**EMCS2000 Introduction to Computer Security |** Module Report 7

Brian Russel Davis, brian\_davis@brown.edu

## Section 1

### Scenario

“The Network” was born in a safe place, among a group of trusted connections. Many have said that the network was built with no security in mind, but I respectfully disagree. The security implicit in the network was the relationship between the people using it. We understand this concept of trust because we experience it every day. We trust that the food we receive at a restaurant is not laced with arsenic. We trust that the family member walking through the door is actually our family member not an assassin in disguise. We don't double check the work of the safety technicians who do the inspections on the planes we fly in. Why do we trust these things, people and processes? When does a Zero Trust Network make sense and when does it not? In this part of the security plan I hope to explore the meaning of trust in a network, how we cultivate the level of relationship implicit in the minds of the creators ( since we are still using their framework ) and how we evolve that framework to accommodate the current level of relationship based on our usage and communication.

### Security Objective

The objective of this plan is to dynamically determine the trust that principals ( users, agents, applications, and devices ) should granted in the network and define the protocols in which this trust will be granted. The protocols of the network have already been established, therefore we will have to provide a method for invoking these protocols that help us understand who the principals are, and by way of authentication understand what their function in the network should be. In a previous section, we noted that “who” is merely a form following the function of what. In other words, identity is formed by a history of actions. If we determine what a principal is ( man, beast, vegetable ) then we will make a determination of who they should be allowed to be on the network. It is now our job to recreate the implicit trust experienced by the creators of the network, by altering the tools they gave us or by layering new tools on top or underneath what we have.

### Scope

#### General Audience

Network administrators are the general audience for this security plan because it contains technical concepts, terms, and documentation that only they will understand.

#### General Target

Our target for the security plan is the entire network at NASA HQ, including the remote staff but not including the other NASA campuses.

### Technical Description

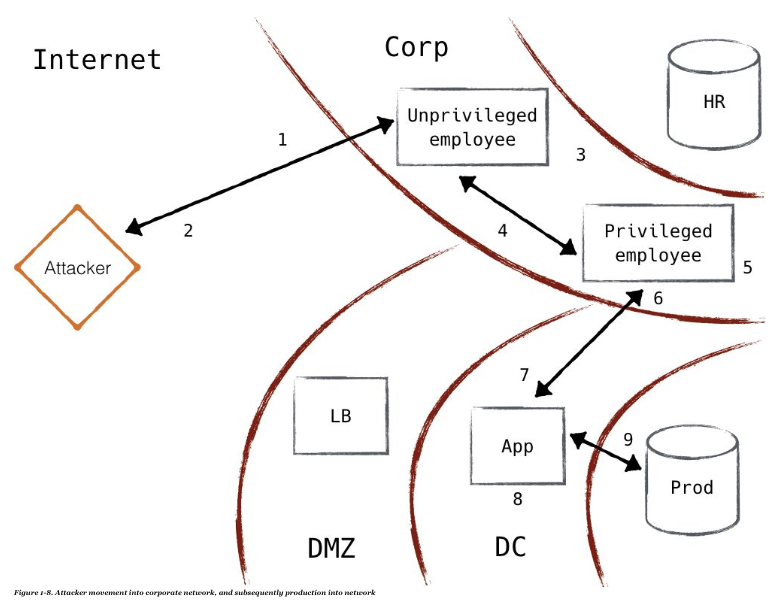
#### Abstract Concepts

Organizational networks differ in many ways from the first network conceived, created and used by the government and academia during the Cold War. While there are differing stories surrounding the who and how the network came to be, we do know with some sense of certainty what was built. The identification system was modeled after physical addresses ( like one would have on a house ), and the protocols to establish communication were modeled after a pattern of call, acknowledgment, and response. Trust was assumed because as we said, in the beginning, the people using the network knew each other. However, there another important assumption, people who were not known did not have the address of the members in the network. Yes, I know this is security by obscurity, and it can't last for long, but it does give us some insight into the mindset. They weren't worried about “unwanted calls” because only people they knew had their “phone number”. When organizations began to open up a part of their private networks to public traffic network administrators began to create “roadblocks” for outsiders. This type of security is called “Perimeter Security.”

*“Even though the perimeter security model still stands as the most prevalent model by far, it is increasingly obvious that the way we rely on it is flawed. Complex (and successful) attacks against networks with perfectly good perimeter security occur every day. An attacker drops a remote access tool (or RAT) into your network through one of a myriad of methods, gains remote access, and begins moving laterally. Perimeter firewalls have become the functional equivalent of building a wall around a city to keep out the spies.” [[1]](#footnote-0)*

**For example :**

1. *Employees targeted via phishing email*
2. *Corporate machine compromised, shell shoveled*
3. *Lateral movement through corporate network*
4. *Privileged workstation located*
5. *Local privilege escalation on workstation — keylogger installed*
6. *Developer password stolen*
7. *Compromised prod app host from privileged workstation*
8. *Developer password used to elevate privileges on prod app host*
9. *Database credentials stolen from app*
10. *Database contents exfiltrated via compromised app host*

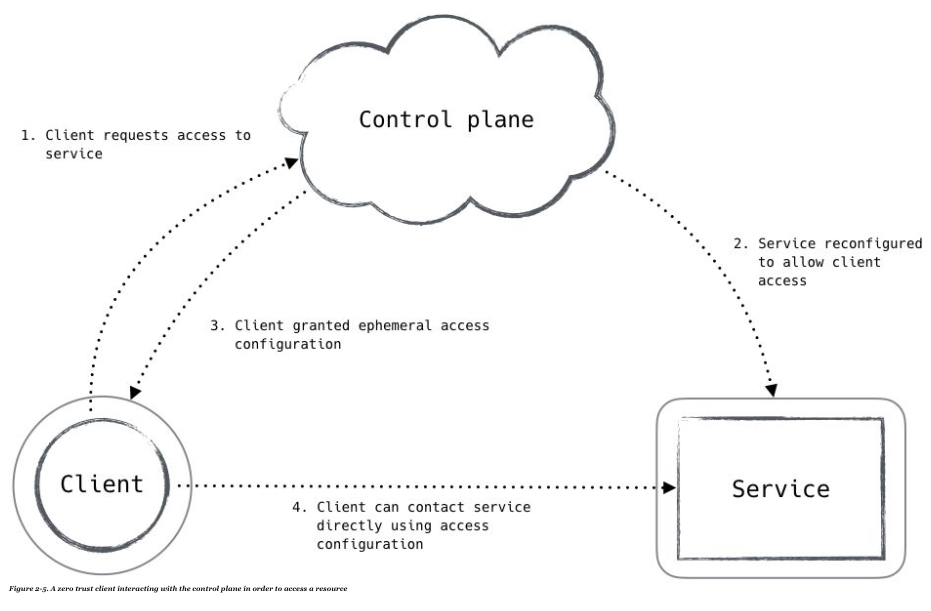


*“The perimeter and zero trust models are fundamentally different from each other. The perimeter model attempts to build a wall between trusted and untrusted resources (i.e., local network and the internet). On the other hand, the zero trust model basically throws the towel in, and accepts the reality that the “bad guys” are everywhere. Rather than build walls to protect the soft bodies inside, it turns the entire population into a militia.”*

So now we know what we need to do. We need to authenticate and analyze the identity of everything and everyone who accesses resources in the network before they access it and while they are accessing it. This obviously cannot be done manually and requires some type of automation.

#### Creating a Zero Trust Network: Control Plane vs. Data Plane

Creating a Zero Trust Network first involves properly segmenting network resources, defining their access and scope. For example, a machine that is running a web server should not have privileges to authorize “anything and everything” on a network because it is inside a network. We must always assume the worst, that this machine may become compromised, what harm could it do? Additionally, if we define its access according to its task, it only needs access to the Application Server and not a direct access to the Database Server. Doing this one time for a pair of servers is easy, doing it for hundreds of applications and users involves creating a “control plane” that grants access to a “data plane”.



*“The data plane in such a network is made up of the applications, firewalls, proxies, and routers that directly process all traffic on the network. These systems, being in the path of all connections, need to quickly make a determination of whether traffic should be allowed. When viewing the data plane as a whole, it has broad access and exposure throughout the system, so it is important that the services on the data plane cannot be used to gain privilege in the control plane and thereby move laterally within the network.”*

*Furthermore,*

*“The control plane in a zero trust network is made up of components that receive and process requests from data plane devices that wish to access (or grant access to) network resources, as shown in Figure 2-5. These components will inspect data about the requesting system to make a determination on how risky the action is, and examine relevant policy to determine how much trust is required. Once a determination is made, the data plane systems are signaled or reconfigured to grant the requested access.”*

The sophistication of the Control Plane will determine the relative speed, performance, and accuracy of a network in terms of the way it grants access, protects itself from attacks, and avoids false positives. For example, in a more sophisticated model a control plane agent will analyze the log of a server that is requesting access before granting access.

#### Creating a Trust Engine inside an SDN with MTD

Since we are now operating in a network environment that does not assume trust based on location ( IP address ), role or authentication, and we need to continually calculate trust. This type of calculation can be accomplished with a Trust Engine.

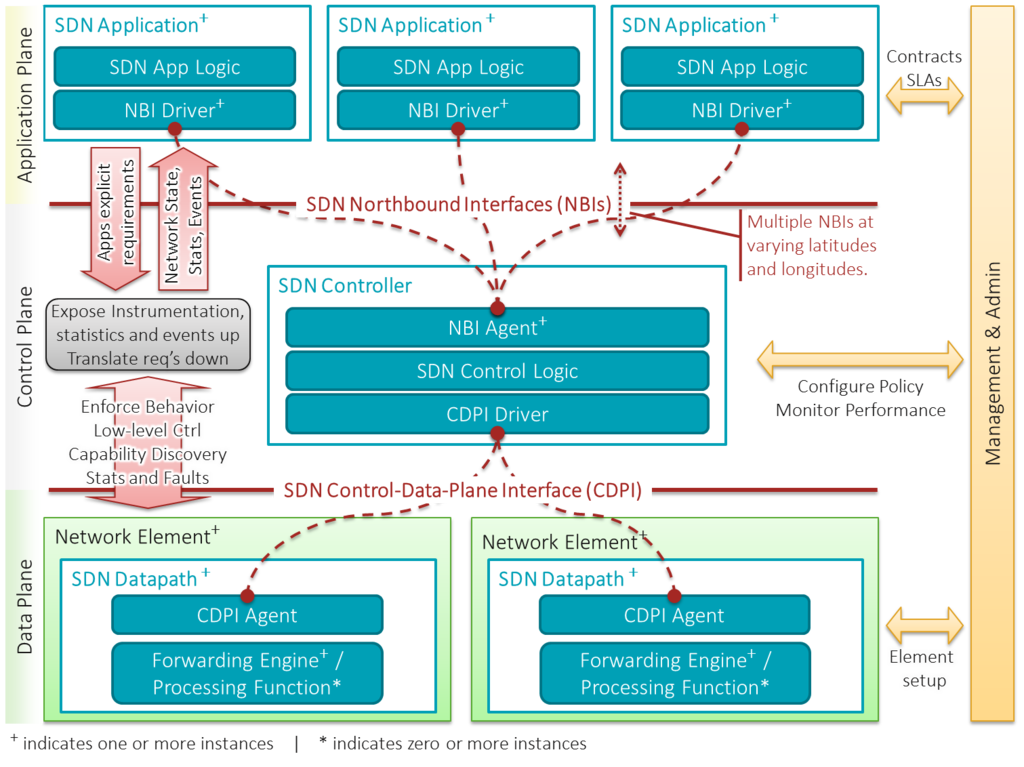
*“The trust engine is the system in a zero trust network that performs risk analysis against a particular request or action. This system’s responsibility is to produce a numeric assessment of the riskiness of allowing a particular request/action, which the policy engine uses to make an ultimate authorization decision.” [[2]](#footnote-1)*  
  
Trust engines can be implemented in several different ways, through software or as a part of a hard appliance. One of the more current/popular approaches is to deploy a trust engine as a part of a Software Defined Network or SDN. While this is not the solution for all of the issues presented in our modern world of SAAS, Cloud and Remote employees, it does provide a framework in which to use a Trust Engine. Furthermore, the SD-WAN can implement and manage components like Web Application Firewalls, firewalls that are configured specifically for an application, rather than the “one point of failure” firewall in the traditional perimeter model.

*“SDN suggests to centralize network intelligence in one network component by disassociating the forwarding process of network packets (data plane) from the routing process (control plane). The control plane consists of one or more controllers which are considered as the brain of SDN network where the whole intelligence is incorporated.”* [[3]](#footnote-2)

The SDN gives the Trust Engine a home and more importantly allows the Network Admins to define automated rules for scoring requirements.

One of the more advanced ways of protecting networks involves the development of Moving Target Defense or MTD. MTD is used to confuse and frustrate the attacker by hiding and changing the presence of ports and keys, very similar to a honeypot merged implemented like a wack-a-mole game.

*“One application can for example periodically assign virtual IPs to hosts within the network, and the mapping virtual IP/real IP is then performed by the controller. Another application can simulate some fake opened/closed/filtered ports on random hosts in the network in order to add significant noise during the reconnaissance phase (e.g. scanning) performed by an attacker.”[[4]](#footnote-3)*



#### Summary

In summary, this approach to network design is antithetical to the concepts of the traditional networks and introduces a number of new features, components, and concepts that make it safer but more complex. The benefits however include a foil to a lot of common attacks, maybe not against a state actor with unlimited resources, but surely against most SKIDs and common criminals.

**Port Scanning:** These attacks are foiled with the use of Zero Trust and MTD. The servers with open ports do not respond to just respond to anyone or any request. Furthermore, they may respond with false information if the agent's authenticity cannot be verified.

**Man-In-The-Middle Attacks:** Since authentication is required to establish communication MiTM attacks are foiled when the attacker is asked for credentials to prove their identity.

**Traversing the Network:** If an attacker is able to penetrate the firewall and they are on the system, their first order of business to find another device in the system within proximity of the information they want access to. Again the attacker will be foiled since looking at resources inside of a network will require authentication especially if the resources are not normally requested by the user the attacker has assumed the identity of.

**Phoning Home:** In traditional networks, outbound traffic is pretty much unrestricted. In a Zero Trust Network, this type of communication with an malicious IP, target or user is seen by the Trust Engine and flagged ( at the very least ).

### Troubleshooting

Troubleshooting an SDN takes some knowledge and sophisticated tools. However, most of the issues will fall into one of the following buckets according to Microsoft [[5]](#footnote-4) :  
  
**Invalid configuration**  
A user invokes the NorthBound API incorrectly or with invalid policy.  
  
**Error in policy application**  
Policy from Network Controller was not delivered to a Hyper-V Host, significantly delayed and/or not up to date on all Hyper-V hosts (for example, after a Live Migration).  
  
**Configuration drift or software bug**  
Data-path issues resulting in dropped packets.  
  
**External error related to NIC hardware/drivers or the underlay network fabric**  
Misbehaving task offloads (such as VMQ) or underlying network fabric misconfigured (such as MTU)   
  
Furthermore it can be said that SDN succeeds in solving an orchestration problem but fails to address a management issue. If we understand orchestration to be the provisioning of permissions, whereas management relates to the ongoing maintenance of these permissions with the appropriate policies. This is why ongoing curation is still the most important part of any system. When we stop learning about the types of activities that are happening in a system, we hinder our ability to address problems dynamically.

“Regardless of the ultimate solution, it must address both management and orchestration, and enable a more centralized means of doing so. That means both operations and administration must be considered, lest we shift the burden of manual operations to technology and free up engineers and architects to innovate only to realize we actually can’t, because they're still stuck managing devices manually, one by one.” [[6]](#footnote-5)

## Section 2

### Rationale

The rationale for the shift away from the perimeter defense strategy has been discussed throughout this plan. We understand clearly why early networks did not need much security because everyone knew each other. We have now shifted to a time where the access to devices and information through a much larger network, the internet, is ubiquitous. The tiny village of users is now a teeming city of strangers. Trying to lock everything up hinders our ability to perform. Using the old ways of securing ourselves leaves us open to attack. We have to think differently about the way we interact with everything, so we can re-invent protocols and restructure networks. Failing to do so will leave us with a horde angry users and very little trust in technology. By assuming the worst ( which may sound bad, but has great utility ) and making authentication and encryption the defacto standard we are extending the dream of the internet’s founders, by creating a process to trust to exist even in the absence of a personal relationship. In fact, when security and authenticity is the norm, trust will also be the norm, thereby creating a path for technology to facilitate personal relationships where none existed before.

### Note

Implementing SDN is not a trivial task even though it may be advertised as such. A search for SD-WANs or SDN with instantly produce a list of service providers and manufacturers promising to protect your network within 5 minutes. Don’t be fooled! The 5-minute promise includes implementation of a template that may or may not be applicable to the network we are trying to protect. Implementing the boilerplate controls is much better than nothing, but almost guarantees some loss of usability. Fine tuning the policies in a SDN is the work of a seasoned security experts, diligently trying to find better examples of bad behavior that may be invisible to the algorithms or validating good behavior that was falsely flagged. In summary there is no silver bullet. Creating trust in human relationships is delicate and complex, computer networks are equally as difficult to define, implement and maintain.

It should also be noted that the success of a Zero Trust Network should not be measured by simply tracking the number of breaches. Breaches will happen. The point of a Zero Trust Network is “minimization of the blast radius”, meaning an attacker may breach the system, but will ultimately not be able to continue to abuse the network by compromising all the “walls”. Eventually the attacker will come across a process or a target that asks for authentication or challenges the attackers access. In traditional systems, once trust is given it is never challenged again. In a Zero Trust Network trust is never permanently granted, so getting away with steal some credentials doesn’t guarantee an unlimited free pass.

### Glossary

**SDN Application**  
SDN Applications are programs that explicitly, directly, and programmatically communicate their network requirements and desired network behavior to the SDN Controller via a northbound interface (NBI). In addition, they may consume an abstracted view of the network for their internal decision-making purposes. An SDN Application consists of one SDN Application Logic and one or more NBI Drivers. SDN Applications may themselves expose another layer of abstracted network control, thus offering one or more higher-level NBIs through respective NBI agents.

**SDN Controller**  
The SDN Controller is a logically centralized entity in charge of (i) translating the requirements from the SDN Application layer down to the SDN Datapaths and (ii) providing the SDN Applications with an abstract view of the network (which may include statistics and events). An SDN Controller consists of one or more NBI Agents, the SDN Control Logic, and the Control to Data-Plane Interface (CDPI) driver. Definition as a logically centralized entity neither prescribes nor precludes implementation details such as the federation of multiple controllers, the hierarchical connection of controllers, communication interfaces between controllers, nor virtualization or slicing of network resources.

**SDN Datapath**  
The SDN Datapath is a logical network device that exposes visibility and uncontested control over its advertised forwarding and data processing capabilities. The logical representation may encompass all or a subset of the physical substrate resources. An SDN Datapath comprises a CDPI agent and a set of one or more traffic forwarding engines and zero or more traffic processing functions. These engines and functions may include simple forwarding between the datapath's external interfaces or internal traffic processing or termination functions. One or more SDN Datapaths may be contained in a single (physical) network element—an integrated physical combination of communications resources managed as a unit. An SDN Datapath may also be defined across multiple physical network elements. This logical definition neither prescribes nor precludes implementation details such as the logical to physical mapping, management of shared physical resources, virtualization or slicing of the SDN Datapath, interoperability with non-SDN networking, nor the data processing functionality, which can include OSI layer 4-7 functions.

**SDN Control to Data-Plane Interface (CDPI)**  
The SDN CDPI is the interface defined between an SDN Controller and an SDN Datapath, which provides at least (i) programmatic control of all forwarding operations, (ii) capabilities advertisement, (iii) statistics reporting, and (iv) event notification. One value of SDN lies in the expectation that the CDPI is implemented in an open, vendor-neutral and interoperable way.

**SDN Northbound Interfaces (NBI)**  
SDN NBIs are interfaces between SDN Applications and SDN Controllers and typically provide abstract network views and enable direct expression of network behavior and requirements. This may occur at any level of abstraction (latitude) and across different sets of functionality (longitude). One value of SDN lies in the expectation that these interfaces are implemented in an open, vendor-neutral and interoperable way.

**Phoning Home**

(also called Phone Home or Call Home), in computing, refers to an act of client to server communication which may be undesirable to the user and/or proprietor of the device or software. It is often used to refer to the behavior of security systems which report network location, username, or other such data to another computer.[[7]](#footnote-6)  
  
Phoning home may be useful for the proprietor in tracking a missing or stolen computer. This type of phoning home is frequently used on mobile computers at corporations. It typically involves a software agent which is difficult to detect or remove. However, there are malicious types of “phoning home” such as surreptitious communication between applications or hardware installed at end-user sites and their manufacturers or developers. The traffic may be encrypted to make it difficult or impractical for the end-user to determine what data are being transmitted.[[8]](#footnote-7)  
  
The Stuxnet attack on Iran's nuclear facilities was facilitated by phone home technology as reported by The New York Times.[[9]](#footnote-8)

**Network address translator traversal**

is a computer networking technique of establishing and maintaining Internet protocol connections across gateways that implement network address translation.

The following NAT traversal techniques are available:  
  
**Socket Secure (SOCKS)** is a technology created in the early 1990s that uses proxy servers to relay traffic between networks or systems.

**Traversal Using Relays around NAT (TURN)** is a relay protocol designed specifically for NAT traversal.

**NAT hole punching** is a general technique that exploits how NATs handle some protocols (for example, UDP, TCP, or ICMP) to allow previously blocked packets through the NAT.[[10]](#footnote-9)

**Session Traversal Utilities for NAT (STUN)** is a standardized set of methods and a network protocol for NAT hole punching. It was designed for UDP but was also extended to TCP.

**Interactive Connectivity Establishment (ICE)** is a complete protocol for using STUN and/or TURN to do NAT traversal while picking the best network route available. It fills in some of the missing pieces and deficiencies that were not mentioned by STUN specification.

**UPnP Internet Gateway Device Protocol (IGDP)** is supported by many small NAT gateways in home or small office settings. It allows a device on a network to ask the router to open a port.

**NAT-PMP** is a protocol introduced by Apple as an alternative to IGDP.

**PCP** is a successor of NAT-PMP.

**Application-level gateway (ALG)** is a component of a firewall or NAT that allows for configuring NAT traversal filters. It is claimed by numerous people that this technique creates more problems than it solves.[[11]](#footnote-10)

1. Gilman, Evan, and Doug Barth. Zero Trust Networks: Building Secure Systems in Untrusted Networks. OReilly, 2017. [↑](#footnote-ref-0)
2. Gilman, Evan, and Doug Barth. Zero Trust Networks: Building Secure Systems in Untrusted Networks. OReilly, 2017. [↑](#footnote-ref-1)
3. Benzekki Kamal et al.Devolving IEEE 802.1 X authentication capability to data plane in software‐defined networking (SDN) architecture., Security and Communication Networks 9.17 (2016): 4369-4377. [↑](#footnote-ref-2)
4. Kampanakis, Panos; Perros, Harry; Beyene, Tsegereda. SDN-based solutions for Moving Target Defense network protection (PDF). Retrieved 23 July 2014. [↑](#footnote-ref-3)
5. “Troubleshoot the Windows Server Software Defined Networking Stack.” Windows IT Pro Center, Microsoft, 13 Aug. 2018, docs.microsoft.com/en-us/windows-server/networking/sdn/troubleshoot/troubleshoot-windows-server-software-defined-networking-stack. [↑](#footnote-ref-4)
6. MacVittie, Lori. “The Problem With SDN.” Network Computing, IDC, 15 Jan. 2016, www.networkcomputing.com/networking/problem-sdn/1791956585. [↑](#footnote-ref-5)
7. Technology Meetings Website - http://technologymeetings.com/ar/meetings\_catch\_laptop\_thief/index.htm [↑](#footnote-ref-6)
8. ZoneAlarm phones home, Apple throws Intel a bone "Archived copy". Archived from the original on 2006-02-06. Retrieved 2006-03-03. [↑](#footnote-ref-7)
9. DAVID E. SANGER; THOM SHANKER (Jan 14, 2014). "N.S.A. Devises Radio Pathway Into Computers". The New York Times. Retrieved 27 May 2014. [↑](#footnote-ref-8)
10. Schmidt, Jürgen. “The Hole Trick.” How Skype & Co. Get Round Firewalls, Rovi Corporation, 15 Dec. 2006, web.archive.org/web/20110810233331/http://www.h-online.com/security/features/How-Skype-Co-get-round-firewalls-747197.html. [↑](#footnote-ref-9)
11. "NAT Traversal Techniques and Peer-to-Peer Applications". Helsinki University of Technology. CiteSeerX 10.1.1.103.1659. [↑](#footnote-ref-10)