to a different organization to support a mission. For example, a Radio Company from Brigade A could be assigned to Brigade B by simply causing it to inherit Brigade B’s policies. All equipment in the Company can be updated by a single reassignment action which causes the address structure and network access policies associated with Brigade B to be applied to the Radio Company in transition. The next time anyone in the Radio Company logs in, that person will now be part of Brigade B and have the IP address associated with that organization (along with all approved access policies). The host and user IDs typically will remain constant while the group tag will now reflect the new organizational hierarchy (along with any network prefix changes). We have augmented our Network Operations control console with the ability to see and manipulate an *Organizational View* of the network, in addition to a *Topological View*, which is more typical for network operations. This Organizational View makes it easy to manipulate organizational entities, such as the example Radio Company, through a simple click-and-drag interface, so that the reassignment of the Radio Company to Brigade B is effected through a single mouse action that applies to the entire Company. This action causes all systems associated with that company to be updated with the policies associated with Brigade B including re-addressing all systems within that Radio Company. In this specific case, the host and user identities still remain the same but the group tag portion of the IP address has been updated to reflect the change in organizational "ownership." This tag can then be used to create access control policies specific to the organization (as well as finer grain controls associated with the user/host). In addition, tags can be dynamically assigned as part of a holistic view of what is happening. In addition to user, host, and organizational identities, this tag could also represent physical location, time of day, proximity to recent login, and many other concepts not envisioned as yet. It provides the potential to associate the address with contextual understanding. We have only started to explore the many possible uses for this type of policy-based logical segmentation through policy definition.

IV. On-going Development

In 2012, two key on-going research and development activities are an IPv4 version of an IBIP-enabled network and a Trust Gateway to permit communication between an IBIP-protected network and the rest of the world.

The IPv4 version of IBIP is based on a transport layer *shim* that sits between the IP header and the TCP or UDP header (using an experimentation protocol type in the protocol field). Its field break out is virtually identical to the 128 bit IPv6 address structure with one exception. In lieu of the (32-bit) IPv6 network prefix, the IPv4 address is kept in that slot. This permits the "hidden" IPv4 source address to be swapped for the proxy's address while still carrying the proxy-to-hidden-source association in the packet. In theory, servers do not need to have such handling. However, if such handling is omitted from server communication, the ability to use the group tag (contained in the shim) would disappear. As such, the current investigation is examining the pros and cons for adding this shim to all devices (but at a cost of increasing the packet size shim by 32 bytes vice 16 bytes without server support). Various means to compress this information using technology related to robust header compression and dictionary look up substitution are possible but presently outside the scope of this effort.

The Trust Gateway (TG) is the IBIP boundary device that performs IBIP specific network address translation between IPv6 and IPv4 systems as well as providing the DNS translation services. In addition, as packets enter the IBIP network "from the outside," the TG can appropriately mark the group tag of the source along with validating the host and user identities (if appropriate) such that fine grain (if desired) access policies can be enforced on that traffic. The TG will support external VPN connectivity along with user and host authentication.

V. Results and Accomplishments

The prototype IPv6 infrastructure has been operating for several months. Some preliminary performance and penetration testing has been conducted with encouraging results.

Currently, the IBIP proxy represents a performance constraint given the IP header processing and access control filtering it conducts on a per packet basis. One instance of our proxy, currently implemented on a Dell Precision 690 (Core-2 Quad Processor), can support a sustain throughput of 400 Mbps (on a one gigabit per second interface card) using 1 KB packets while supporting a total of 1300 filtering rules. This translates to being able to support 130 workstations (or servers) with each system having, on average, 10 filtering rules. In most cases, the typical workstation is only expected to support a handful of rules (primarily source IP admission). Servers, on the other hand, can have much more complex filter sets. However, it is the aggregate in the proxy that limits the overall packet per second threshold. The unidirectional throughput (ingress direction) of this proxy as a function of frame size (and aggregate filter rule size) is illustrated in figure 5.

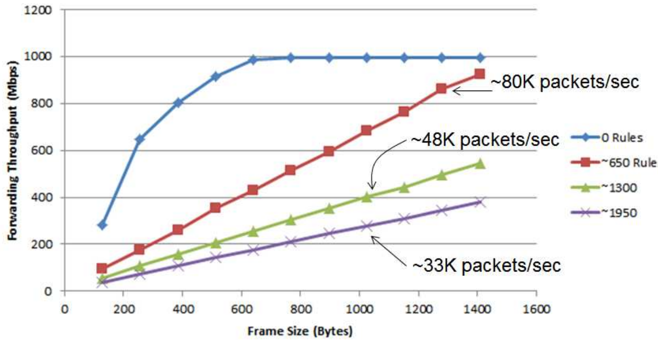

Figure 5. Proxy Throughput as Function of Filtering Rules

For penetration testing, the test team was comprised of MITRE individuals skilled in penetration testing practices, uninvolved with the development of the IBIP capabilities, but fully briefed and tutored on its theory of operation and implementation by the IBIP research staff. The penetration test was conducted in a progressive manner with each objective having sub-objectives. If the pen test team could not accomplish the initial objective, they were provided with additional information (such as credentials) to move to the next sub-objective. A network operations team was constituted to monitor the NetOps displays. Their role was strictly passive with explicit instructions not to prevent any detected penetration other than what automated policies carry out. In addition, they were to reconstruct, in real time, what the penetration testers were doing and not rely on forensics. The penetration test had five key objectives:

1) Gain access to the network.

2) Use unregistered or stolen user credentials to gain access to the network.

3) Conduct network reconnaissance (once validated with full network access).

4) Impersonate or spoof another IP address (fake or validated in-use address).

5) Gain access to a server (write a file on target server) using a simulated zero-day exploit or a pre-planted backdoor.

The penetration test team was ***not successful*** in any of their objectives. Throughout the test, the pen test team was provided with any information they needed such as Ethernet addresses and IP addresses. They had full knowledge of the entire network topology and, in many cases (objectives 2- 5), operated as full insiders with valid user and host credentials. They were provided with a computer platform that had valid credentials (except during test objective 1 where only valid hardware certificates were provided) and were also allowed to bring their own machines (to which we would provide valid hardware certificates). For objective 5, the team was provided a username/password to access a telnet server (representing the pre-planted backdoor) on a server platform that was registered to operate as a web server (our permissible use policy). Our network operations team was the defensive blue team and was instructed not to take any corrective actions. They were able to reconstruct a majority of the pen test team activities with few missteps. They were also able to carry out their situational understanding without having to pore over historical log files.

Another outgrowth of this study has been the ability to *quantitatively* collect security metrics that provide some sense of the "goodness" of the security posture of the network – a historically difficult challenge. Being able to say that there have been no network scans or unauthorized applications now has more meaning. However, exploits that operate through approved channels (ports and protocols) are still undetected by the IBIP architecture. By corralling potentially malicious activities within authorized channels, host-based application oriented firewalls (or specific application layer firewalls) can take on a new meaning and purposefulness.

The results of these initial tests indicate that these attributes increase overall network security. If the role of network security is to authenticate the origin of all traffic (including anti-spoofing), enforce a need-to-know for access/reachability, and force the traffic to operate within approved channels (ports and protocols), these enforceable policies can provide the following security benefits:

- Limit classical network reconnaissance
- Prevent unauthorized applications from entering the network
- Reduce the threat surface – potentially slowing down the spread of malware
- Enable accountability and reliable traceback mechanisms, and
- Improve situational understanding that can provide quantitative security metrics that enables confidence in the "goodness" of the network security status.

VI. Conclusion

We believe that there is strong potential that the IBIP network architecture can significantly improve the network's security posture and enhance the situational understanding of the infrastructure. The IBIP network use of authentication and admission control does not permit anonymous traffic and enables IP traceback (within the enterprise, thus, enabling accountability into the network). It also attempts to define "permissible use" of all servers – thus enabling the monitoring of potential policy violations. IBIP's policy-based approach to network access, configuration, and monitoring requires network and security operations teams to understand what applications and servers are running on the network. They also need to consciously decide what the policies *should be* regarding those applications and servers. These policies are reusable, durable, and the key benefit is a more secure network – one that has "raised the bar" for cyber attackers to establish and maintain a foothold within the enterprise. However, there is added work to understand and create these policies. In addition, an IBIP network will provide one with the most up-to-date knowledge of what and who is on the network as well as provide unprecedented situational understanding of policy violations. If IBIP concepts can force all activities (including malware and insider activities) to operate within approved authorized "channels" (e.g., ports and protocols), can host based security benefit and focus on more specific application layer behavioral monitoring and reduce the attack surface of the host? This is a hypothesis that is being further explored in FY12.

Acknowledgment

The early work that spawned the fundamental ideas promoted in this paper originated from a DARPA study referenced earlier [1]. The following personnel were involved in that study that gave rise to the foundation concepts: Dr. K. Fall from Roland Computing Services; R. Watro and P. "mudge" Zatko from BBN/Raytheon; S. Lee, C. Corbett, and G. Stoneburner from John Hopkins University Applied Physics Lab; G. McAlum from Deloitte; and E. Giorgio from Ponte Technologies. In addition, the authors would like to thank Dr. V. Swarup for his support in overall project direction. This work was supported by the MITRE Innovation Program.

References

[1] Nakamoto, S., Durst, R., Corbett, C., Fall, K., Lee, S., Watro, R., Zatko, P., McAlum, G., Brusseau, F., Wu, A., "Alternative Network Architecture Analysis," MTR 100088, April 2010

[2] Gurtov, A., "Host Identity Protocol," John Wiley & Sons, 2008

[3] Anderson, D., Balakrishnan, H., Feamster, N., Koponen, T., Moon, D., Shenker, S., "Accountable Internet Protocol," SIGCOMM '08, August 2008

[4] Ballani, H., Chawathe Y., Ratnasamy, S., Roscoe, T., Shenker, S., " Off by Default," SIGCOMM '05, November 2005