UNISYS

A Better Way to Segregate Data by Classification Level

By David Frymier, Vice President and CISO, Unisys Corporation

White Paper

The Problem

Today, many companies see their once protective corporate network perimeter starting to crumble. Some would say that the secure perimeter is already gone due to a variety of factors including botnet malware that tunnels port 80 with encrypted traffic, ubiquitous wireless access from coffee shops and other possibly unsecure locations, BYOD programs, and consumerization of IT. This means the traditional “hard exterior, soft interior” network security model has become outdated and internal corporate applications need some heavier armor than the chain mail they are currently wearing. So in response, some companies have started to compartmentalize their networks and data centers by reclassifying their data based on need-to-know access.

A complex, expensive and risky approach to data security

Consider the recent problem faced by a hypothetical company, WidgetCo. The company has both externally and internally facing applications. Their external applications all conform to a classic three tier defense-in-depth[1] model. Their internal applications sit on the intranet and have historically relied on general network perimeter security – tiers, like the way the external applications are set up, and zoned. There are dozens, or perhaps hundreds of variations on these themes, including implementations using new “virtual” firewalls, but the basic ideas are shown in Figure 1 – Tiered Model, and Figure 2 – Zoned Model.

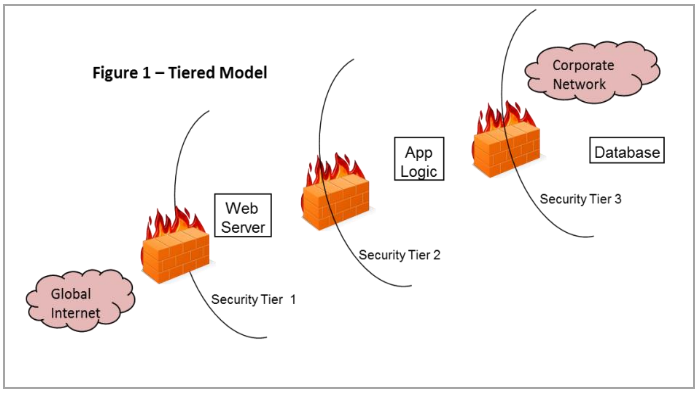


Figure 1 – Tiered Model

The tiered model isolates the major processing elements from each other using firewalls, with the most sensitive data (usually held in the application database) in the deepest tier behind most firewalls.

The zoned model is based on the concept of a hosted colocation cage. All the application functions sit inside a security zone defined by a firewall perimeter. All users are assumed to be potentially hostile, are strictly authenticated and their important activity is logged. Required external services (ISP, DNS, etc.) come from the hosting facility through the firewall.

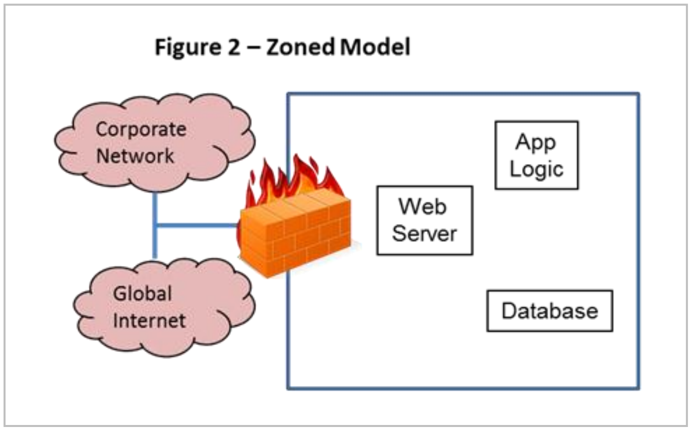


Figure 2 – Zoned Model

Each of these architectures provides security and connectivity using a combination of routers, switches and firewalls. These designs require expertise to design capital equipment, installation labor and ongoing operations and maintenance expense – all adding to the cost of the existing network.

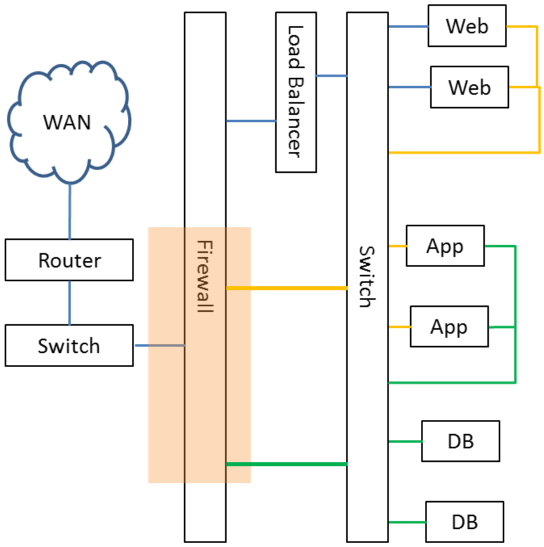
To deal with the crumbling perimeter problem, WidgetCo management felt like they needed to better secure their applications by compartmentalizing access. Let’s consider what this would generally look like with a traditional firewall-based approach.

In a tiered model, the general data path would be as shown in Figure 3 - Data Path. In actual implementation, what appears to be different devices in a logical representation like Figure 1 can actually be different ports on the same physical device, as shown in Figure 3. Incoming traffic from the wide area network (WAN) is routed to the firewall, then to the web server (the blue path) then to the application logic (the yellow path) and finally on to the database servers (the green path). The bulk of the actual traffic that comprises the work load for the firewall(s) is

[1] Defense-in-depth refers to the concept of layered security defenses, such that if an attacker breaches one countermeasure, they are immediately faced with another

between the application logic and database tiers. In a tiered model, this drives up the cost of the firewall from both a capacity and port count aspect.

Figure 3 – Data Path



In the zoned model, the firewall is only involved at the edge of the zone, so the ports and traffic represented inside the yellow box in Figure 3 go away. This makes the zoned model less expensive, but also less secure since it doesn't have the defense-in-depth of the tiered model.

Next let's look at the costs and structure of how WidgetCo could configure to solve the compartmentalization problem traditionally. WidgetCo costed out three approaches: putting everything into tiers, zones, and a mixed model approach – zoned for low and medium business impact application data, and tiered for high business impact data.[2] The mixed model approach was chosen since it offered the best mix of protection for high business impact data and lower cost for low impact business data.

Traditional Approach Requires a Lot of Work

Typical of many companies, let's assume WidgetCo has a portfolio of 320 internal applications, 120 of which are web enabled and fit roughly into the three tier model. The rest are utilities used by the IT organization; many don't even have a GUI. 30 of the 120 apps have high business impact data, and would go into tiers; 90 have low or medium impact data and would go into zones. The 90 apps fit into a total of five zones. WidgetCo has a data center population of 500 production servers and 700 development/test/quality assurance servers spread across two facilities.

The project boils down to inserting firewalls – the switch fabric and load balancers were already in place. The costs fall into five main buckets – capital equipment for the firewalls, labor to install the firewalls, labor to design and install network equipment associated with the rearrangement, labor to analyze, modify and test applications as they fit into tiers or zones, and ongoing maintenance.

The labor needed to establish a tier consisting of analytical work required to identify the members of the tier. Once that is accomplished, either physical cabling or VLAN definition (or some combination of the two) is required to establish the LAN environment. Standard firewall configuration and placement is not a big deal, but once that is done, firewall rules need to be designed, configured and implemented to establish the desired separation. Some changes in server configuration for default gateways and the like are also required. This needs to be done in the development, test, quality assurance and production environments. Establishing a zone requires the systems that will comprise the zone to be identified, VLANs or cabling work done to establish the zone edges, and firewall rules established to permit the allowed traffic across the zone perimeter.

In a modern data center, at the very least the quality assurance and production environments will require change requests and formal documentation, and WidgetCo is no exception. Extensive testing – which needs to be carried out by the application groups - of each new environment will be required, and while the mechanics of the new design can be worked out in the test environments and thoroughly tested in the quality assurance environment, the physical changes are still different in the production environment, introducing almost inevitable disruption.

[2] See U.S. Government publication FIPS-199 for a definition of these terms and an explanation of how to do the data classification

Initial design estimates call for several big firewalls for the tiered environments and several smaller ones for the zones yielding a project with a substantial large cash outlay and taking a year or more to complete. Furthermore WidgetCo was also concerned about the interference to their operations this data center work would cause.

A Better, Faster Approach

In late 2011, Unisys released a commercialized version of a product designed to provide encryption to multi-level secure networks in government and military settings. This product is called *Unisys Stealth^{TM} Solution for Network*. The earliest form of Stealth is network virtualization and protection software and appliances aimed at reducing the cost of multiple physical networks in multi-level secure environments. Think of a master sergeant sitting at a desk with three PCs – one Unclassified, one SECRET, and the other TOP SECRET. Each PC is on a separate physical network where all the machines share its security classification. That's a lot of network equipment and a lot of PCs. Stealth addresses this equipment proliferation problem by using a combination of encryption technology and "communities of interest" (COI) to create dynamic virtual networks that enable these different classified networks to be consolidated onto one physical infrastructure – saving roughly 2/3rds of the cost. While Stealth was originally developed with the military in mind, it turns out that there are many non-military environments with physically segregated networks – in education, hospitals, pharmaceutical companies, utilities, and others – where significant cost savings can be achieved by applying the COI concept. Also, as information security techniques get more sophisticated in response to the increasing capabilities of attackers, enterprises besides governments are finding it cost-effective to segregate their data by classification level.

There are examples of segregated and restricted access all around us in the non-cyber world. Consider a sports stadium. In addition to the different classes of seating and the skyboxes, customers who are there to watch the game aren't allowed in the food preparation areas, or on the field. Team members and coaches aren't allowed in the opposing team's locker room. These restrictions seem natural and obvious; if we turn back to the IT example, why should a computer system or workstation used to coordinate health care benefits have visibility and access to systems used to place manufacturing purchase orders?

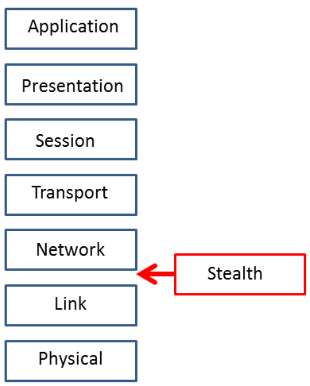
If we look at the concept of Stealth COIs and apply it to the problem of segregating zones and tiers, it turns out to fit pretty well. In fact, WidgetCo could meet four ambitious objectives:

1) ***Enhance application security*** and deal with security perimeter erosion by hiding the application infrastructure servers from visibility to any endpoint that does not have a direct need for access
2) ***Make no network changes*** (no cabling, no VLAN or LAN changes, no firewall rules)
3) ***Make no application changes*** – either code or configuration
4) ***Have end-users be unaware*** that any changes had taken place

Item #1 – Enhance application security

Ever since the dawn of the internet, security has been provided by firewalls at the Internet Protocol (IP) layer of the network stack, involving IP addresses, ports and protocols. This has been tedious and cumbersome, but effective. The objective has long been to "move up the stack" to where security aware applications could protect themselves, or "data should protect itself." This is an admirable goal but let's face it – it just hasn't happened, at least not yet. Stealth takes a step in that direction by moving the security function out of dedicated network equipment onto the workstations and servers, as shown in Figure 4, Stealth in the Network Stack. Stealth installs itself as a "shim" in the network stack between the link and network layers, where it can see the IP header fields and control traffic before it gets to the "real" IP layer.

Figure 4 – Stealth In the network stack

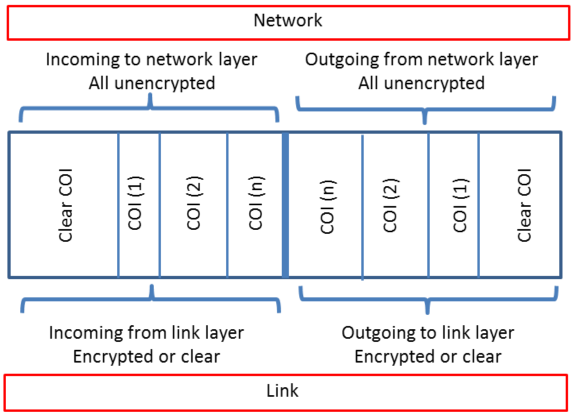


Stealth COI members are defined by Active Directory (or LDAP-based) group definitions. During the “opening” of a Stealth session, Stealth equipped endpoints authenticate to an authorization server, which returns the assigned COI key or keys to enable the endpoint to communicate within the COIs to which they belong. Somewhat analogous to firewall rules, Stealth COIs also have “filters” that allow traffic based on IP address, port or protocol. If a system is not a member of a Stealth COI by virtue of having the proper COI key or otherwise configured as part of a “clear” COI using filters, systems that are members of the COI become undetectable, or “cloaked” to these non-member systems because the Stealth software simply drops non-member traffic.

Items #2 and #3 – No network or application changes

The Stealth software operates in the network stack by encrypting the data payload of packets. It does this by intercepting traffic after the application decides to send data, but before it gets to network equipment. How this is done is shown in Figure 5 – Stealth network shim. The net effect of this implementation is that applications are unaware of the Stealth software; this is why they don’t have to change. Similarly, since network equipment uses packet headers – untouched by Stealth - to perform network functions and doesn’t care about the data content of packets, no network changes are required to support the Stealth implementation.

Figure 5 – Stealth network shim



Item #4 – End-users unaware of changes

Let’s assume WidgetCo is interested in dramatically improving the level of protection offered high business impact systems in their data center from infected systems or other intrusions on their intranet, so it chose not to extend the Stealth client to end-user systems. This meant that end-users were completely unaffected by the changes - how this was done is shown in *Figure 6 – Web server in clear.* A Stealth COI with encryption is defined to include the backend of the web server, the app logic server, and the database server. A Stealth COI is also defined without encryption to allow certain IP addresses or ranges of IP addresses access to the Web servers on typical browsing ports. In this manner, the application end-user can only determine the existence of the web server, and it will only respond on ports 80 (http) and 443 (https). As far as the end-user is concerned, the app logic and database servers don’t exist.

Putting it all Together

Let’s look at an example of how we can put all this together to protect a representative application. This application has a web server and application logic running in two virtual machines on the same physical server, and a database server on a separate physical box. There are global users all around the WidgetCo intranet, with a single application administrator and about a dozen infrastructure administrators in the data center who have access to its server components.

Figure 6 – Web server in clear

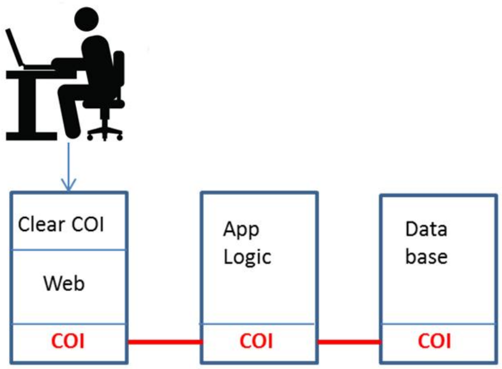


Figure 7 – Application in Stealth environment

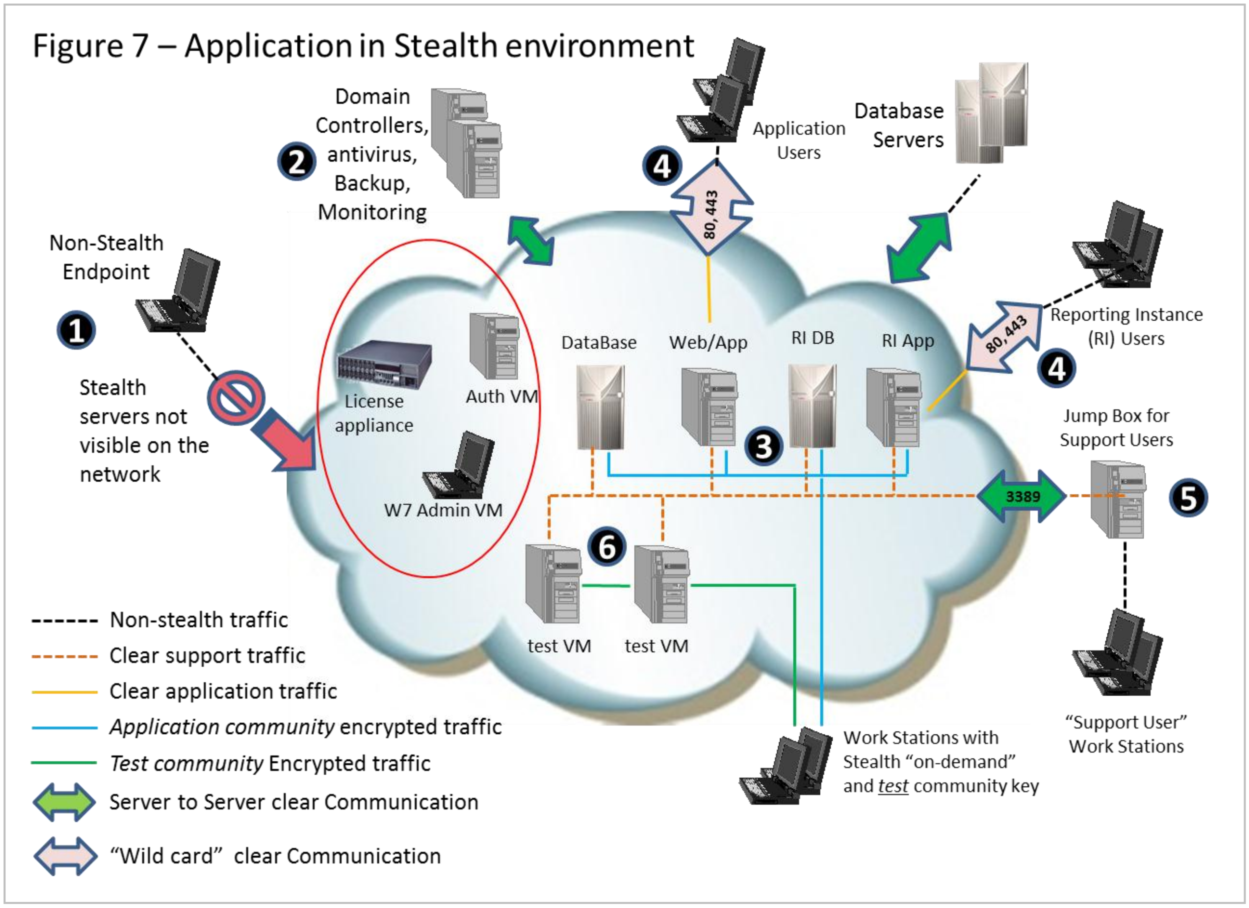

Domain Controllers, antivirus, Backup, Monitoring
2
Application Users
4
80, 443
Database Servers
Non-Stealth Endpoint
1
Stealth servers not visible on the network
License appliance
Auth VM
W7 Admin VM
DataBase
Web/App
RI DB
RI App
3
Reporting Instance (RI) Users
80, 443
4
Jump Box for Support Users
3389
5
6
test VM
test VM
------ Non-stealth traffic
------ Clear support traffic
Clear application traffic
Application community encrypted traffic
Test community Encrypted traffic
Server to Server clear Communication
"Wild card" clear Communication
Work Stations with Stealth "on-demand" and test community key
"Support User" Work Stations

Figure 7 depicts an example of a Stealth production environment, along with a few extra components that were used for testing. The Stealth components are inside the red circle, and consist of a license appliance, which manages and counts licenses in use; the authorization server, which performs authentications and hands out encryption keys; and an administrative workstation configured to be able to access the other two servers. The numbered points below correspond to the numbered areas of Figure 7.

1. The non-Stealth endpoints are all the IP hosts in the WidgetCo network who do not have access to the application – these systems can't ping or in any way discover the existence of the application logic and database servers, and can only infer the existence of the web server (running on the application box) as an application log-in prompt on http port 80.
2. A clear COI (the green double headed arrow) with IP address based filters has been defined to allow communication with Microsoft domain controllers, anti-malware parent servers, monitoring and backup systems. These are the common sorts of infrastructure services found in any data center and are needed by almost all applications.
3. The application reports data to the WidgetCo reporting instance for report generation; the reporting instance servers are configured on the same blue COI as the application servers so this communication can occur.
4. Both reporting instance and application users access the web servers of these applications through a clear COI on ports 80 and 443, shown as the light pink double headed arrow.
5. A "jump box[3]" has been configured as part of a clear text COI filtered to only allow the IP address of the jump box to access the six servers shown here for the purposes of system and database administration. In this manner, the jump box can control and log the traffic across the administration interface, which could originate from anywhere on the WidgetCo network.

[3] So called because it is used to "jump" or login to other servers

6. The two test VMs on the lower part of the cloud are there to demonstrate the logical separation afforded by their separate green COI definition. They can see each other, the jump box, and the workstations that are a member of their COI but they can't see the application or reporting instance servers, and the application and reporting instance servers can't see them.

Summary

Unisys Stealth Solution for Network allows IT organizations to deal with the security erosion of the general network perimeter by using data classification to establish smaller perimeters around related data and allowing access on a strict need-to-know basis. Establishing these smaller perimeters would traditionally be done using either tiered or zoned architectures requiring firewalls and modifying networks and applications to fit into the new security posture, incurring substantial capital costs and operational disruption. Stealth implements security perimeters by establishing communities of interest (COI) using encryption and group membership driven by LDAP access groups, including Microsoft Active Directory. Stealth operates as software inside server and workstation components, so companies can implement communities of interest without network changes, application changes, or end-user disruption.

Unisys has implemented Stealth technology for its own use throughout its primary and secondary data centers. In doing so, we've developed methodologies for analyzing the traffic environment around our applications to establish the required COI membership, as well as best practices for managing and maintaining COIs. If you are interested in the Stealth solution, we'd be delighted to have you visit our data and engineering centers to observe the implementation and its efficient results.

For more information visit www.unisys.com

Printed in the United States of America 01/14 14-0043