Global Information Assurance Certification Paper

Incident Response in a Zero Trust World

GIAC GCIA Gold

Author: heath.lawson@student.sans.edu
Advisor: *Lenny Zeltser*

Accepted: *January 15th, 2020*

Abstract

Zero Trust Networks is a new security model that enables organizations to provide continuously verified access to assets and are becoming more common as organizations adopt cloud resources (Rose, S., Borchert, O., Mitchell, S., & Connelly, S., 2019). This new model enables organizations to achieve much tighter control over access to their resources by using a variety of signals that provide great insight to validate access requests. As this approach is increasingly adopted, incident responders must understand how Zero Trust Networks can enhance their existing processes. This paper provides a comparison of incident response capabilities in Zero Trust Networks compared to traditional perimeter-centric models, and guidance for incident responders tasked with managing incidents using this new paradigm.

1. Introduction

A perfect storm of conditions is setting the stage for organizations to adopt new models for securing their resources. As cloud services become more ubiquitous for even the most critical of business functions, organizations realize they must extend their security boundaries outside of their traditional network perimeters. Additionally, the prevalence of connectivity and variety of devices, as well as a surge in the workforce for always-on, always available resources, changes how and where employees perform work duties.

All of these factors contribute to a new set of requirements for the way enterprises approach securing their assets. This new approach is popularly known as Zero Trust Networking, or Zero Trust Architectures, which focuses on protecting resources rather than network segments, as is common today (Rose, S., Borchert, O., Mitchell, S., & Connelly, S., 2019).

At the same time, incident responders and enterprise defenders face an increasingly hostile threat landscape, with more determined and advanced adversaries than ever before. Guidance for defending and responding to incidents in traditional networks is well proven, but when coupled with Zero Trust models, gaps are exposed in many incident response guidelines available today.

As a result, the nexus of these facts presents a glaring question:

Do concepts of Zero Trust Networks enable Incident Responders to be as effective, or even more effective, when used in conjunction with cloud services?

This research aims to answer that question through an analysis of common cloud security incidents viewed through the lenses of network-perimeter security and Zero Trust Network architectures.

Heath Lawson, heath.lawson@student.sans.edu

2. Incident Response

A computer security incident is a series of observable events that collectively form an activity with potentially negative consequences on the confidentiality, integrity, or availability of systems or data assets in an organization (SANS, 2019). The National Institute of Standards and Time further defines a computer security incident as "a violation or imminent threat of violation of computer security policies, acceptable use policies, or standard security practices" (Cichonski, P., Millar, T., Grance, T., & Scarfone, K., 2012).

Incident response, also known as incident handling, is the process by which a computer security incident is managed using an established model (Cichonski, P., Millar, T., Grance, T., & Scarfone, K., 2012). Like many operational aspects of information security, these models usually follow a lifecycle from the beginning of the incident to the remediation and closure of the incident.

A very prevalent process for incident response is provided by SANS, commonly known by the acronym PICERL with six key phases (SANS, 2019):

- In the **preparation phase,** an organization builds the written policies, acquires the necessary materials and resources needed, and prepares to respond to an incident.
- In the **identification phase**, the organization identifies the scope and severity of an incident, and a response is put into motion.
- Once the organization identifies the incident, they move to the **containment phase** and take steps to prevent further movement or damage by the attacker.
- Once the organization contains the incident, the organization begins the **eradication phase** to remove any traces of the attacker from the targeted systems.

- In the **recovery phase**, affected systems and data are verified and returned to regular service.
- Finally, in the **lessons learned phase**, the organization extracts insights and opportunities for improvement from the incident and feed this information to the preparation phase to complete the cycle.

3. Network-based security model

Enterprises usually adopt some variation of a common security architecture, centered on a minimum of three network zones, including the Internet, DMZ, and Intranet or Private networks (Scarfone, K., & Hoffman, P., 2009). Hosts are grouped by purpose and sensitivity and assigned to a zone. Each zone carries a different level of trust, with hosts on the public network trusted less than the DMZ, and the DMZ zone trusted less than the Private zone.

To keep up with current threats, and best practices in building securable networks, many security defenses have generally found a home at each "choke point" on the network to ensure adequate coverage. In this network-centric model, security defense like Intrusion Detection/Intrusion Prevention Systems, Data Loss Prevention tools, and web proxies operate at the network borders, which means any activities that must cross these network boundaries can be monitored, inspected, and secured (See Figure 1).

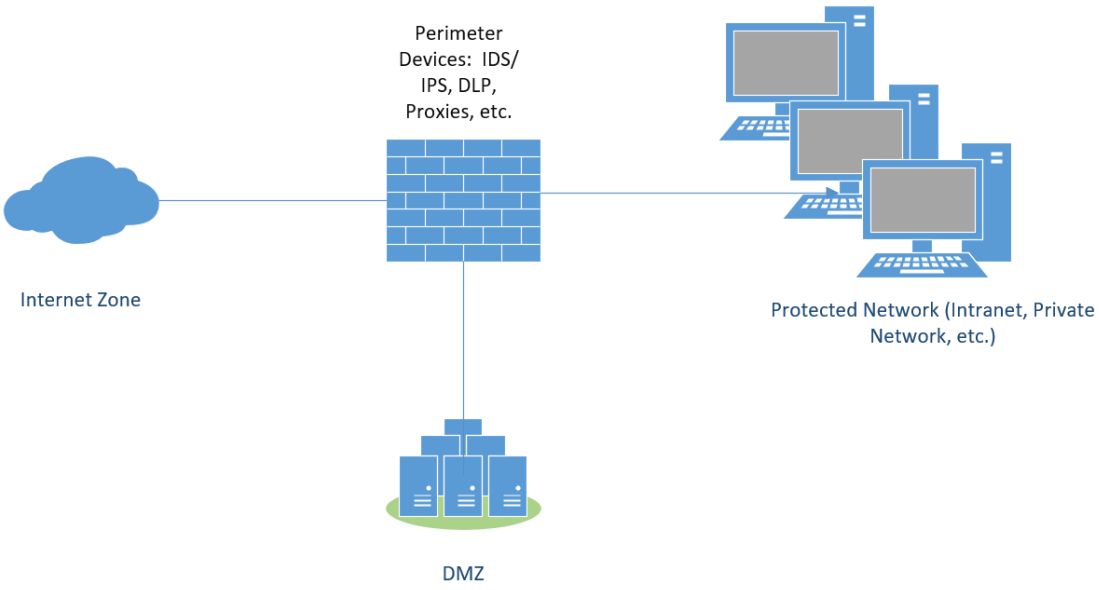

Figure 1. Common network layout with security defenses

There are multiple potential downsides to this approach, however. The first is that there is an implicit trust that any device connected behind the perimeter security defenses is secure to the level of that zone. When applied to conventional enterprise networks, this means that a compromised endpoint doesn't have to traverse much in the way of network defenses, and an attacker could move undetected after the initial compromise.

Another fundamental limitation is that traffic must always pass through these network perimeters to be protected. In the world of highly mobile users and cloud services, this can present numerous challenges for the user experience. Organizations frequently mitigate this limitation by using technologies like Virtual Private Networks (VPN) to bring all traffic back to the secured network then to be routed out through the perimeter controls. However, this approach creates other challenges, including additional complexity and potentially higher latency, and can introduce privacy concerns on non-organizational owned devices in scenarios like Bring Your Own Device (BYOD).

4. Zero Trust Networks

Given the challenges of traditional network defenses in a modern computing environment, Zero Trust Networks are gaining prevalence in corporate networks. To enable access from any device, anywhere under a variety of conditions, this new model must ensure that only authorized parties have access to resources, but we must be more granular. In the era of containers, infrastructure-as-code, and billions of devices, we can no longer just rely on the network to give us the control we need. Instead, this new model is needed to solve the problem – one that uses a consistent control plane across all users, devices, apps, and the data they touch.

Zero Trust Networks, also referred to as Zero Trust Architectures, breaks the broad implicit trust highlighted in the previous section, by enabling tighter control over resource access. At its core, it ensures that every access attempt is verified and uses all available data to validate it's a legitimate request.

To better describe Zero Trust in the context of incident response, it can be best distilled to the following principles:

- **Assume the corporate network (and as a result, the perimeter defenses and internal occupants) cannot be trusted.** This principle runs counter to the perimeter-focused approach, where anything behind security devices is inherently trusted and considered safe. Instead, focus on the idea of breach containment and limiting damage from an incident (Rose, S., Borchert, O., Mitchell, S., & Connelly, S., 2019).
- **Identity, Device, Application, and Data insight is required.** Historically, many of these assets would take a backseat to network-based detections. In assuming the network can't be trusted, organizations are left with these four common factors to inspect for every transaction (Rose, S., Borchert, O., Mitchell, S., & Connelly, S., 2019).

- **Every resource access attempt must be validated.** Rather than inherently trusting that a user and device should have access because they are on a "secure network", verify using all available signals above, that it is a legitimate request (Rose, S., Borchert, O., Mitchell, S., &Connelly, S., 2019).
- **Automated response is critical.** In the current threat landscape, automated detection and remediation is the only way we can analyze enough data, and potentially respond quickly enough, to stand a chance of catching and stopping advanced adversaries in time (Rose, S., Borchert, O., Mitchell, S., & Connelly, S., 2019).

With the principles of Zero Trust identified, the logical components can be explained (Figure 2). In the NIST model, the idea of a *Policy Decision Point (PDP)* is the nexus of activity where administrator desire is enforced over resource access. This might sound like a perimeter, but it is fundamentally different from the broad perimeters the industry is most familiar with. Instead, think of it is as a secure enclave per resource – one that we can control to incredibly concise requirements. Within that enclave, we can apply a granular policy to enforce least privilege across our resources.

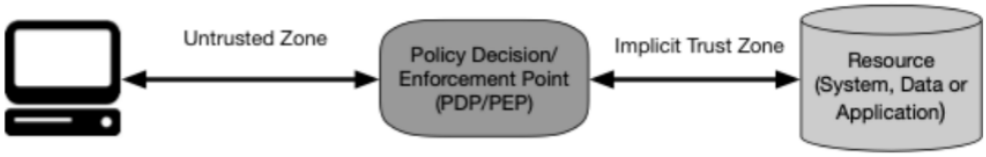

Figure 1: Zero Trust Access

Figure 2. Conceptual model of Zero Trust Access (NIST, 2019)

In terms of the policies and subsequent decision requirements, the next asset of Zero Trust focuses on minimizing unauthorized access. With a strong control plane of a minimal trust zone and robust enforcement point (PDP), signals can be combined to ensure access request decisions are made with the best data available. This means

Zero Trust systems inherently bring together sources like the risk-based models of user behavior, device health, data classifications, and even compliance boundaries to ensure resources accessed in the right way, under the right conditions, for the right reasons (Rose, S., Borchert, O., Mitchell, S., & Connelly, S., 2019).

In a zero trust world, there will still be security incidents. There is probably no amount of technology that can magically remove the threat of a determined adversary gaining access to systems. Instead, it's important to consider that the cornerstone to Zero Trust Networks, this idea of shrinking trust zones, also can result in the idea of shrinking the involvement of other resources for any single incident. This means when an incident happens, the smaller trust zone reduces the widespread risk to the other systems. By doing this, we can also reduce the delay in detection and make incident response more efficient for everyone.

Finally, it's important to note only recently have standards bodies like NIST started providing guidance on Zero Trust Networks, like in the NIST draft Special Publication, 800-207. This is important because it shows that there is still much variance in what is popularly defined as Zero Trust. Additionally, many security vendors have latched on to the tag line of "zero trust" and used it to market their products. To be clear, Zero Trust is still in its early stages and has more growing to do, though there are meaningful steps everyone can be taking today.

5. The Experiment

An experiment has been devised to capture data from representative examples of each environment, traditional and Zero Trust, with a controlled series of incidents to quantitatively compare them.

The test process is comprised of four scenarios to simulate common real-world incidents when using cloud services.

- **Use of an unsanctioned cloud service** is simulated by uploading a test folder of data containing Microsoft Word documents to a consumer cloud service.

- **Compromised user credentials** account for nearly 29% of security incidents (Verizon, 2019), and is simulated by using legitimate user credentials that could be obtained through phishing or social engineering to access company services. In this scenario, the adversarial actions will be simulated outside of the corporate network.
- **Suspicious use of mailbox forwarding** is a common post-exploit technique for data exfiltration or to cover further efforts by an adversary (MITRE, 2017). This scenario is simulated by creating mailbox rules to forward mail to external domains delete from automatically delete from sent items.
- **Inadvertent file oversharing** is simulated by sharing a sensitive file from a cloud storage service with external recipients. This scenario takes on many forms, though is commonly an accident on behalf of the user.

In order to quantitatively evaluate the outcomes, a scoring model is used to evaluate the environment against the identification and containment phases of PICERL. These two phases were selected to simplify testing, and because the detection and initial response are foundational to subsequent phases of incident response. In other words, if one of the environments is unable to detect the incident or is unable to take any action to contain the incident, the following phases are less effective as a measure. The scoring is based on the following criteria:

Score	Outcome
0	Was not able to complete the objective
3	Was able to complete the objective partially. Further work is required to move to the next phases of incident response.
5	Able to complete the phase of incident response.

In addition to the scoring, the pros and cons of each environment are captured in a matrix show below:

Phase	**Zero Trust Architecture**	**Perimeter-based Architecture**
Identify	Pros: Cons: Score	Pros: Cons: Score
Contain	Pros: Cons: Score	Pros: Cons: Score

5.1. Network-based security environment

A simple network consisting of a Windows 10 client PC and a pFSense firewall served as the test bed for evaluating incident identification and containment in network-based security environments (Figure 3).

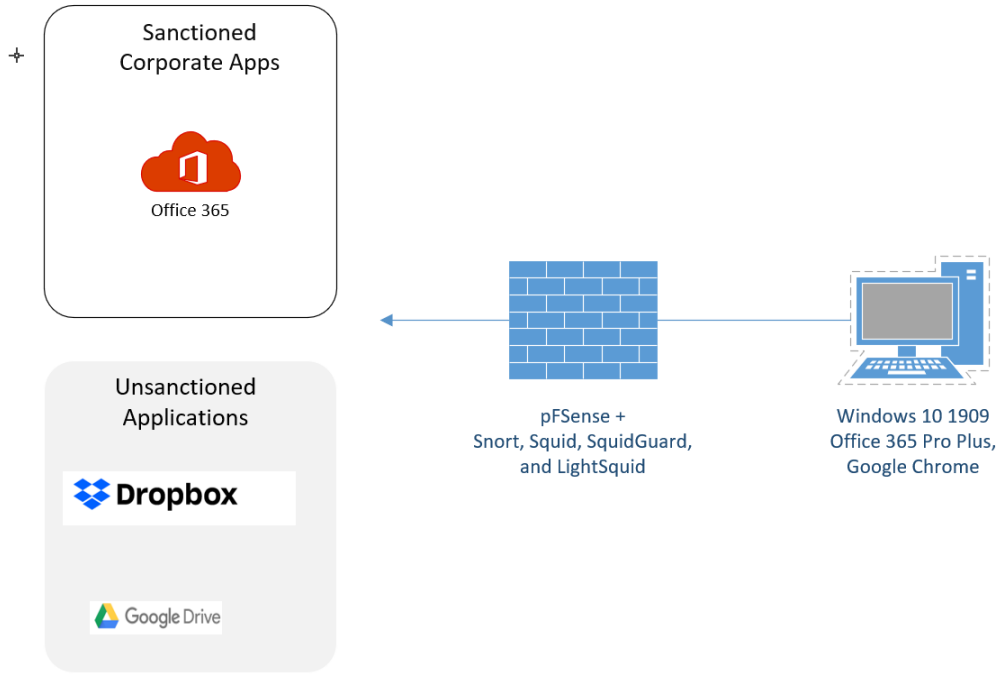

Figure 3. Network diagram of perimeter-based security environment

The Windows 10 1909 client device was running on VMware Fusion (hostname "Win10Trad") as a virtual machine, and had Office 365 ProPlus installed, along with all applicable updates. This served as the device from which a simulated user or attacker would perform their actions.

A pFSense firewall was configured, also running on VMware Fusion Pro, hostname FW01. Snort was installed, along with the OpenAppID rules, to provide context on SaaS software usage. Snort operated with a default configuration, with the addition of enabling the appropriate OpenAppID rule categories and enabling IPS mode with the Security setting.

Squid was also installed on the PFSense firewall VM to serve as a proxy for visibility into user activities and was also configured to perform SSL inspection. SquidGuard was also installed to provide URL filtering for specific application URLs seen in Figure 4 below.

Installed Packages

	Name	Category	Version	Description	Actions
✔	Lightsquid	www	3.0.6_6	LightSquid is a high performance web proxy reporting tool. Includes proxy realtime statistics (SQStat). Requires Squid package. Package Dependencies: lighttpd-1.4.51 lightsquid-1.8_5	
✔	snort	security	3.2.9.10	Snort is an open source network intrusion prevention and detection system (IDS/IPS). Combining the benefits of signature, protocol, and anomaly-based inspection. Package Dependencies: snort-2.9.15 barnyard2-1.13_1	
✔	squid	www	0.4.44_9	High performance web proxy cache (3.5 branch). It combines Squid as a proxy server with its capabilities of acting as a HTTP / HTTPS reverse proxy. It includes an Exchange-Web-Access (OWA) Assistant, SSL filtering and antivirus integration via C-ICAP. Package Dependencies: squidclamav-6.16 squid_radius_auth-1.10 squid-3.5.27_3 c-icap-modules-0.5.3_1	
✔	squidGuard	www	1.16.18_3	High performance web proxy URL filter. Package Dependencies: squidguard-1.4_15	

Figure 4 Packages installed on pFSense firewall.

5.2 Zero Trust Network environment

In keeping with the simplicity of the network-based environment, a single PC and several cloud components served to model Zero Trust Network concepts (Figure 5).

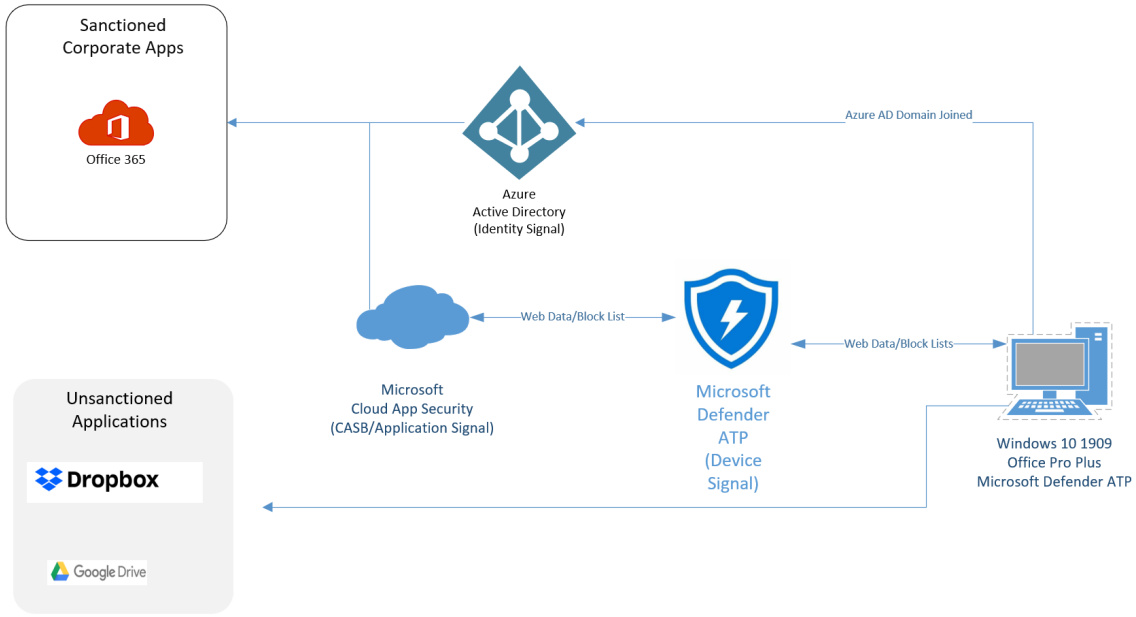

Figure 5. Zero Trust Network environment

Heath Lawson, heath.lawson@student.sans.edu

A Windows 10 1909 virtual machine running on VMware Fusion (hostname Win10-ZTN), configured with direct internet access, served as the testbed for client activities. Office 365 ProPlus was installed along with all applicable updates.

Other security tooling was enabled to provide signaling for identity, device, and applications. Microsoft Azure Active Directory (AAD) provides an identity-centric signal on user authentication, including suspicious logins. Microsoft Defender ATP (MDATP) provided signal from the client, including visibility to URLs visited and the ability to respond by block URLs on the endpoint. Finally, Microsoft Cloud App Security (MCAS) was configured to analyze cloud applications in use, and policies were deployed to look for anomalous or malicious activities.

It is important to note the components selected to model Zero Trust Networking were chosen by the author due to pre-existing familiarity with the tools. Other components provided by other vendors may offer similar capabilities.

6. Findings

6.1. Use of unsanctioned cloud application

In this scenario, a batch of documents totaling 30 megabytes were uploaded from each endpoint to a file storage application, represented as a free-tier DropBox account.

The network-based environment provided data that was useful in identifying information sent to an unsanctioned cloud service. LightSquid shows a number of transactions, with sizes, to DropBox URLs (Figure 6). While this isn't necessarily solely indicative of data leaving the network, it would warrant further investigation. In terms of containment, a competent team could move quickly to block the offending application within the proxy, and then work with the user to ensure the data is removed from the cloud service.

22	uc90c20053c67268b76efe6c6fc3.previews.dropboxusercontent.com	1	20 614	69.9 M	0.0%
23	ucf003b71ee4cdda9fe14b1aac9c.previews.dropboxusercontent.com	1	20 613	69.9 M	0.0%
24	uc34a2ea785a31eba1242302bc74.previews.dropboxusercontent.com	1	20 613	69.9 M	0.0%
25	uc8618d98892c63debe966e78dae.previews.dropboxusercontent.com	1	20 613	69.9 M	0.0%
26	uc4f5e7f628c5b15382ff688889d.previews.dropboxusercontent.com	1	20 613	70.0 M	0.0%
27	uc7d4b2a7cbe4496f40d38ddd712.previews.dropboxusercontent.com	1	20 613	70.0 M	0.0%
28	uc0e8d0ee6f496e3ce38f6d68889.previews.dropboxusercontent.com	1	18 410	70.0 M	0.0%

Figure 6. LightSquid showing data egress to DropBox

The Zero Trust Network also provided data that was useful in identifying data was sent to an unsanctioned cloud service. Fig 2. Shows the MCAS alert generated for Dropbox, and also highlights in the red box where containment/eradication activities can be initiated, by blocking the site *from the endpoint* (Figure 7). For incident responders and analysts, this means that by taking action in the CASB where the event is detected, it will enforce actions on the endpoint, regardless of where the device might be connected.

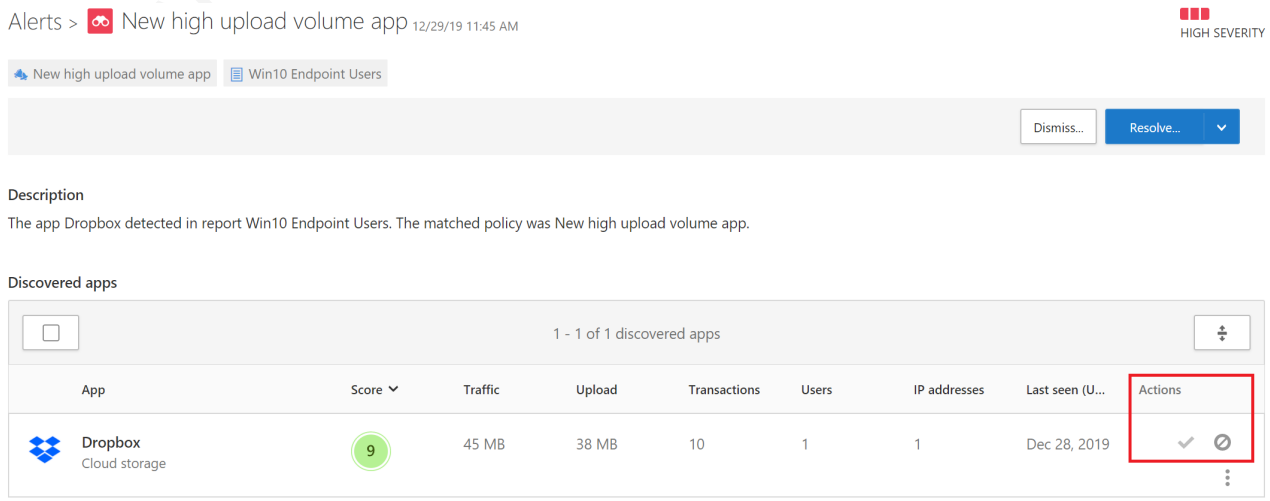

Figure 7. Microsoft Cloud App Security showing detection and options to block application

Each environment scored well in this test, with perfect scores for both. While the scores are identical, there are differences that show a clear advantage for Zero Trust Networks for both identification and containment of the incident.

Phase	**Zero Trust Network**	**Perimeter-based Architecture**
Identify	Pros: Signaling from the endpoint provided data without being on corporate network. Alert is not in real-time but highlighted the unsanctioned application quickly. Cons: Had to configure a policy to identify new high-volume applications, requires endpoint to be managed/enrolled (can be automated & enforced) Score: 5	Pros: LightSquid provided reporting to highlight the data flows, showed full URL, and was able to highlight DropBox traffic. Cons: In a busy environment, this reporting would have been difficult to interpret. Raw URLs for some cloud service providers (Amazon, Microsoft, etc.) could in fact be used by legitimate services. Also, had this endpoint not been behind perimeter, this visibility would not be available. Score: 5
Contain	Pros: Was able to quickly block the application, and ensure block was functioning regardless of network location. Cons: Blocking, in this case, was very broad. If more granular control was needed, per user or per group, the application would be sanctioned and onboarded for richer control mechanisms. Score: 5	Pros: Snort could be configured to proactively block known bad applications, and potentially allow for more granularity in the case of specific business units being allowed access applications. Cons: Block is only in effect if on the protected network, or if always-on VPN connects remote devices to the protected network. Score: 5
Total Score	10/10	10/10

6.2. Compromised user credentials

This scenario represents one of the most common incidents in cloud services, where user credentials are compromised due to phishing or other social engineering efforts. Once an adversary has the user's credentials, they will use them to access resources as the employee and continue their attack. The user's credentials were used from a Tor connection to simulate an adversary accessing resources remotely to test this scenario.

In this scenario, the perimeter-based architecture was ineffective without additional controls. Because an adversary is using credentials from an endpoint outside of the perimeter network, a lack of visibility exists. In an enterprise scenario, generally there would be additional controls not represented in this simple architecture that could have helped identify the scenario, with items like centralized logging from an identity provider or the Office 365 activity logs.

The Zero Trust Network identified the sign-in as anomalous and benefitted from rich signaling of the identity to bring visibility, and contain, eradicate, and recover the user credentials (Figure 8). In this example, the login triggered two risk alerts, based on a new location and that the attacker was originating from a Tor address. Further, the user was then prompted to change their password only after verifying their identity through multifactor authentication.

Detection time	User	IP address	Location	Detection type	Risk state
12/28/2019, 8:45:32 PM	Test User	195.206.105.217	Adliswil, Zuerich, CH	Atypical travel	At risk
12/28/2019, 8:42:30 PM	Test User	195.206.105.217	Zuerich, Zuerich, CH	Anonymous IP address	At risk

Figure 8. risky sign-ins detected.

There are many defenses commonly deployed to prevent this scenario that are purposely disabled or not deployed in the two environments. As an example, controls like multifactor authentication would prevent the use of these compromised credentials, as well as other identity-centric controls that limit logins from only healthy devices as indicated by an Endpoint Detection and Response or Mobile Device Management solution.

Phase	ZTA	Perimeter-based Architecture
Identify	Pros: Able to quickly raise an alert on anomalous login based on multiple factors. Cons: Can be prone to false negatives based on employee travel, etc. Score: 5	Unable to measure without additional capabilities deployed. Score: 0
Contain	Pros: The next resource accessed by the user prompted for verification of identity through multifactor authentication, and the user's password would be changed. Cons: Without additional work, this only applies to applications leveraging the identity provider, and may create gaps. Score: 5	Unable to measure without additional capabilities deployed. Score: 0
Total Score	10/10	0/10

6.3. Suspicious use of mailbox forwarding rules

Once an adversary gains access to a user's credentials, the next step is often to access the user's mailbox to search for sensitive information or further their cause in

compromising others (MITRE, 2019). One of the common outcomes of this activity is that the adversary will create mailbox rules to either exfiltrate emails via forwarding or delete or move messages out of the inbox to another folder.

This is difficult to detect from network inspection alone, as it requires an in-depth understanding of the APIs and other application insights that will be obscured in network traffic. As a result, the perimeter-focused network was unable to detect this behavior. Again, this is commonly mitigated by ingesting log data from the application and writing detection rules for these activities, but this also requires manual rules to be created.

The Zero Trust network implementation relied heavily on the application signal from the CASB to provide insight over suspicious behavior in the application. In this scenario, a built-in policy created an alert once mailbox forwarding rules were detected in the mailbox (Figure 9). While not configured, additional containment, eradication, and recovery steps could have disabled the user's identity or triggered an additional workflow to take other steps, such as scanning the device for malware.

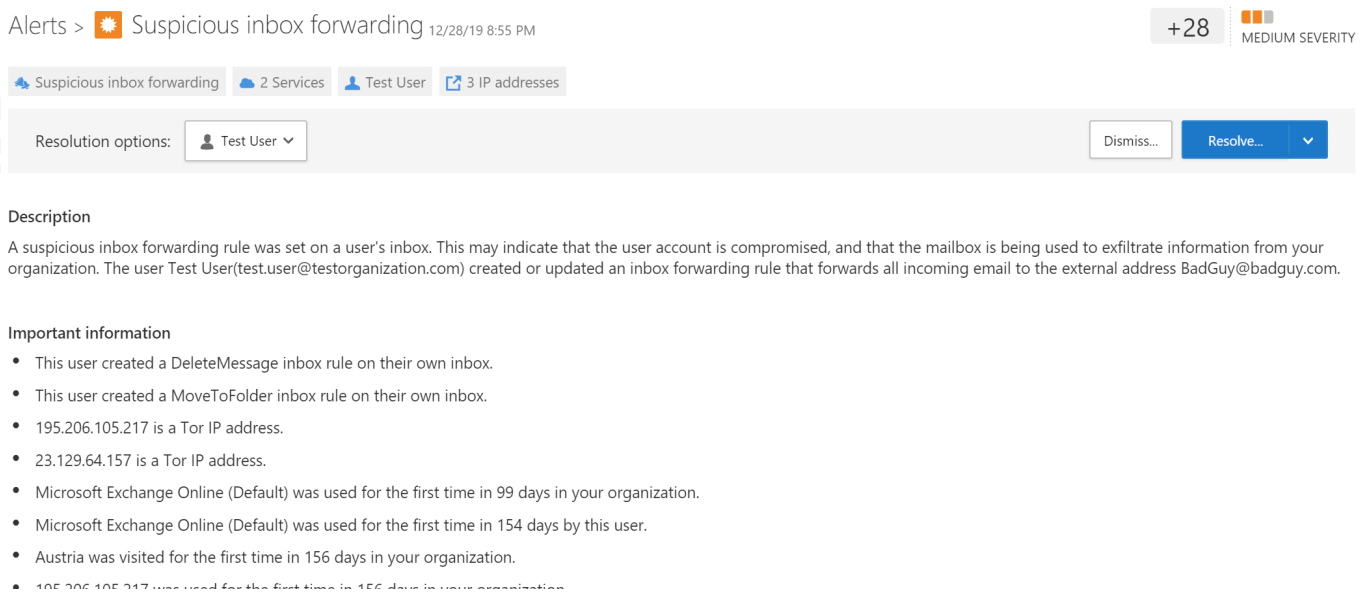

Figure 9. Alert created when suspicious mailbox rules are created.

Phase	ZTA	Perimeter-based Architecture
Identify	Pros: An alert was raised by Microsoft Cloud App Security indicating configuration of suspicious inbox forwarding. Cons: N/A Score: 5	Unable to detect without additional capabilities deployed. This may also be detectable by examining SMTP mail flows or other message journaling. Score: 0
Contain	Pros: Automatic containment, including containment of the compromised user, based on policy definition driving reset of user credentials. Cons: This containment only applies to the user in question – not downstream users affected by the adversary's messages. Score: 3	Unable to detect without additional capabilities deployed. Score: 0
Total Score	8/10	0/10

6.4. Inadvertent sharing of sensitive file by user

This scenario represents an incident not always caused by an adversary, but instead one that can be a user's mistake, though still with serious repercussions. In this scenario, a user will upload a file containing sensitive PII to a sanctioned cloud service,

but inadvertently uploads it to a folder that is shared with the public via an overly permissive sharing link. This specific example used a folder in OneDrive for Business.

The network-based security model was able to identify the traffic to OneDrive for Business, however this would also blend in with other sanctioned traffic based on URL and data flows alone. In terms of containment, eradication, and recovery, the perimeter-based controls offered no additional value in this scenario.

While the perimeter-based approach struggled to detect a very deep, application-specific incident, the zero trust model provided rich context on the scenario. In this case, the CASB identified the activity, and based on a policy defined to look for this scenario, raised an alert that cardholder information was shared with the public (Figure 10).

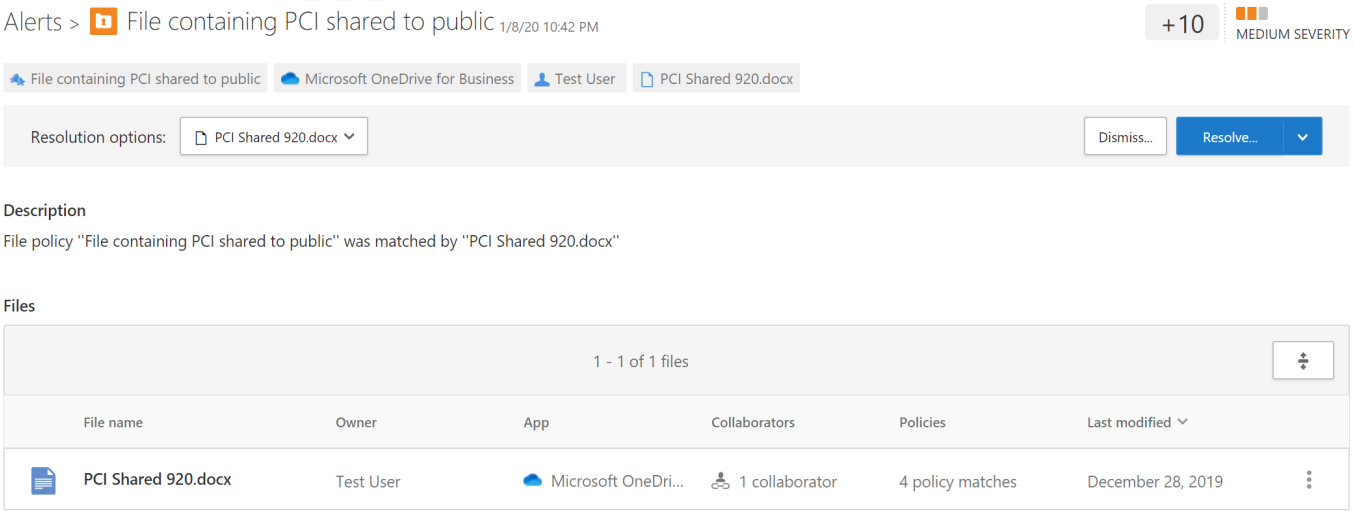

Figure 10. Alert created by MCAS when a sensitive document is shared publicly.

Additionally, in this scenario, the CASB is also able to identify if the document was accessed via the sharing link (Figure 11.). This is important for incident responders because they can use this data to understand how many individuals may have had access to the data.

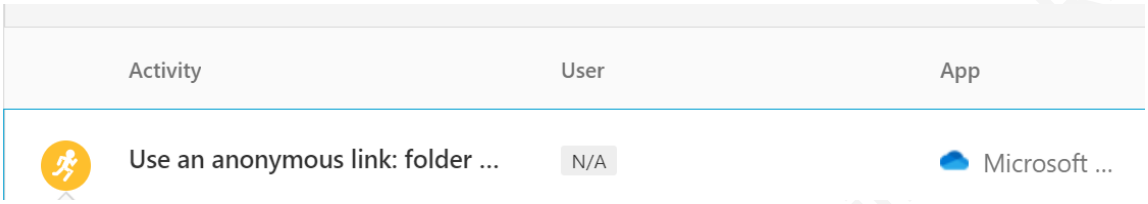

Fig 11. Activity log shows an anonymous link was accessed.

This is another example where a traditional perimeter-focused network requires additional tooling to accurately detect this incident. In this case, a feed from the Office 365 Management API could have provided this data for an alerting mechanism.

	ZTA	**Perimeter-based Architecture**
Identify	Pros: With a policy defined, MCAS was able to identify this application-level signal. Additional details are available on access attempts. Score: 5	Pros: LightSquid was able to highlight data leaving the network, and if additional intercept or DLP tools were deployed, they would have identified sensitive data. Cons: The reporting is primitive for this use case, as it centers on a URL and data flows – ideal for identifying unsanctioned applications, but not enough detail for Score: 3

Contain	Pros: While not configured for this test, MCAS has the functionality to revoke sharing, quarantine the file, apply encryption, notify the user, or start an additional workflow, etc. – all valid containment steps. Additionally, preventative real-time controls could have been deployed to prevent the upload of sensitive files to a shared location. Cons: Score: 5	Unable to contain using without additional capabilities deployed. Score: 0
Total Score	10/10	3/10

6.5 Results

Summarizing the data, it becomes apparent that Zero Trust Network models have a clear advantage when compared to the control environment. As the scenarios evolve to deeper application and endpoint context, perimeter-based security controls lose their ability to provide meaningful insight and control.

Test	Score - Network security	Score - Zero Trust Network
Exfiltration of Data to unauthorized cloud service	10	10
Compromised user credentials	0	10
Mailbox forwarding	0	8
Inadvertent sharing of	3	10

sensitive file		
Total	13	38

7. Recommendations

While these tests were conducted in simple environments, the data highlights key differences in identifying and containing incidents across different security models. Further, it especially illuminates gaps in the perimeter-based model when used with cloud services.

To answer the hypothesis presented earlier in the document, it is apparent that Zero Trust Networks equal visibility for cloud services and can provide even more benefits for incident responders.

Primarily, the network-based security was unable to complete two objectives because it lacked the deep application context required to identify specific scenarios. This is because the visibility was constrained to a network perspective of activities, where without rich packet reassembly and an incredible amount of context, the network device didn't have enough intelligence to identify the activity.

Conversely, the Zero Trust Network was able to quickly identify and contain incidents, because it included the application logs as a signal. This was repeated in other scenarios, with different signals – from the identity and the endpoint.

Incident responders should consider the following recommendations:

- **Centralize application identities.** This can serve as the core PDP and provides a singular logging surface to detect anomalous authentication & authorization.
- **Consume application activity data.** In the example, rich activity data was consumed by the CASB which served as a consolidation point for user activity information inside of the application. Without this, and perhaps as an interim step in network-based security models, feeds from the applications could be consolidated in a central location like a SIEM, and rules could be configured to create alerts.

- **Consider the power of integration in incident response.** In the examples above, there is a direct correlation between identifying an incident and containing an activity, and this was generally completed by built-in integrations between components. The less time that occurs between these two phases, the sooner an incident can be resolved, and possibly more importantly, the criticality or impact may be reduced.

8. Conclusion

This research, while conducted against simple representations of corporate environments, highlights there are clear benefits to defenders and responders in the identification and containment phases as organizations embrace Zero Trust principles in conjunction with cloud services. In scenarios where application insights require more context than the network inspection can provide, or where multiple signals must be combined, Zero Trust Network principles provided an advantage over traditional networks. As a further impetus, and from the data presented in this research, we can see that the current threat landscape nearly requires organizations to embrace these principles as they move to cloud services for critical business functions.

Heath Lawson, heath.lawson@student.sans.edu

References

Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012). Computer security incident handling guide: recommendations of the National Institute of Standards and Technology. Gaithersburg, MD: U.S. Dept. of Commerce, National Institute of Standards and Technology.

Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2019). Zero Trust Architecture, Draft. Gaithersburg, MD: U.S. Dept. of Commerce, National Institute of Standards and Technology.

Scarfone, K., & Hoffman, P. 800-41 Guidelines on Firewalls and Firewall Policy, 800-41 Guidelines on Firewalls and Firewall Policy (2009). Retrieved from https://csrc.nist.gov/publications/detail/sp/800-41/rev-1/final

Class materials for SANS SEC504, Hacker Tools, Techniques, Exploits, and Incident Handling. (n.d.).

Email Collection. (n.d.). Retrieved January 1, 2020, Retrieved from https://attack.mitre.org/techniques/T1114/.

Verizon. (2019). 2019 Data Breach Investigations Report. Retrieved from https://enterprise.verizon.com/resources/reports/2019-data-breach-investigations-report.pdf

Heath Lawson, heath.lawson@student.sans.edu