Network Virtualization
To Enhance Visibility and Containment

B. Germain  ccie,cissp
network virtualization and security
Threats Analysis Promises

Reasoning:
- Current information collected by security devices do not provide the appropriate "visibility" to derive "context" and identify threats
- Collecting more information from diversified sources will provide more "visibility" and "context"
A Reality Check

- 53% of breaches were discovered by external parties (partner, customer, law enforcement, etc.) who then notified the victim
  - ✓ 320 Days = Time until 3rd party detection
- 47% detected internally
  - ✓ 56 Days = Time until internal detection


Breach network Nov 12th
- First POS compromised Nov 15th
- Warning from 2 vendors ignored
- Start of data exfiltration
- Fully deployed and upgraded Dec 2nd
- DOJ contacts Target Dec 12th
- Breach contained Dec 15th
- 40M credit cards & 70M client records
For Threats Analysis tools to be effective we need to allow them enough time to correlate the information.

However, more time = more widely compromised as “containment” is not addressed

Our security architecture is not providing the proper foundations to allow these tools to be effective
Anatomy Of An Attack

Our current architectures provide for poor:

• Visibility
• Containment
• Context

It is also mainly relying on blacklisting
Our Current Response

More visibility but need to sort through more irrelevant data

Architectural weaknesses are not addressed:
- Context is not inherent and requires correlation of arbitrary data
- Containment has not improved
- Still based on blacklisting
Context

- Meaningful information related to the state of the application or its communications
- Naturally achieved by grouping element with similar attributes
  - Member of a particular application, a compliance zone, administrative domain, same function, etc.
- Traditionally achieved through segmentation
  - DMZ, DB segment, VoIP, common, etc.

- Issues & limitations
  - Tied to physical infrastructure and cannot extend easily to different physical environments
  - The “reason” for segmenting does not propagate as information in events
  - Events cannot be ascertained positively as threats, more data is required
  - Multifaceted context cannot be build: How do you segment for "a public web server part of Application-A and administered by Admins-Z"?
Visibility

- You cannot “see” what you do not capture or inspect
  - Inline inspection of the aggregate DC traffic in hardware devices is unfeasible
  - Need to distribute the function

- Sending more ”hay”, ie irrelevant events to the SIEM, requires more efforts and time
  - A whitelisting / least privilege model would generate significantly less events and eliminate false positive
Containment

• The ideal situation
  – A system gets breached
  – The attack is contained in the compromised system until the threat analysis tools figure out something is wrong

• The reality
  – Lateral movement is relatively easy as the infrastructure is exposed in the system
  – Endpoint protection is really good with known attacks, not so good with new ones
  – Very few know and lock down the processes required on a system
  – Treat analysis is done off board on groups of system or requires trending over time to be analyzed mostly by humans

• An attacker has a good window of opportunity and a large attack surface by design
Whitelisting / Least Privilege

• Do you know what is running in your Data Center?
• Do you know which system should talk to which other system over which channel?
• Do you know who should be accessing these systems?
• Do you understand how information flow across particular applications?
• Do you know what should exit your company, by who, to who?
• Etc.

• For most organizations, the answer to these questions is no.
• Therefore we fall back on a blacklisting model
  – Block known threat and assume the rest is ok… log and hope for the best.
  – No easy way to implement whitelisting until now
Whitelisting / Least Privilege
3 Disruptive Technologies Ignite ZT (a least privilege model)

John Kindervag
Forrester's Security and Risk
@Kindervag
Father of Zero Trust

1. Development of Next Generation Firewall Technology to serve as a Segmentation Gateway
2. Development and adoption of Network Virtualization
3. Agile programmability powered by centralized management
What Does VMware NSX Network Virtualization Bring?

• It dissociates the infrastructure from the services it delivers
  – If this would be a privacy talk, we would say that we have dissociated your identity from your home address, phone number, etc.
  – Extensible to clouds / multi-site scenarios and third party integration

• Segmentation boundaries are what you want them to be
  – An admin group, a compliance scope, a security zone, etc or all of the above
  – *This brings context*

• It removes the services from the core of the network to ”split and smear” them granularly at each VM, container, etc
  – *This brings visibility and enables whitelisting*
  – *Network speed becomes irrelevant*

• Creates a “security control plane”
  – Enables declarative policies
  – Enables new security model such as Zero Trust
Requirements For An Improved Security Architecture

• Adopt a whitelisting / Least Privilege model
  – Server virtualization and containers makes it feasible
  – Applies also to hypervisors that can be locked down and baselined for least privileges operation
  – Have tools that can create an application baseline
  – Simplified logic: Allow & monitor whitelisted traffic / drop & alert for anything else
    • All drop events are now 100% pertinent and threat analysis tools can concentrate on valid traffic being abused
  – Provides better containment by default

• Embrace network virtualization
  – Dissociate infrastructure from security requirements, ie don’t base your security on IP addresses, subnets, Vlans, etc
  – Establish a central policy / security management plane
  – Distribution of security functions at every system providing true micro-segmentation
Horizon View Networking – flows and protocols

Source Ray Heffer
## Horizon 6 Services (partial)

<table>
<thead>
<tr>
<th>Horizon Service</th>
<th>Protocol</th>
<th>Destination ports</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon6-Agent</td>
<td>TCP</td>
<td>4172,3389,9427,32111,22443,3383,71</td>
<td>any</td>
<td>PColP,RDP,MMR,USB redirection</td>
</tr>
<tr>
<td>Horizon6-ComposerService</td>
<td>TCP</td>
<td>80,443,18443</td>
<td>any</td>
<td>Secure connection between composer service and connection servers</td>
</tr>
<tr>
<td>Horizon6-CS_inbound_client</td>
<td>TCP</td>
<td>4172</td>
<td>any</td>
<td>Client connections to internal connection server</td>
</tr>
<tr>
<td>Horizon6-CS_inbound_client2</td>
<td>UDP</td>
<td>4172</td>
<td>any</td>
<td>If PColP secure gateway is used</td>
</tr>
<tr>
<td>Horizon6_interCS</td>
<td>TCP</td>
<td>4001,4100,8009</td>
<td>any</td>
<td>CS to CS traffic</td>
</tr>
<tr>
<td>Horizon6_SS_to_CS_1</td>
<td>TCP</td>
<td>4001, 8009</td>
<td>any</td>
<td>SS to CS traffic</td>
</tr>
<tr>
<td>Horizon6_SS_to_CS_2</td>
<td>UDP</td>
<td>5,004,500</td>
<td>any</td>
<td>SS to CS traffic</td>
</tr>
<tr>
<td>Horizon6_SS_to_Agent_1</td>
<td>TCP</td>
<td>4172, 9427, 3389, 22443, 4001</td>
<td>any</td>
<td>SS to agent</td>
</tr>
<tr>
<td>Horizon6_SS_to_Agent_2</td>
<td>UDP</td>
<td>4172</td>
<td>any</td>
<td>SS to agent</td>
</tr>
</tbody>
</table>
A Whitelisting Approach

• Take your application one at the time and get a fingerprint of its communications:
  – The internal flows between the components of the application
  – The external flows to other systems in the data center
  – Who the users are and where they come from

• Establish how to group the components for the application
• Create your whitelist ruleset
A Whitelisting Approach
The Horizon View “bubble” – Grouping by functions
A Whitelisting Approach
Establishing the relationships - infrastructure
A Whitelisting Approach
Establishing the relationships - Applications
How Whitelisting Helps Contain Unknown Threats?

Not about technologies
- Whitelisting
- Containment
- Context
- Automation

client <-> sec_srv
sec_srv <-> con_srv
den all

Security Servers
Connection Servers
vDesktops

View Composer
vAPP RDS

vmware
Extending To Communications

- Using the same grouping to ensure the proper system are the ones talking to each other
  - Encryption and / or authenticity and / or integrity
  - Protection against spoofing and eavesdropping
  - Denies any other source, any other communication channels
  - Done at the hypervisor so out of reach from the service itself
  - Key distribution, rotation and revocation managed by the same control plane
Extending To The Cloud – AWS Example

Native support for AWS instances with coherent services and security posture for on and off-premise

**Amazon Web Services**
- Native AWS Server instances (AMI's)
- Added to NSX virtual networks via policy

**On-Premise NSX/vSphere**
- AWS instances are added to logical switch
- Consistent security posture on-premise and in cloud
- AWS instances leverage services

**AWS Cloud**

**Data Center**
- Web Server
- HR Server

**Developer**
Launches instances via Amazon console

**IT Administrator**
Defines network and security policy

**Internet**
Architecture changes provide better hay, not the tools
Summary

• Current threat analysis tools alone will not solve fundamental flaws in the way we architect our network security

• Virtualization in general and Network Virtualization specifically provides security properties that we were not able to get in the physical space

• Network Virtualization brings today
  – Better context
  – Better visibility
  – Better containment
  – Extensions to the cloud, multiple sites and to communications in general

• A security control plane
  – Tracks everything in the infrastructure
  – Translates declarative security into specific rule sets for the technology of your choice
Ever Wonder Why They Don’t Build These Anymore?
Thank you