Dude, Where's My Log?
The Unknown Logging Gaps in Your Environment, Why You Didn't Detect that Pentest, and How Splunk Can Help

Kevin Kaminski
R&D Content Lead | ReliaQuest

splunk> .conf19

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or plans of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results may differ materially. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only, and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionalities described or to include any such feature or functionality in a future release.

Origins

Why this talk?

But Why are the Logs Gone?

Logging Coverage Expectations

Network visibility

Logging Coverage Reality

Visibility gaps

Firewall rule blocking syslog

Outdated syslog server IP

Windows not logging registry mods

"**Not logging** is the only way to know everything you are logging."

Co-worker

Food for Thought

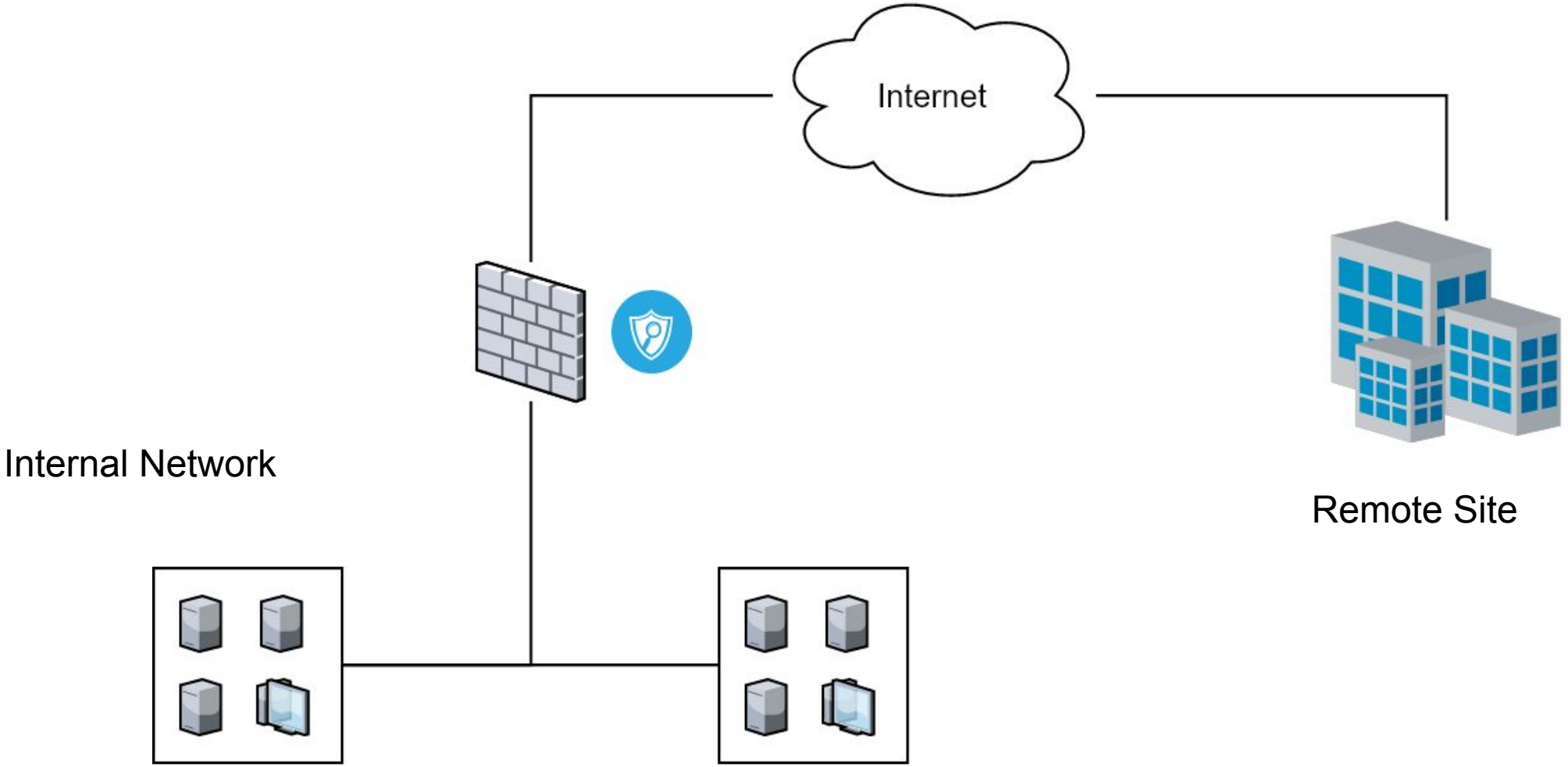
Ask these questions frequently

1. Are my log sources deployed widely enough to cover my environment?
2. Are my log sources configured properly to detect the threat?
3. Are my log sources even capable of detecting the threat?

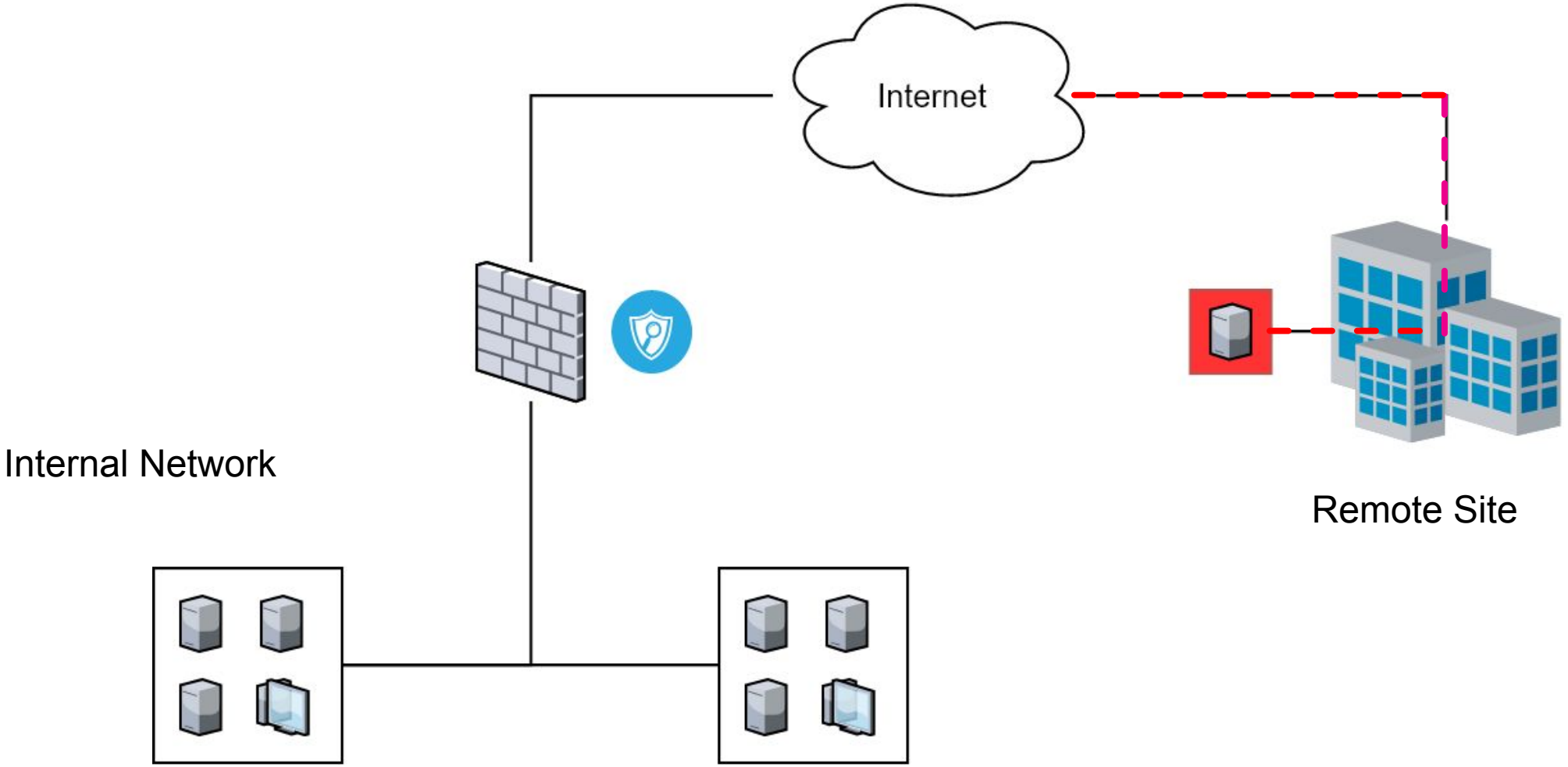
Incomplete Log Source Coverage

Quality of logs is meaningless if coverage is small

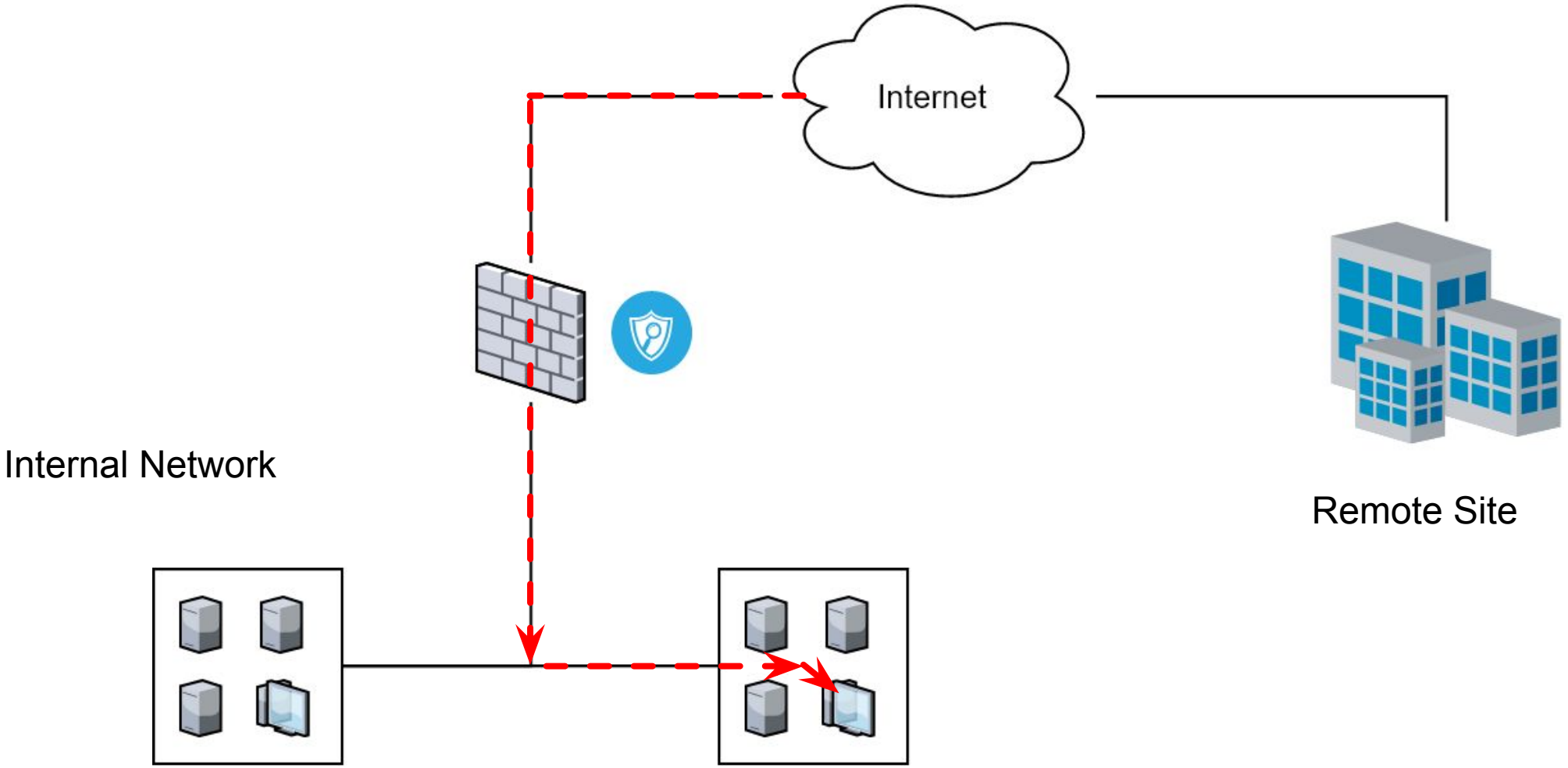
An All Too Common Setup



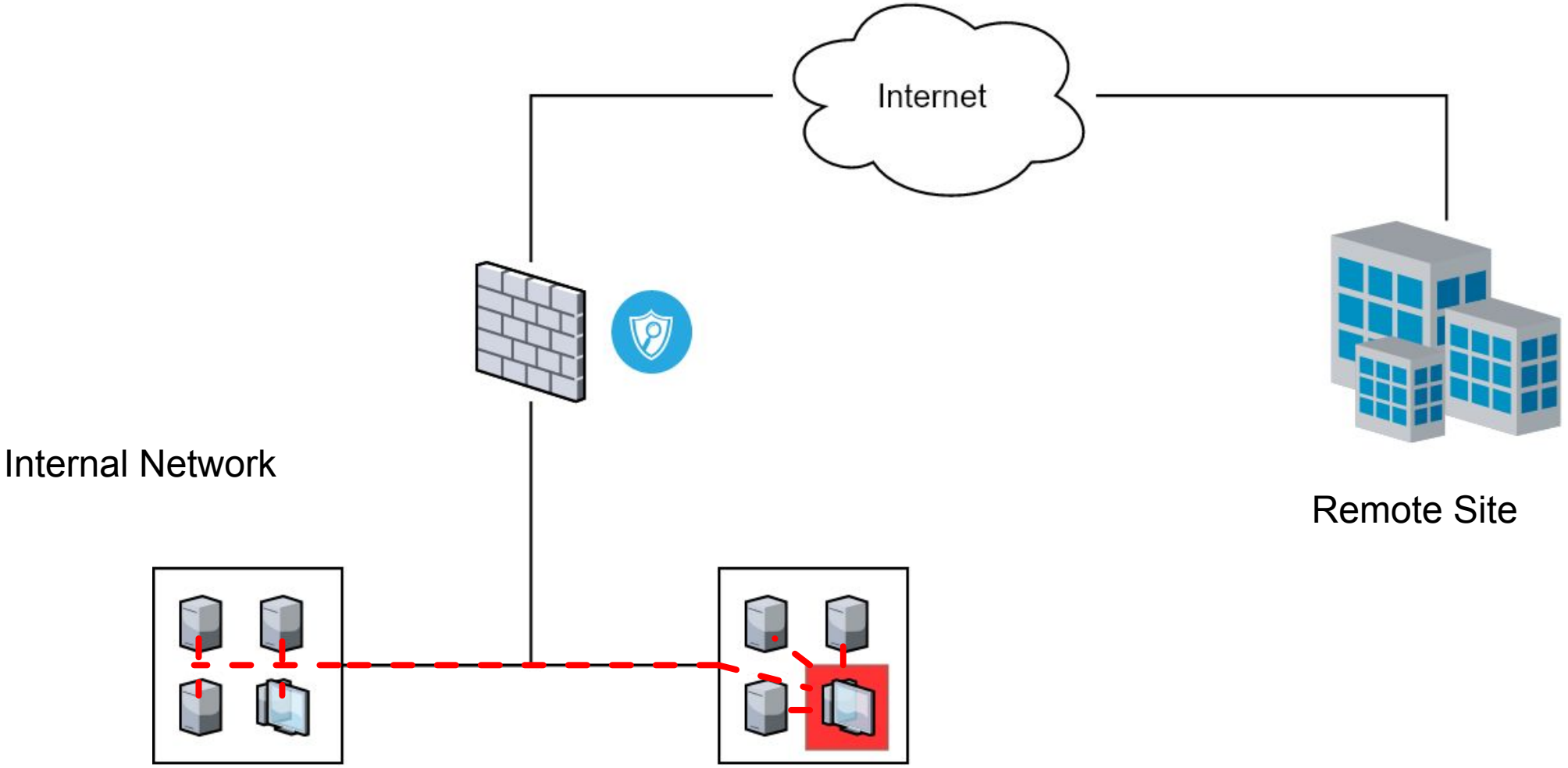
Command and Control Traffic



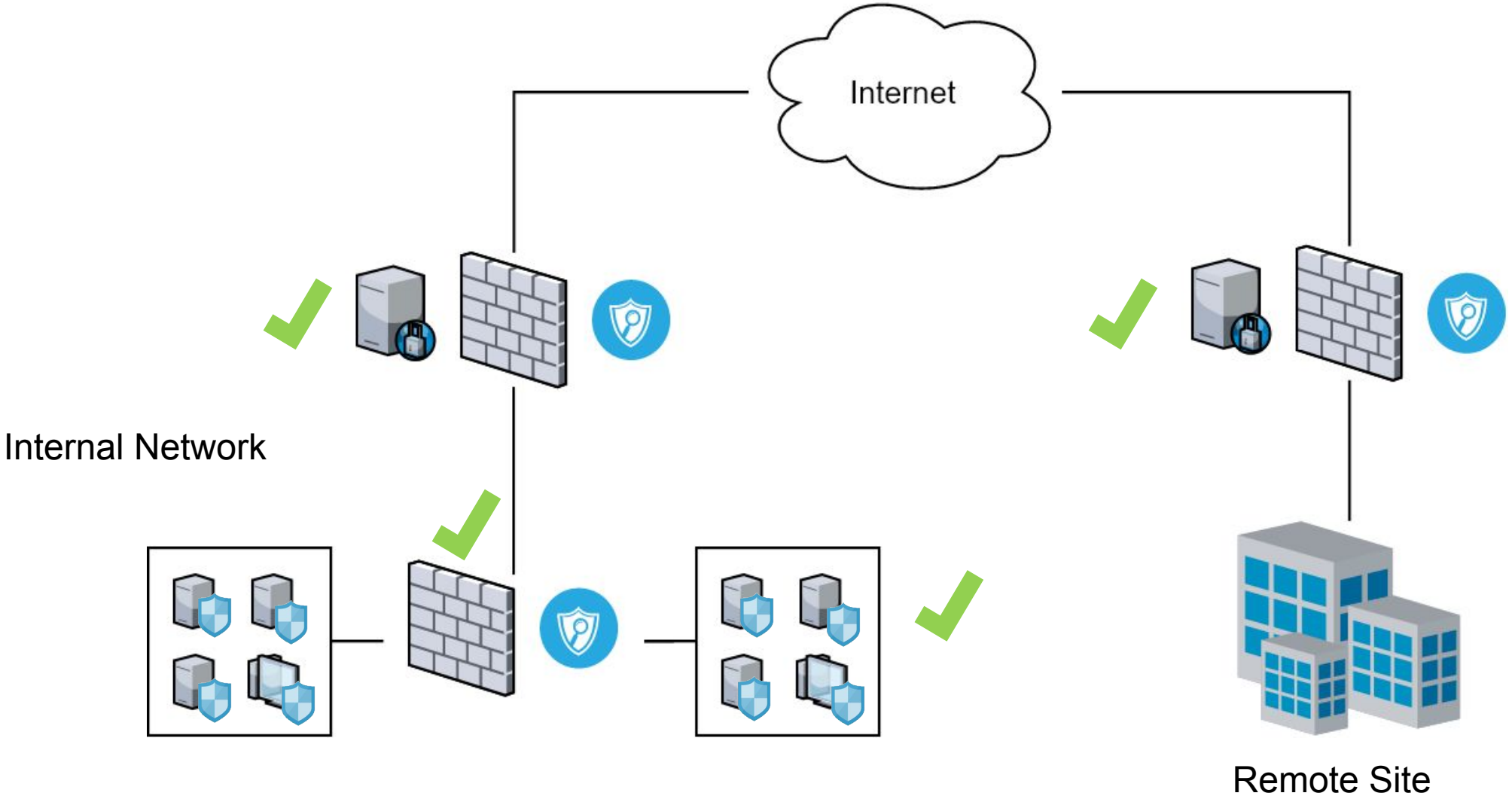
Malicious File Download



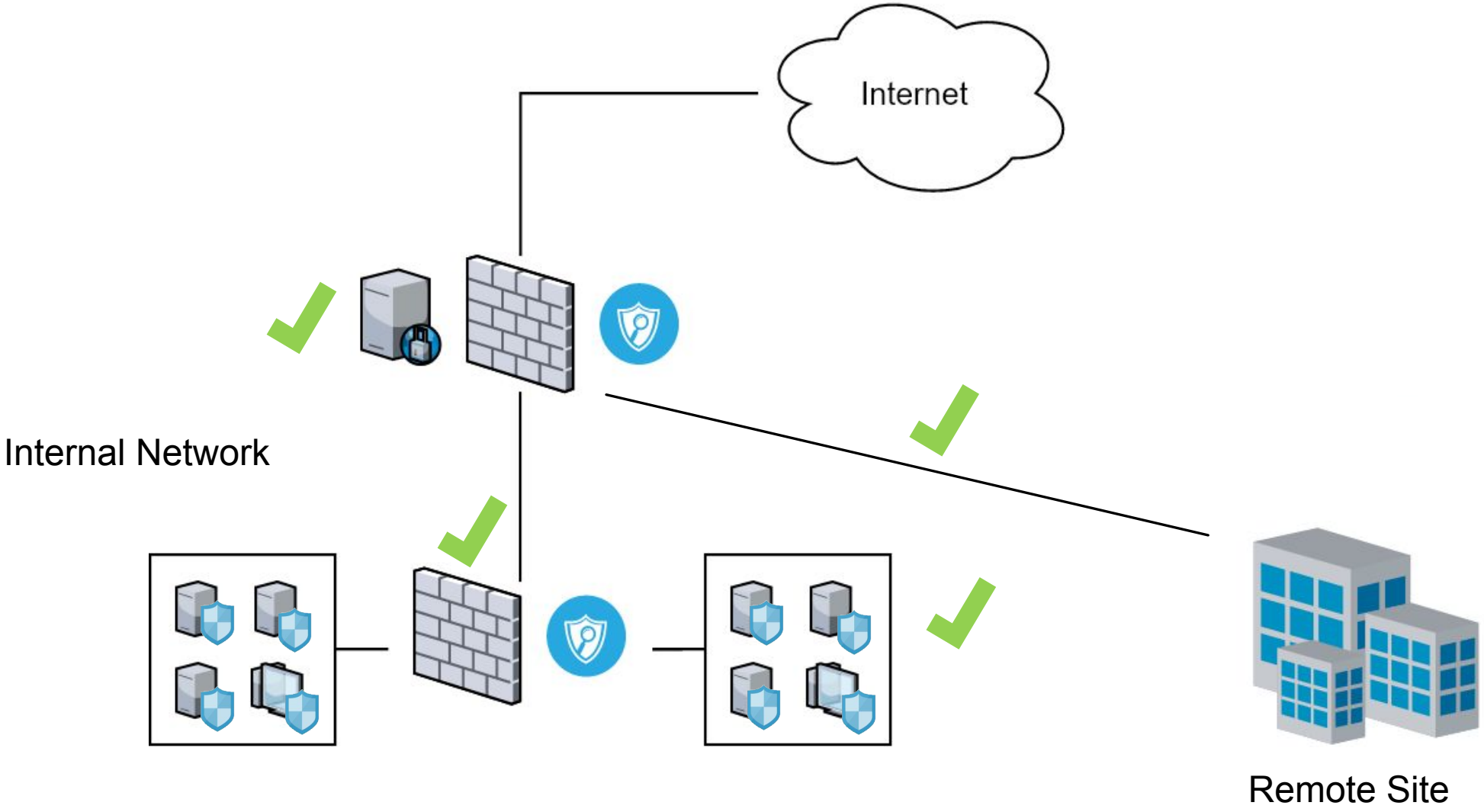
Internal Compromise and Pivot



Fix – Introduce New Devices



Fix – Reroute Traffic



Log Coverage Recommendations

Deploy security technologies between internal segments

- Consider logging flow data

Reroute traffic through logging devices

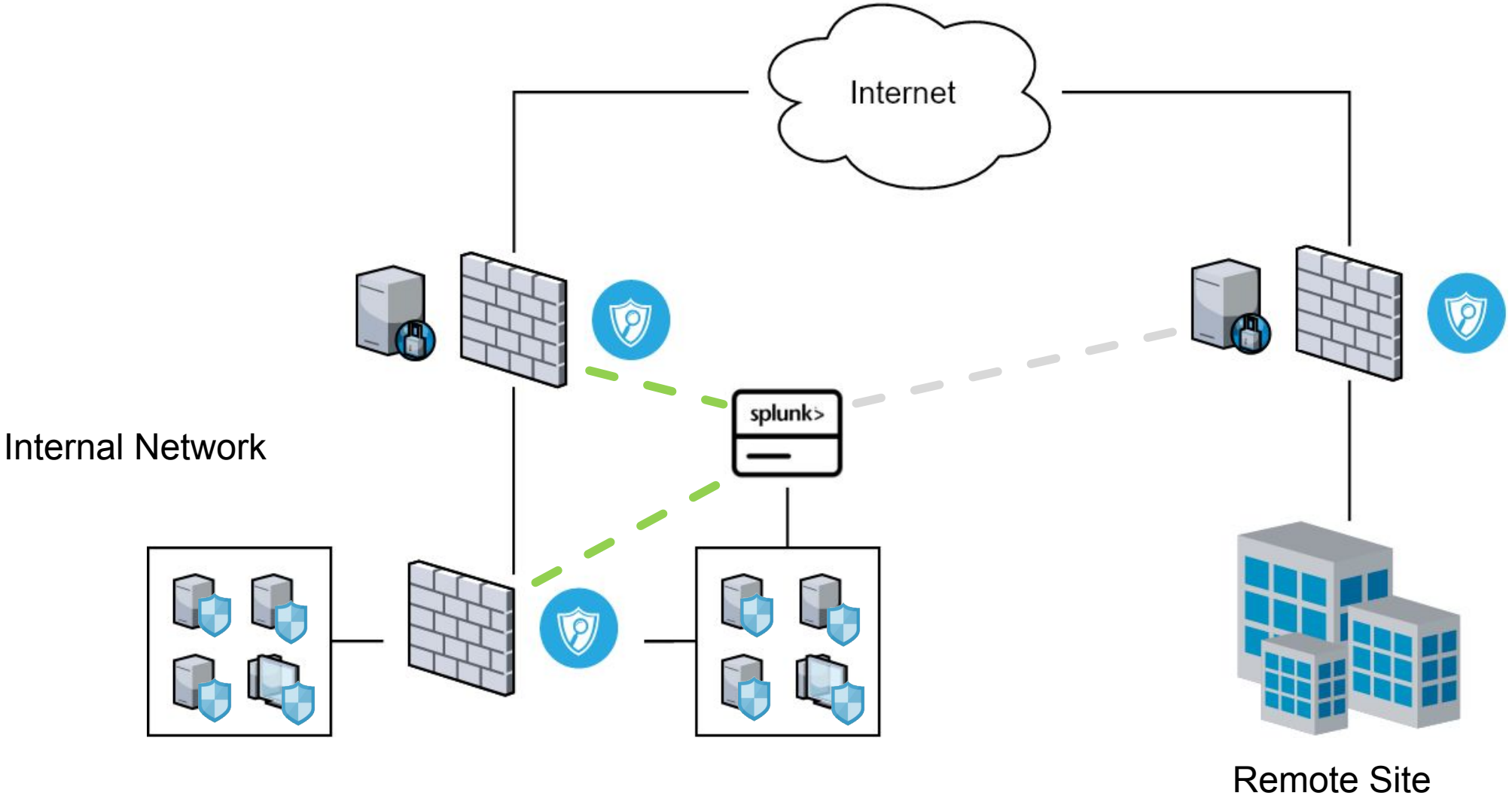
Track endpoint security installs

Consider purchasing additional tech

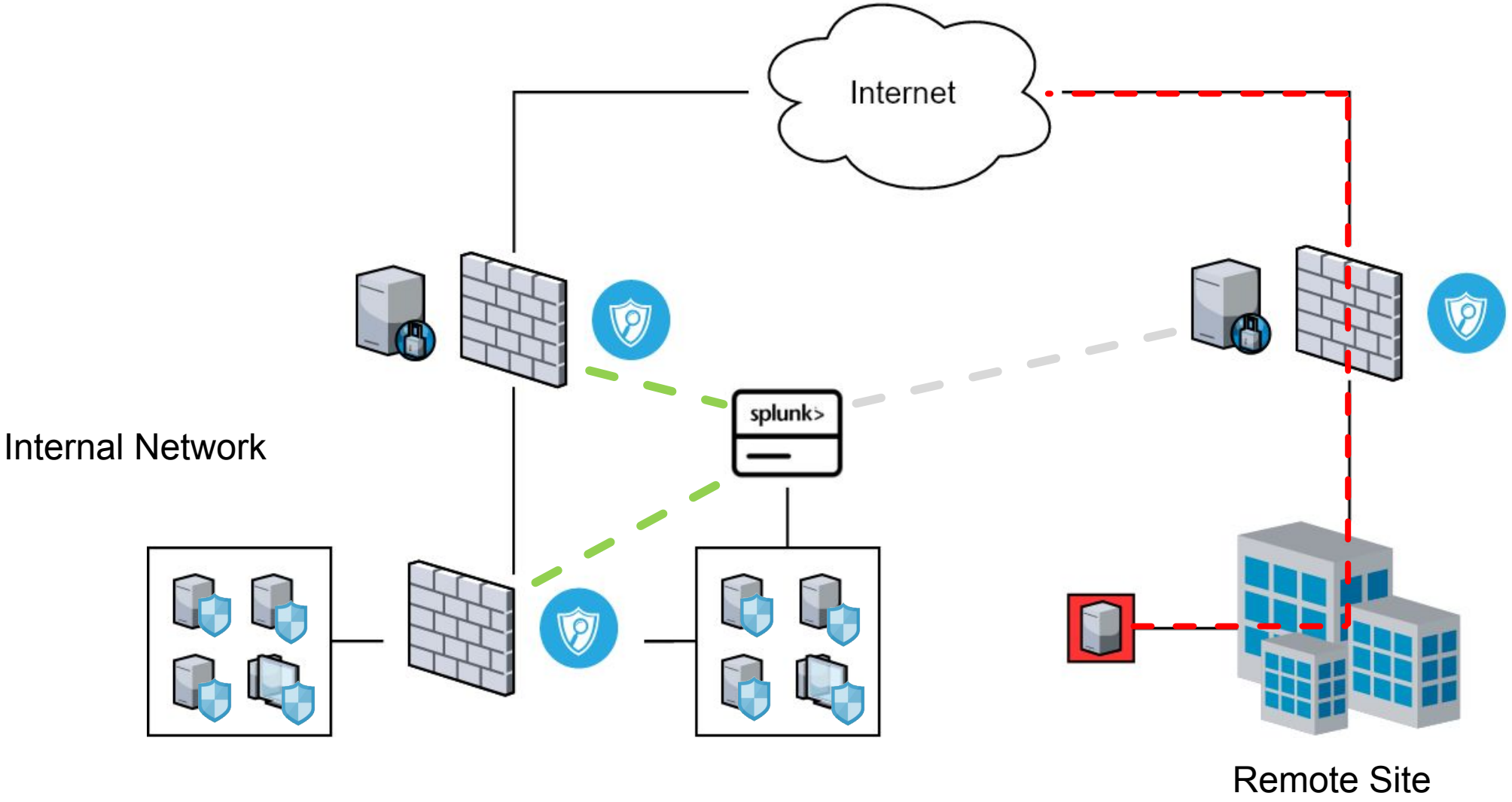
Not Forwarding to Log Platform

Coverage is less useful if there is no visibility

Remote Site Missing Logs

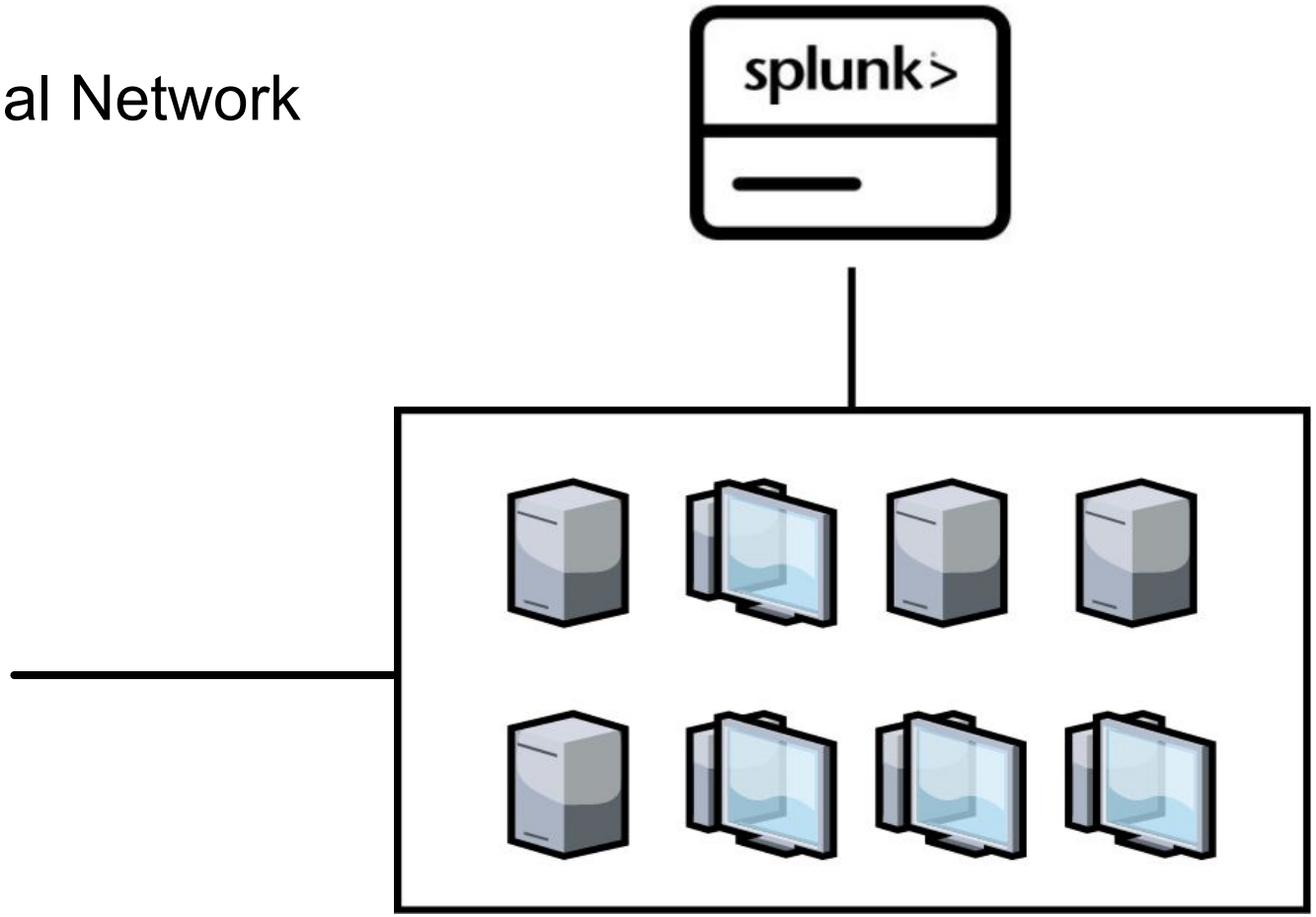


C2 Traffic Also Missed



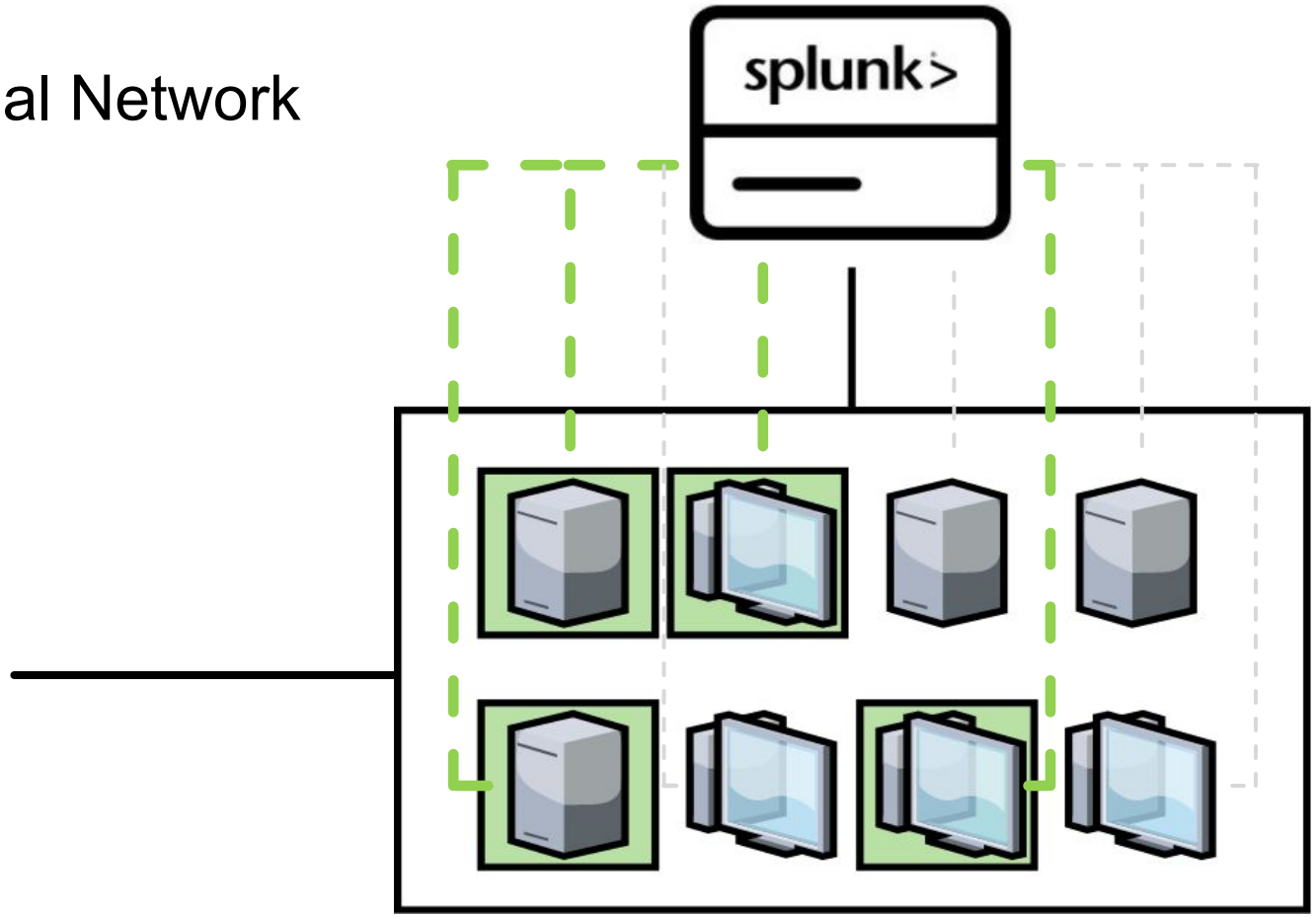
Internal Network Setup

Internal Network



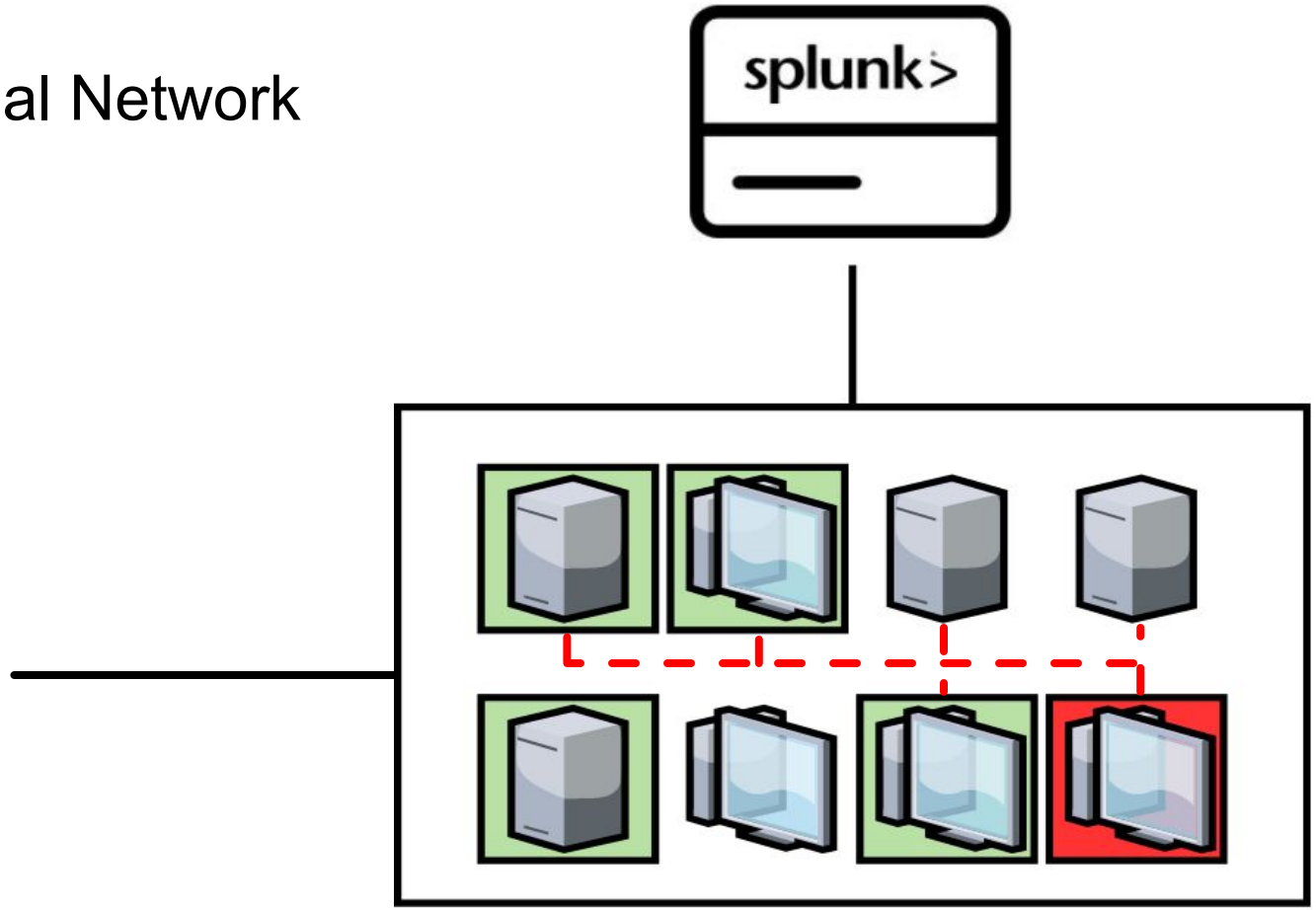
Inconsistent Endpoint Logging

Internal Network



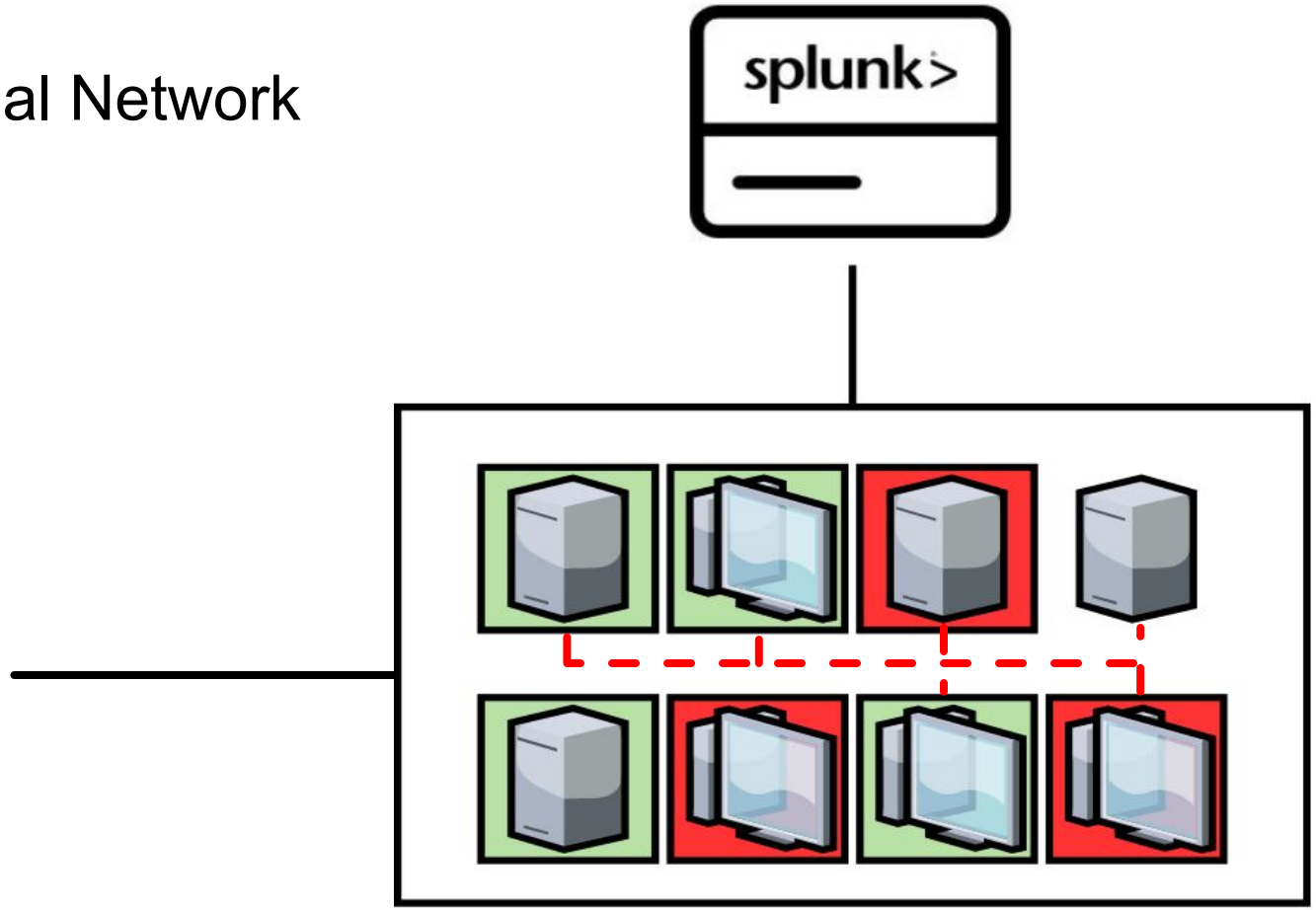
Endpoint Compromise and Recon

Internal Network



Undetected Pivoting

Internal Network



Detect with Splunk

Metasearch command can quickly query for logging hosts

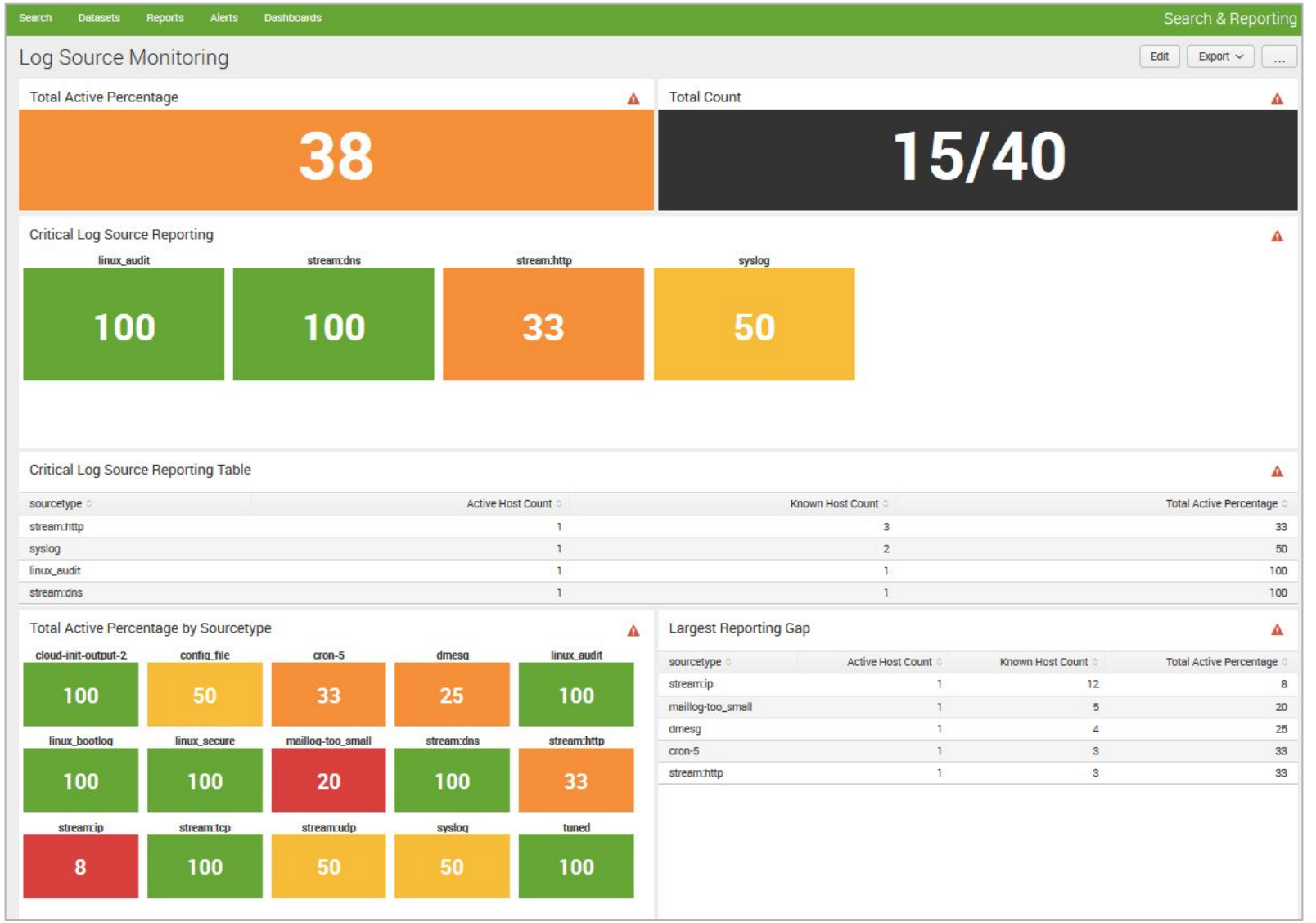
Create a lookup with known counts per sourcetype

- Schedule a report or alert to show deviations

```
| metasearch index=* sourcetype=*
| stats distinct_count(host) as "Hosts Reporting" by sourcetype
```

sourcetype	Hosts Reporting
DhcpSrvLog	18
MSAD:NT6:DNS	22
cisco:asa	9
okta	1

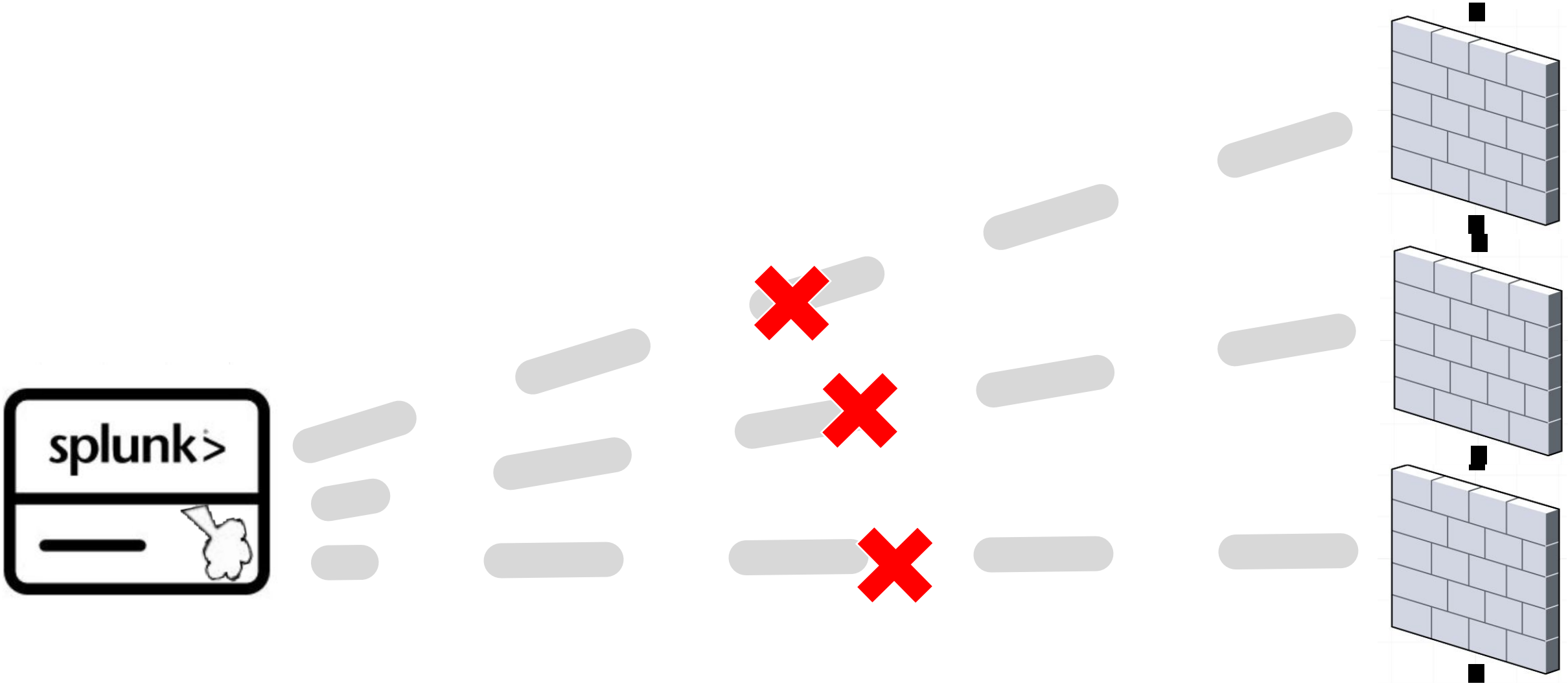
Sample Dashboard



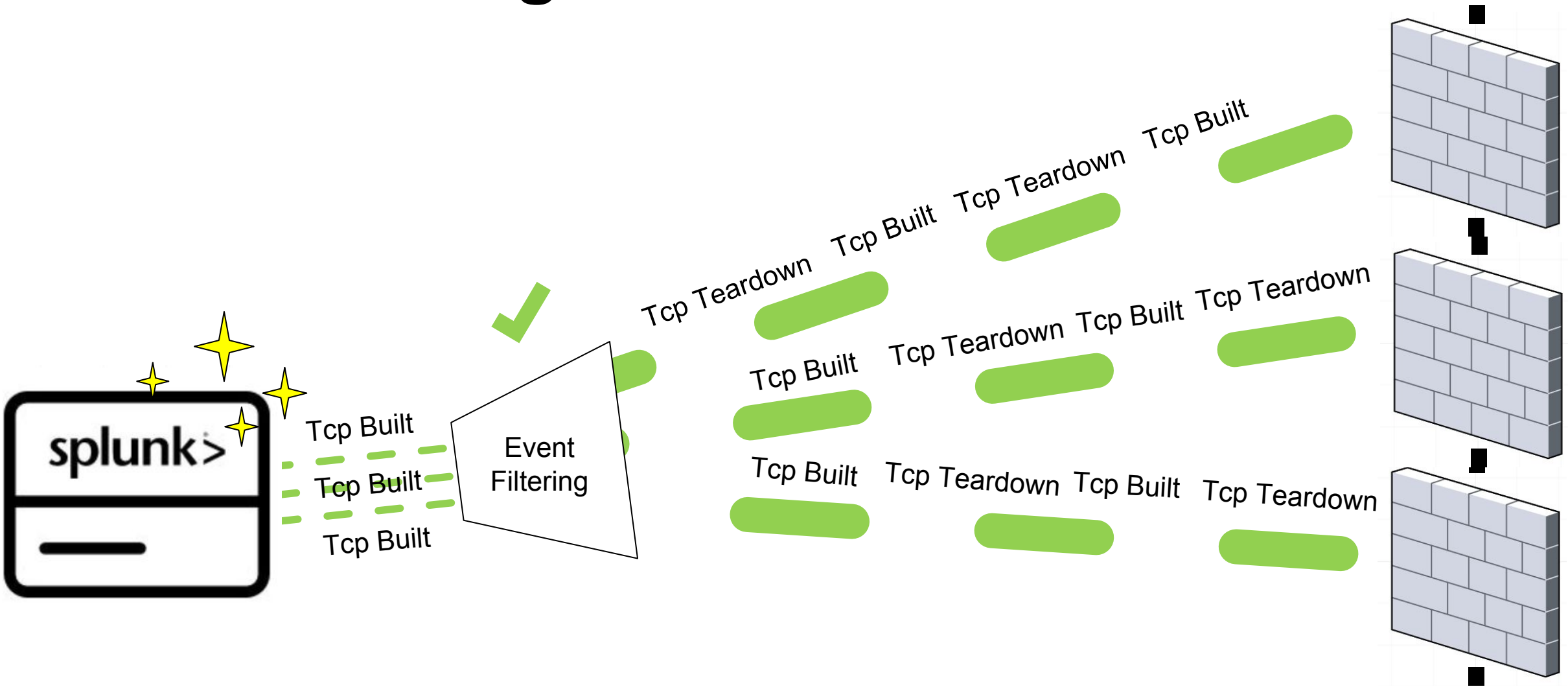
splunk> .conf19

Too High Log Volume

Intentional Removal



Event Filtering



Log Forwarding Recommendations

Remediate logging issues

- Solution dependent on log source and environment

Audit inventory with CMDB

- Compare device counts with Splunk

Automate logging configuration during provisioning

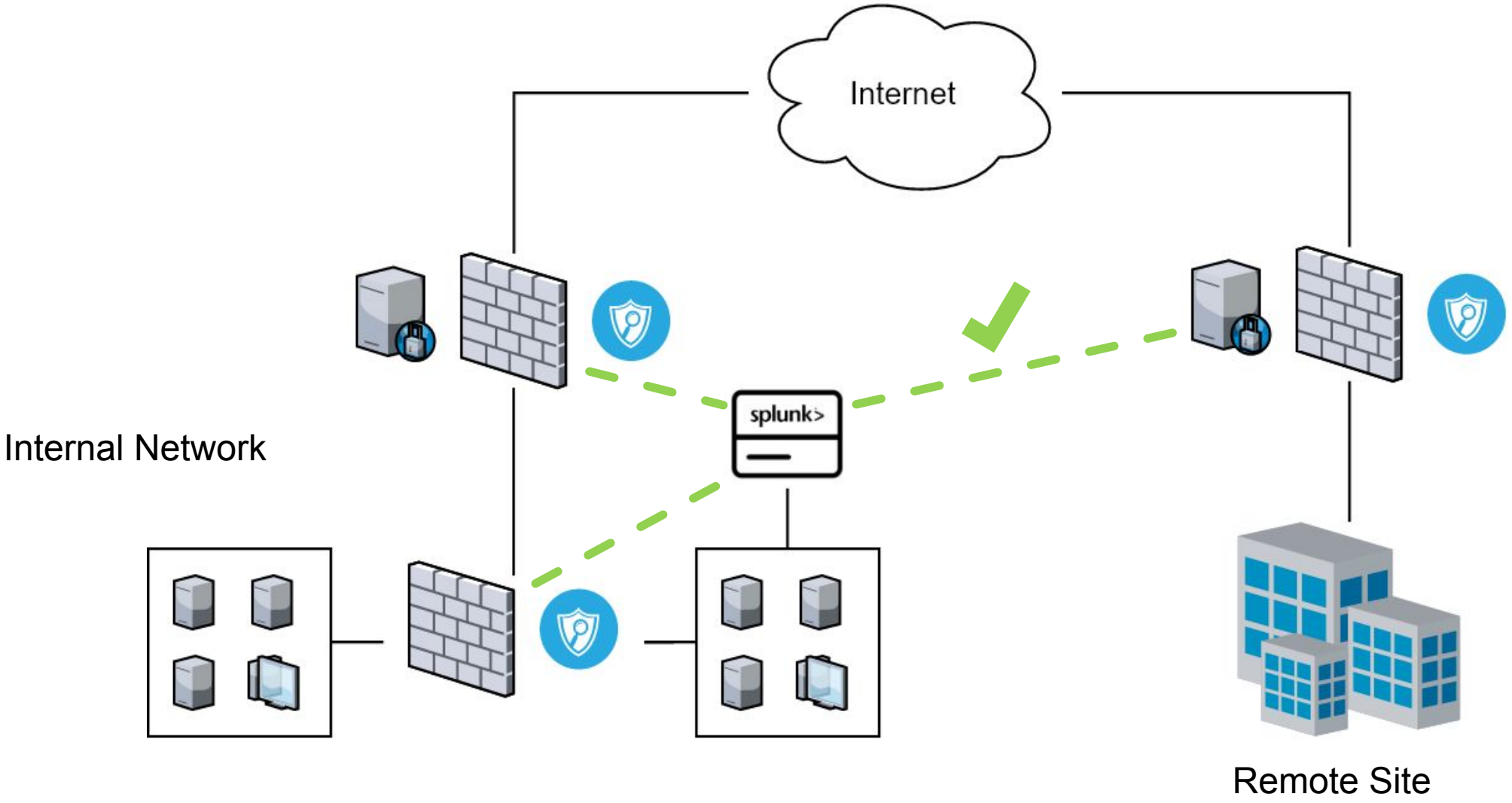
- For Windows, use Windows Event Forwarding (WEF)
- For Linux, manage syslog config with automation tools (Puppet, Ansible, etc.)

Filter event types on high volume log sources

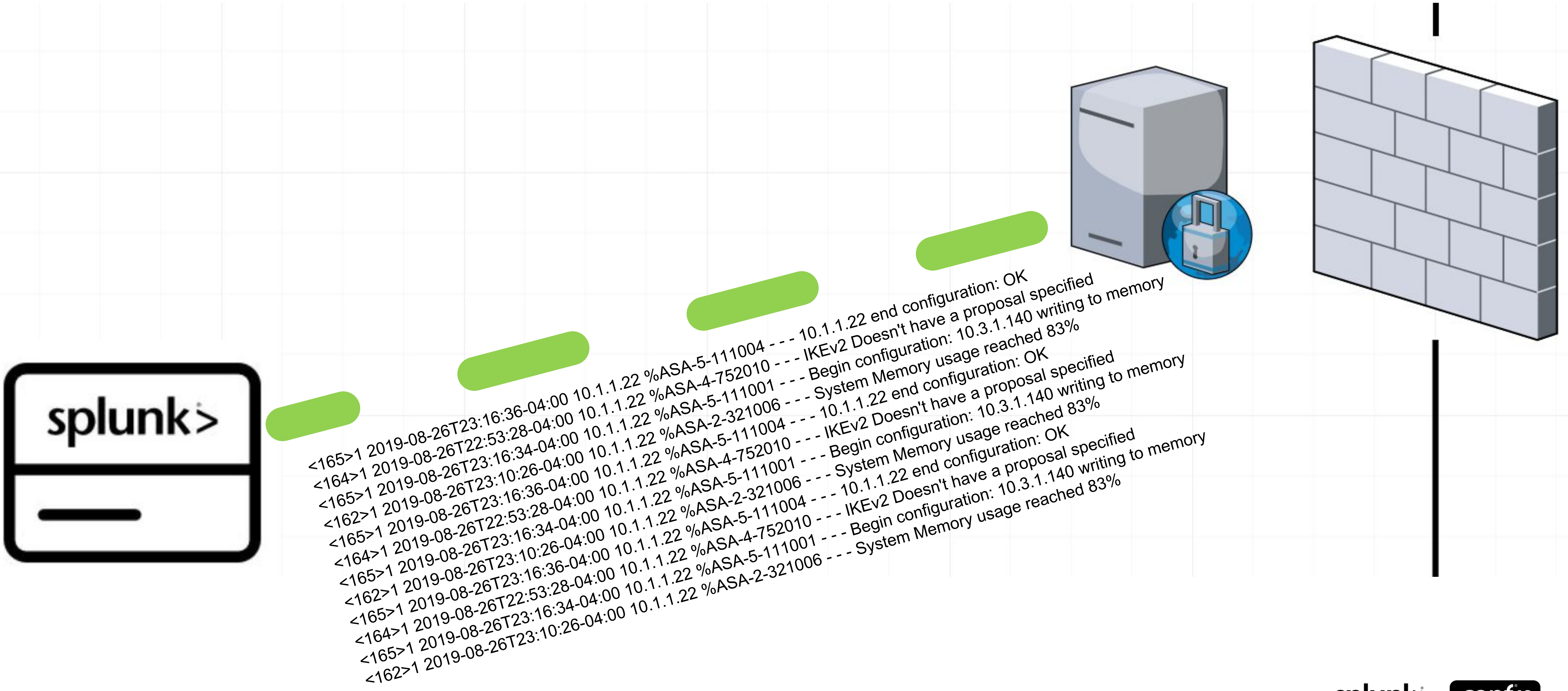
Improperly Configured Logging Levels

Log source not used to its fullest potential

Successful Logging

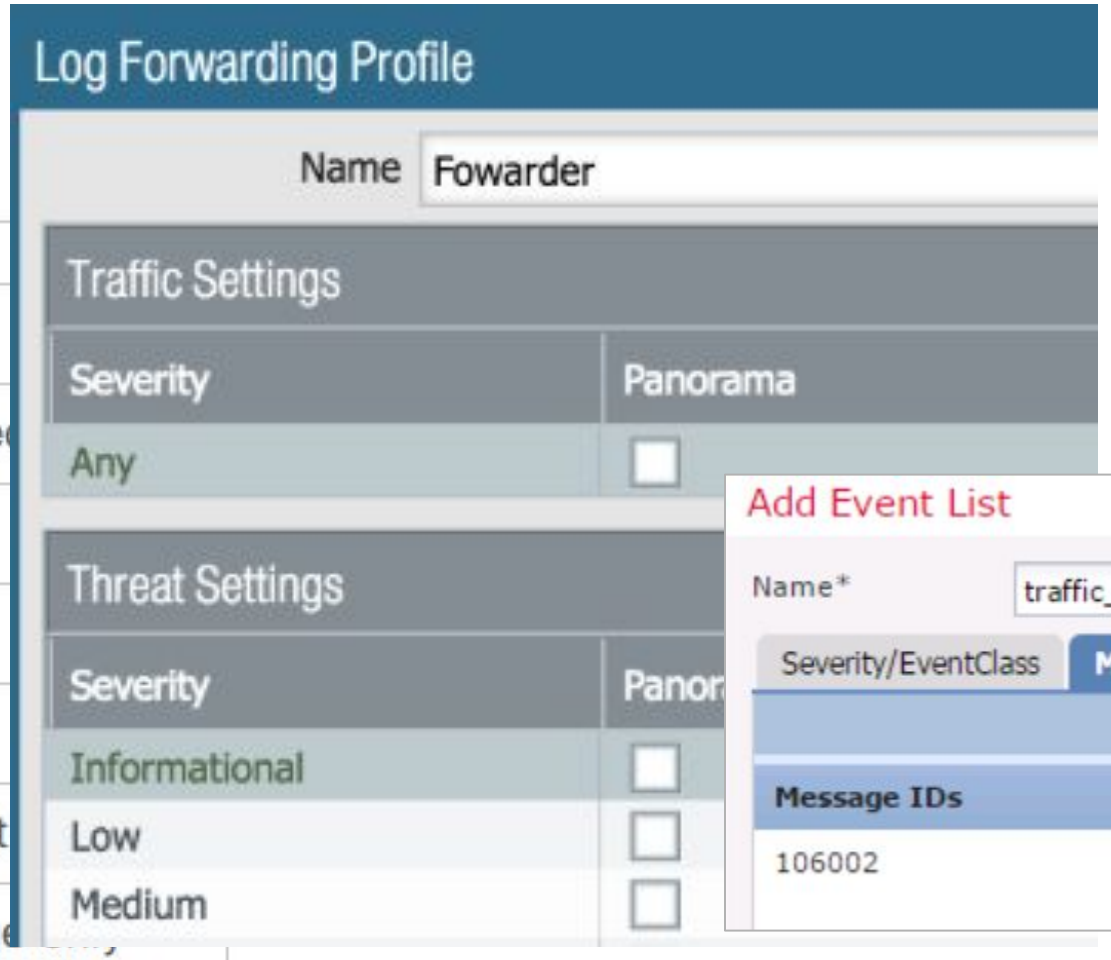


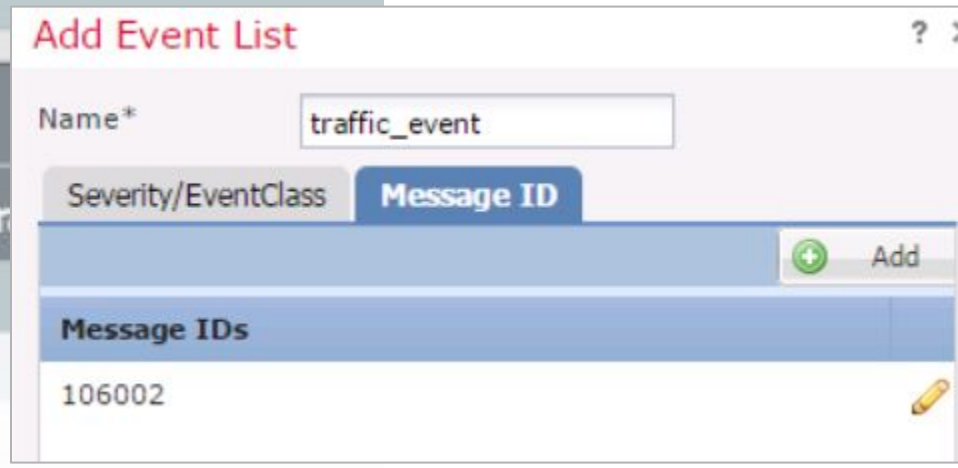
But What is Actually Logging?



Examples of Logging Levels

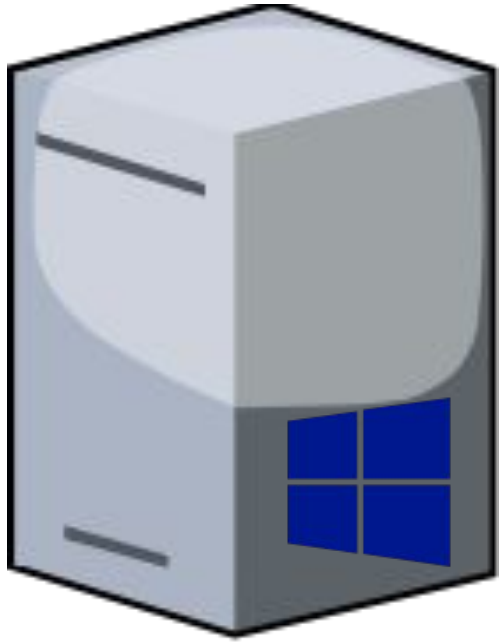
Level	Description
0 – emergency	System unusable
1 – alert	Immediate action nee
2 – critical	Critical condition
3 – error	Error condition
4 – warning	Warning condition
5 – notification	Normal but significant
6 – informational	Informational message
7 – debugging	Appears during debugging only





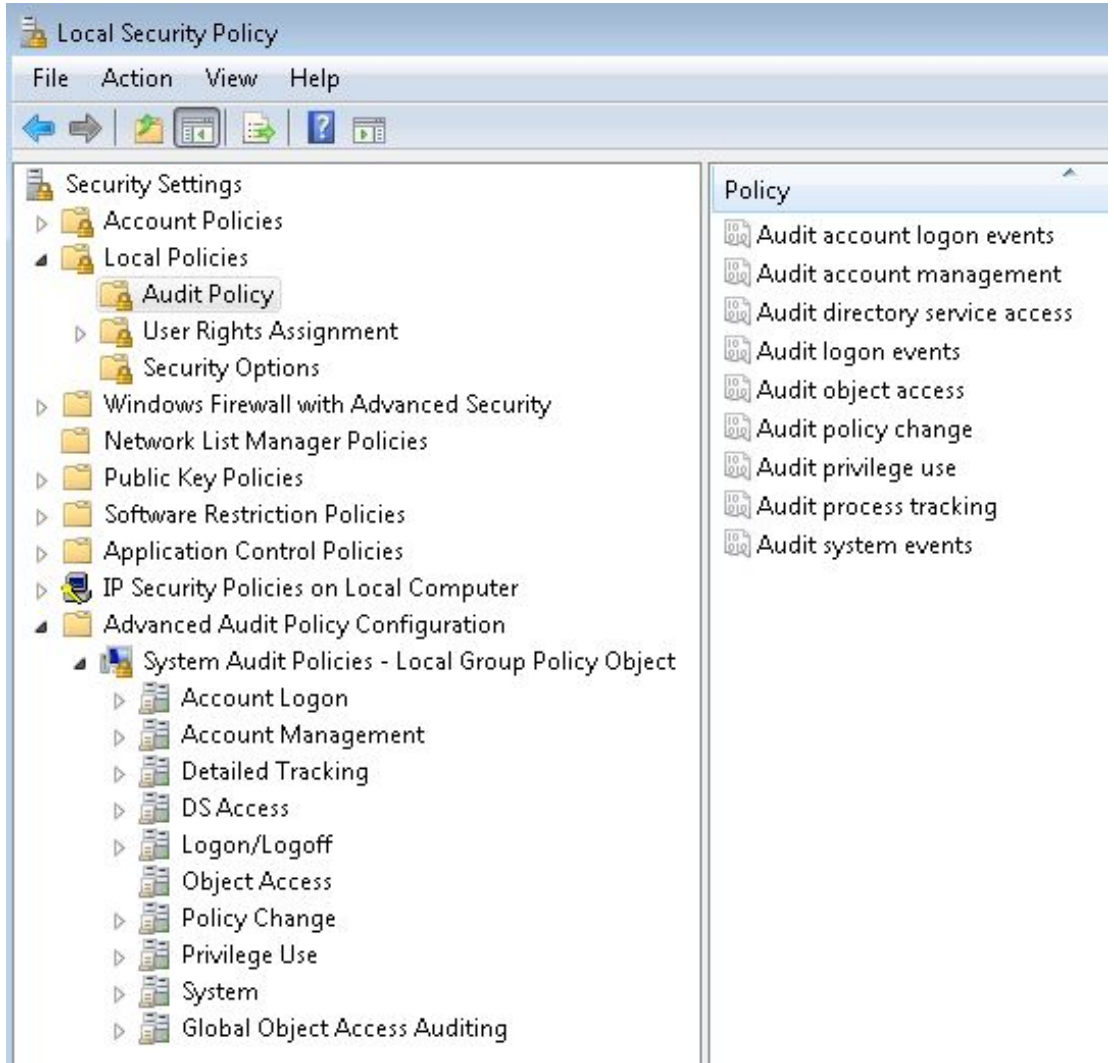
Windows Attack Scenario

PSEXEC with Mimikatz Compromise



- ✗ Attacker connects to network share
- ✗ Registers a new service
- ✗ Registry key modified
- ✗ Credentials scraped with Mimikatz
- ✓ Login with compromised account

Windows Auditing Policies



Object Access/Audit File Share

- Network Share Accessed: 5140

System/Audit Security System Extension

- New Service Installed: 4697

Object Access/Audit Registry

- Registry Modified: 4657

Audit Policy/Audit account logon events

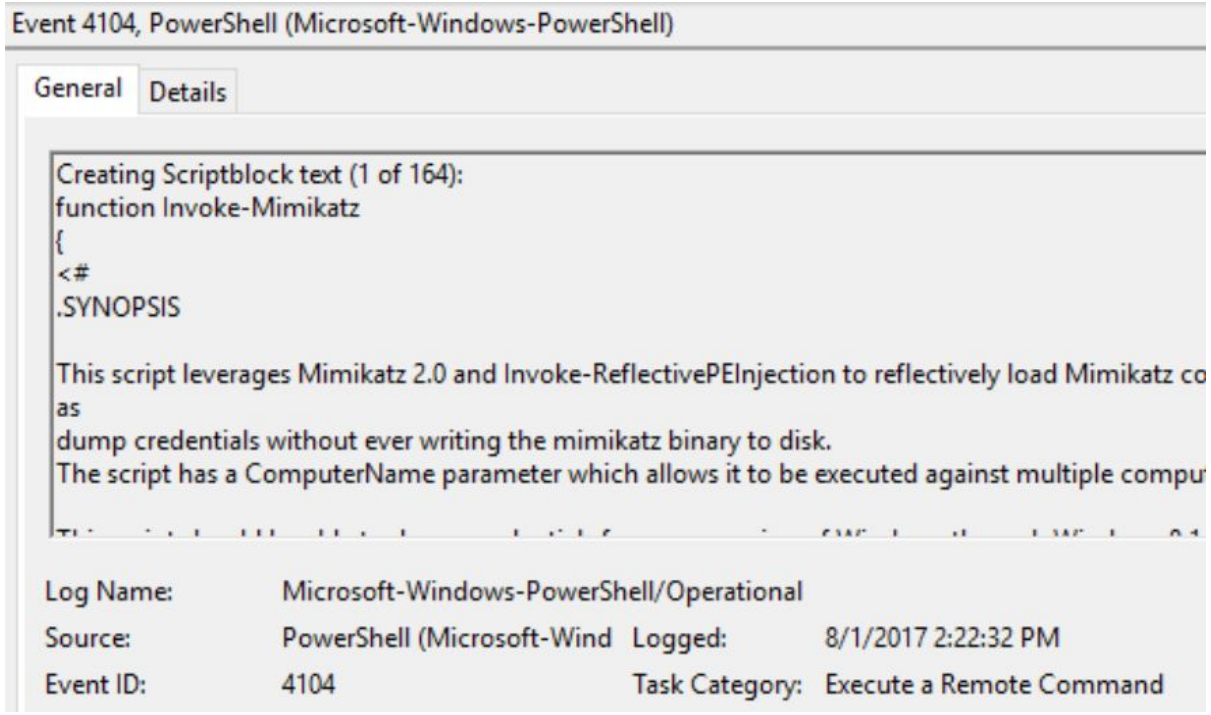
- Account Logon: 4624

splunk> .conf19

PowerShell Logging

Process Creation (4688)

PowerShell Script Block (4104)



splunk> .conf19

Linux Attack Scenario

Malicious Payload to Admin Compromise

- ✗ Malicious file download with curl
- ✗ File executed with parameters
- ✗ Host enumerated
- ✗ Privileges escalated
- ✓ Login with compromised account

Logging Linux Command History

> select time,username,command from users join shell_history using (uid);

auditd(8) - Linux man page

Name

auditd - The Linux Audit daemon

> auditctl –a exit,always execve

Detect with Splunk

Query to view logged Windows event IDs

Can run similar searches for other log sources

```
index=wineventlog sourcetype=WinEventLog:Security
| stats values(name) as Description by EventCode
| sort EventCode -
```

EventCode	Description
4702	A scheduled task was updated
4723	An attempt was made to change an account's password
4724	An attempt was made to reset an accounts password
4728	A member was added to a security-enabled global group
4735	A security-enabled local group was changed
4737	A security-enabled global group was changed
4738	A user account was changed

Logging Level Recommendations

Ensure correct event types are configured

Enable logging for important Windows events

- Enable PowerShell logging (script block or module)

Enable Auditd or osquery

Lack of Detection Capability

No functionality to detect certain activities

Common Detection Failure Examples

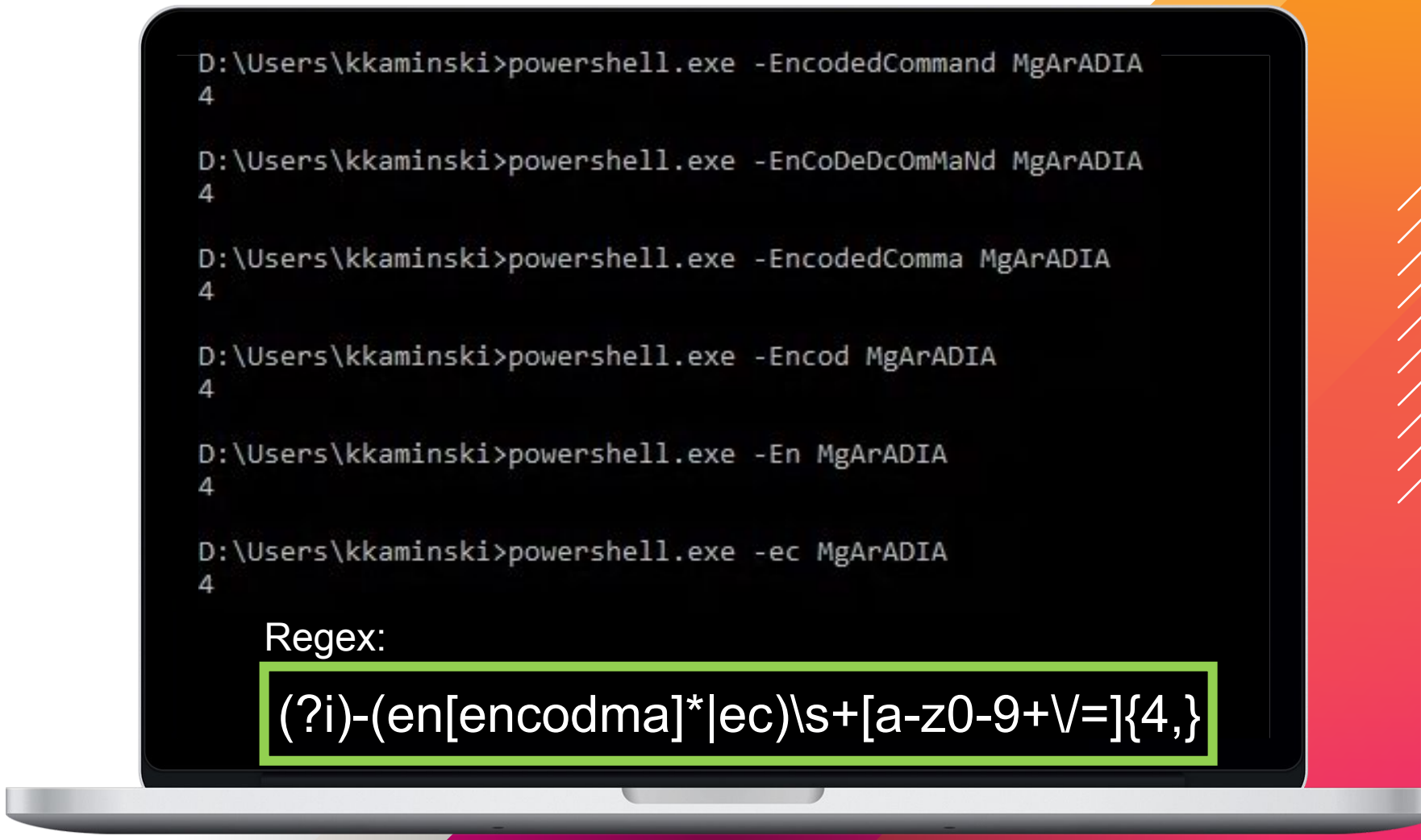
Static IDS/Antivirus signatures

General lack of signatures

Firewalls without application protocol awareness

PowerShell Arguments

- Arguments are case-insensitive
- Arguments are auto-completed as long as the string is unique



Linux Recon Tool Execution

Malicious script

```
root@kali-1:/usr/bin# enum4linux -U 169.254.182.109
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Su
n Aug 25 22:28:44 2019

 ==========================
|    Target Information    |
 ==========================
Target ........... 169.254.182.109
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
```

Log output

username	command
root	enum4linux -U 169.254.182.109

Renaming File Bypass

Malicious script

```
root@kali-1:/usr/bin# cp enum4linux benignfile
root@kali-1:/usr/bin# benignfile -U 169.254.182.109
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux

 ==========================
|    Target Information    |
 ==========================
Target ........... 169.254.182.109
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
```

Log output

username	command
root	benignfile -U 169.254.182.109

Obfuscating Behind Script

Malicious script

```
  GNU nano 2.8.7

#!/bin/bash
enum4linux -U 169.254.182.109
```

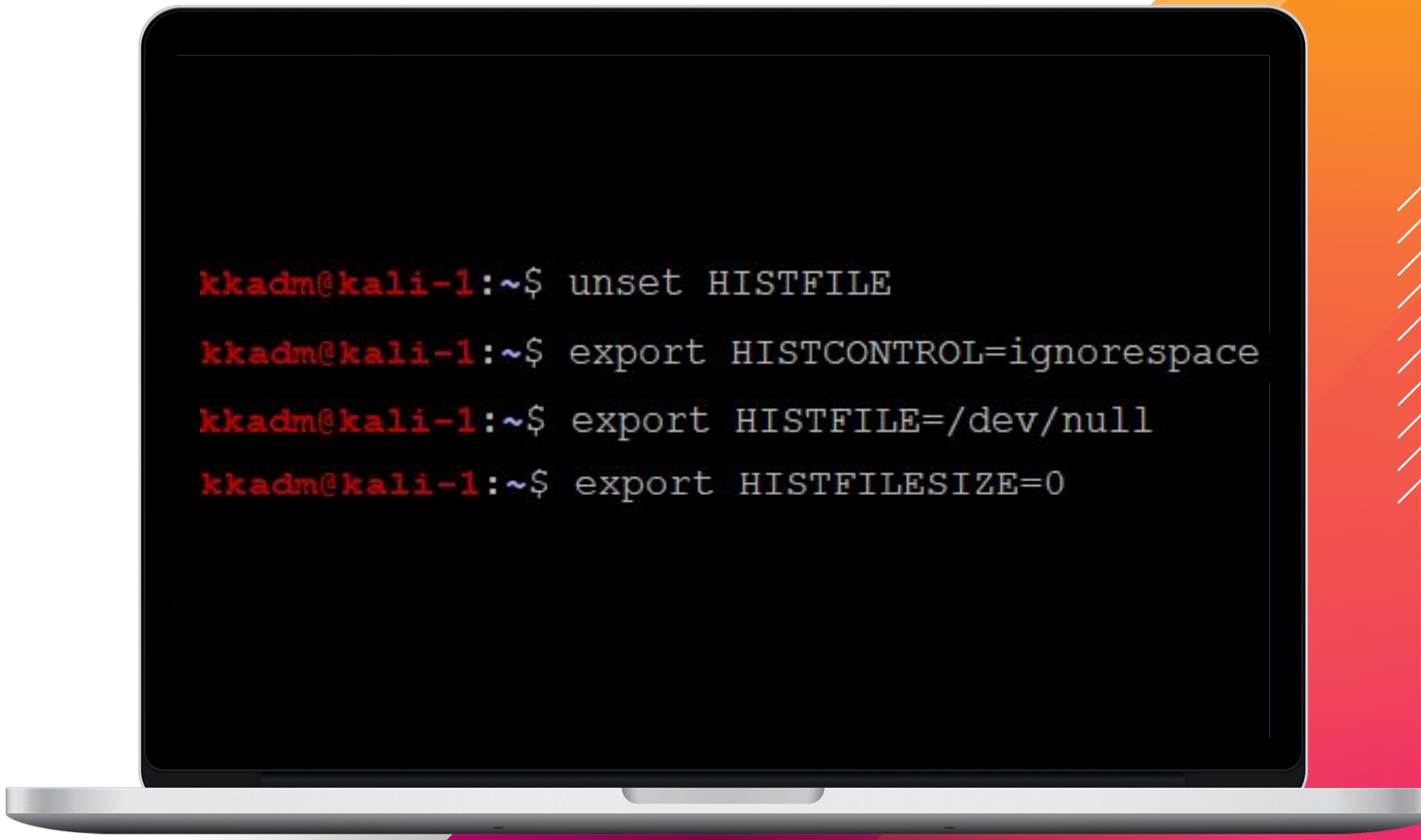
```
root@kali-1:~# nano vim
root@kali-1:~# chmod 755 vim
root@kali-1:~# ./vim
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux

 ==========================
|    Target Information    |
 ==========================
Target ........... 169.254.182.109
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
```

Log output

```
+----------+---------+
| username | command |
+----------+---------+
| root     | ./vim   |
```

Bash History Bypass

Multiple ways to bypass bash history logging



Lack of Detection Recommendations

Behavioral-based tools

Defense in depth

Know the weaknesses in your tools' signatures

- Account for variations of PowerShell arguments and other commands

Enable auditd or osquery

Testing

Proactively discover logging gaps

How to Test

1. Assign roles for the tester and validator
2. Identify the activity you would like to detect
3. Identify the log source dependencies
4. Coordinate with teams to setup necessary infrastructure
5. Emulate activity
6. Verify logs in SIEM

Testing Challenges

Target host does not mirror production config

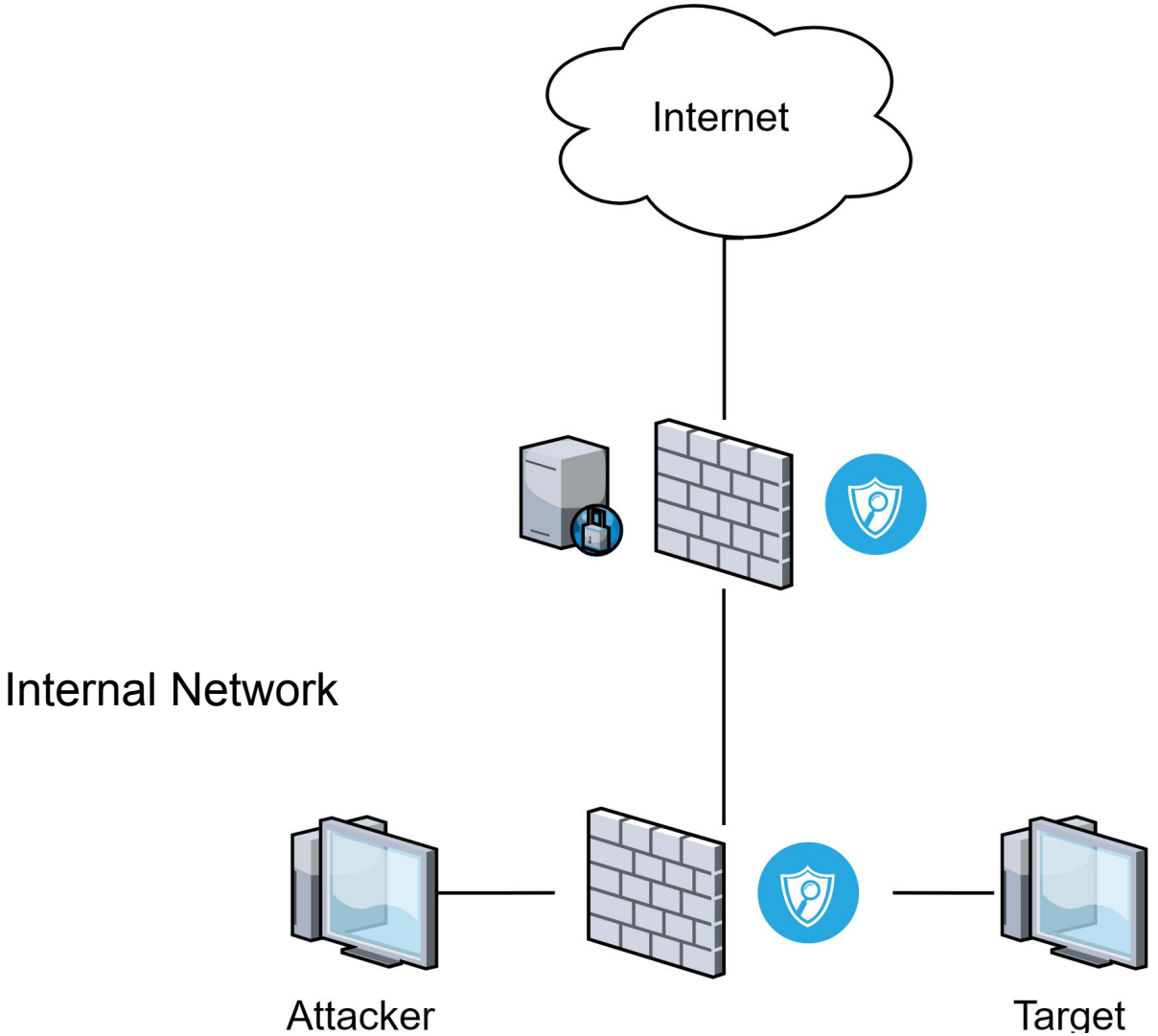
Don’t have admin credentials to target host

Attacker host is not located in the network for proper traffic logging

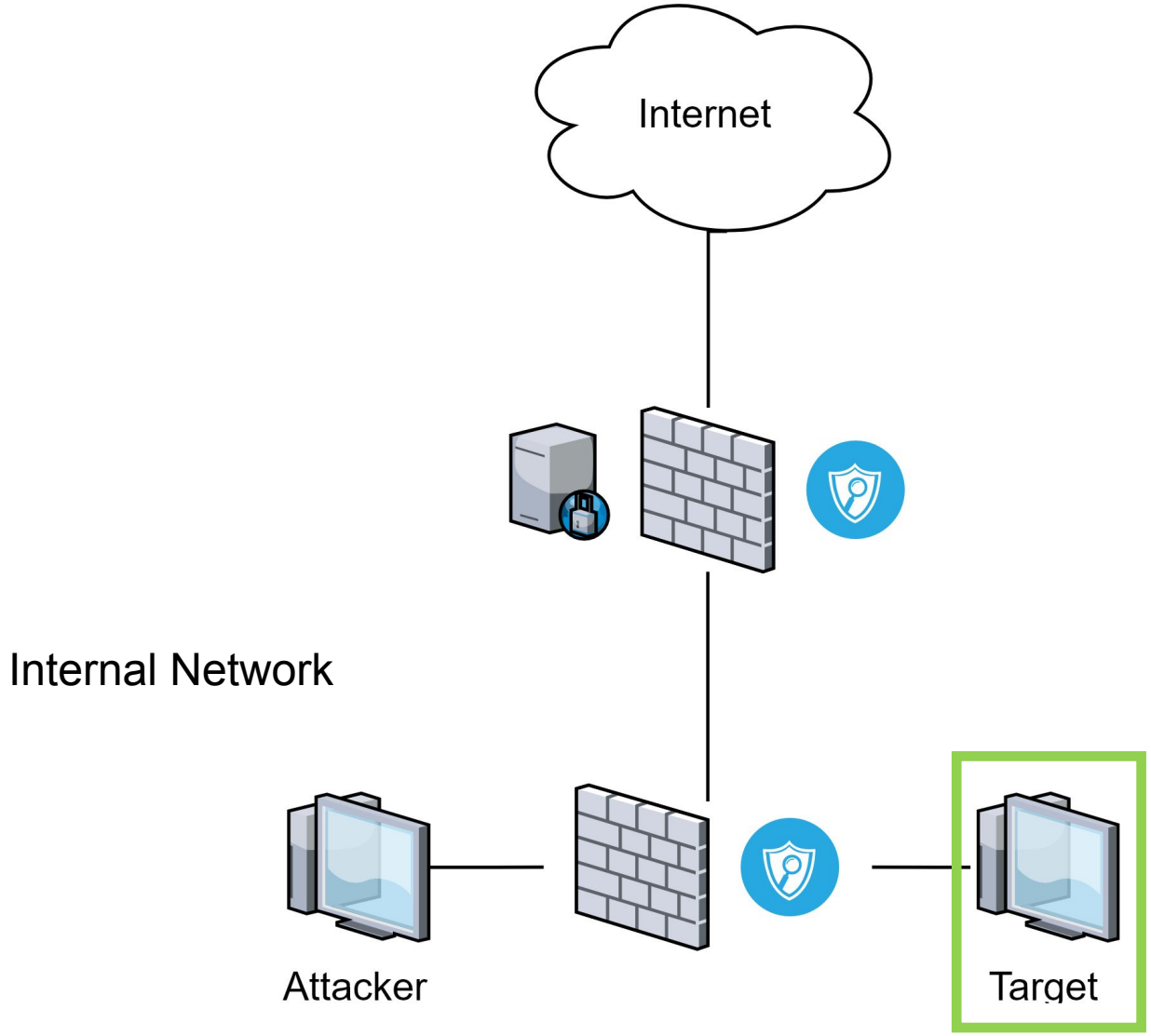
Attacker activity not properly emulated

Testing objectives are too broad

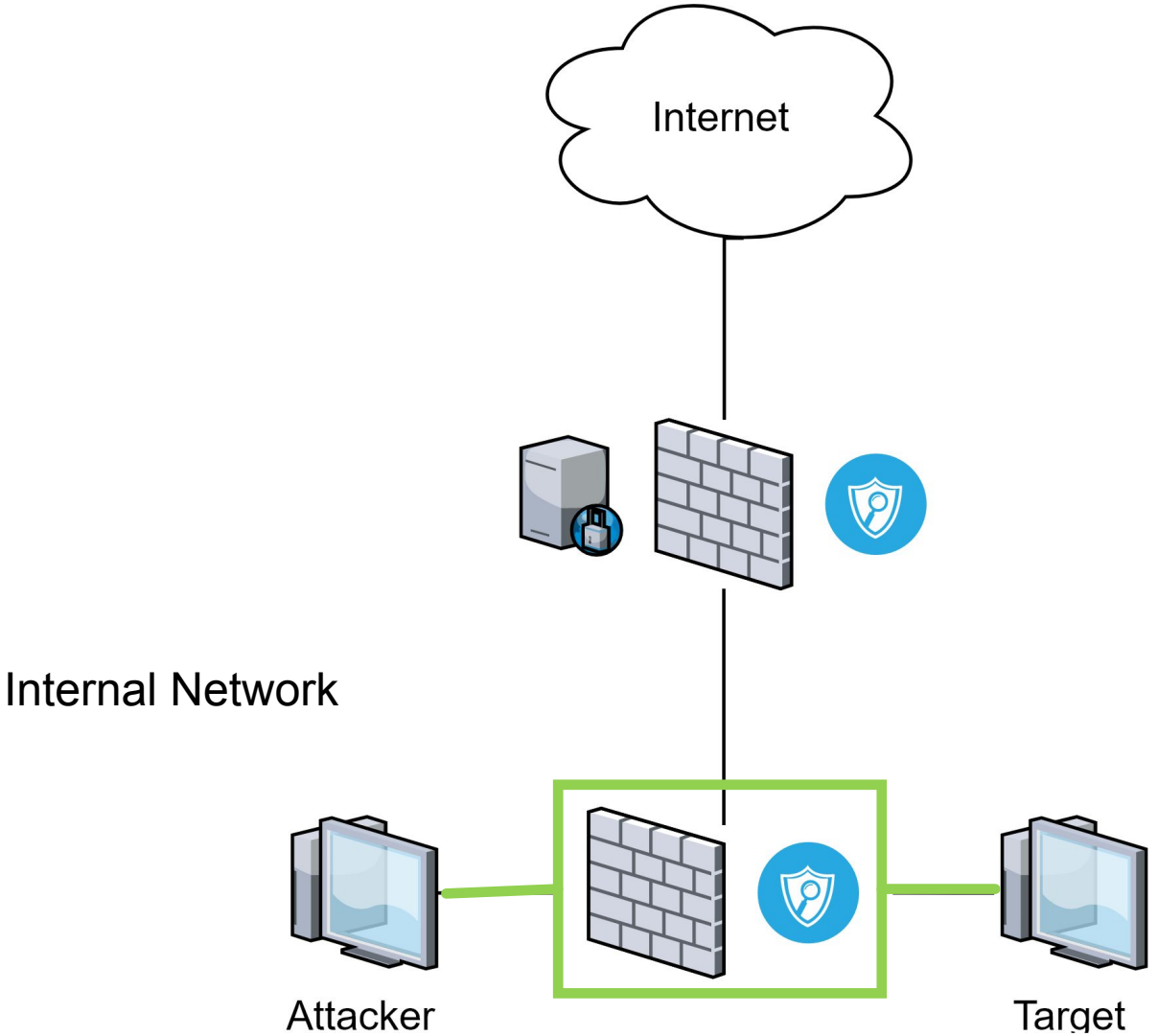
Ideal Testing Infrastructure



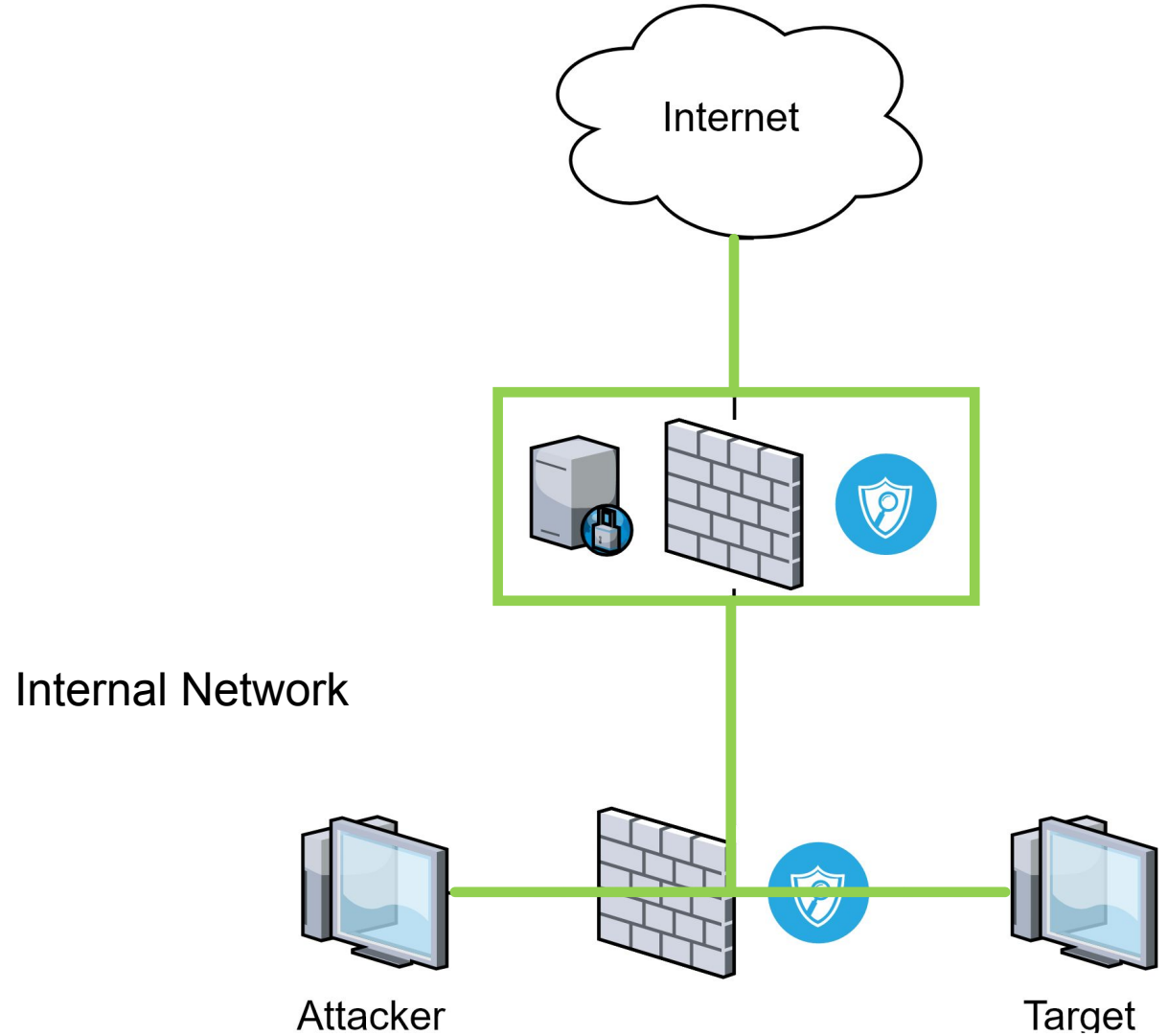
Accurate Target Host



Internal Logging Devices



Perimeter Logging Devices



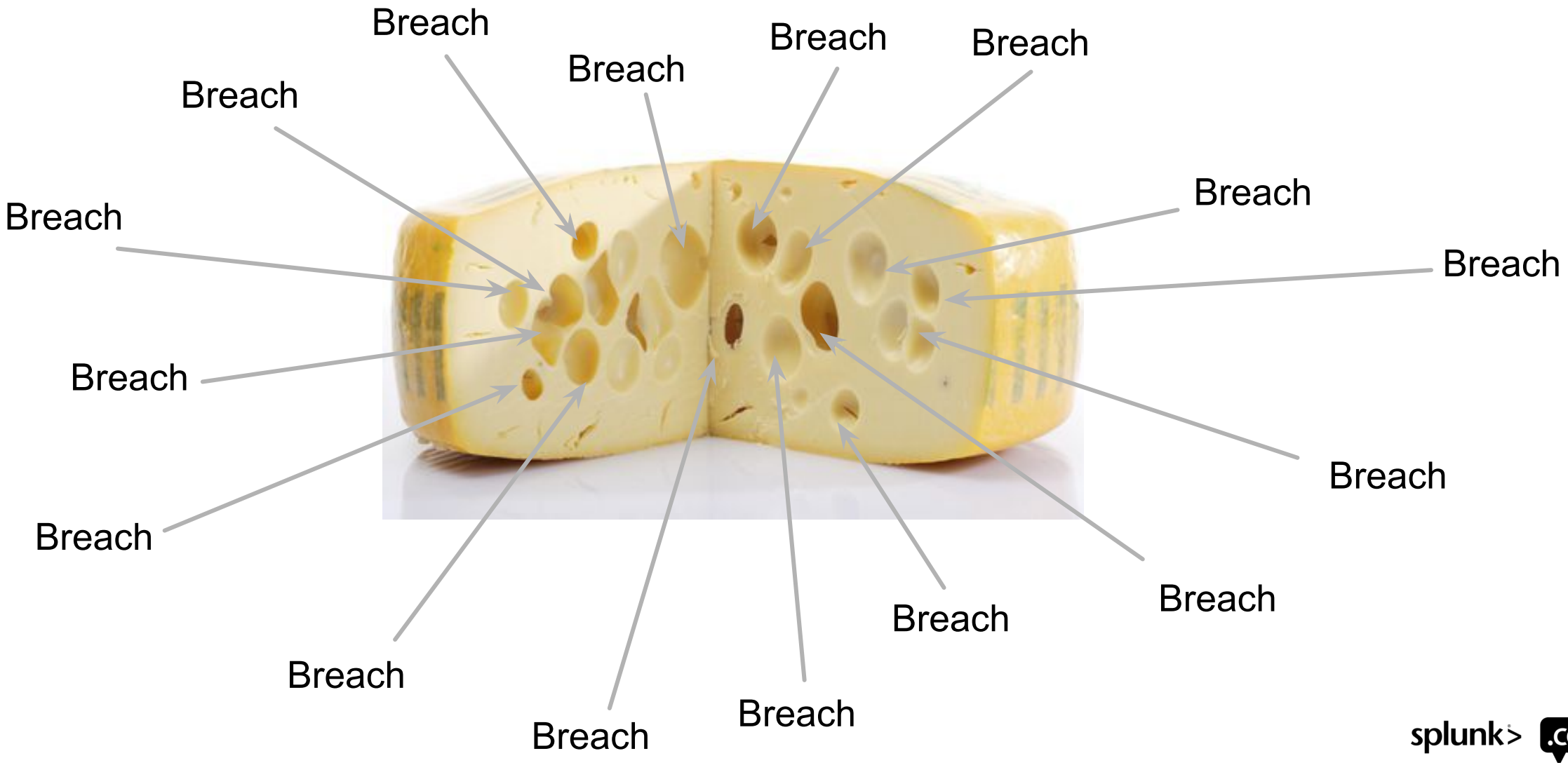
Testing Tips

1. Develop a detailed plan of what to test
2. Target single scenarios – don't boil the ocean
3. Have dedicated testers from offensive and defensive teams
4. **Test the concept, not the content**
5. Take it slow and be methodical

Recap

1. Are my log sources **deployed** widely enough to cover my environment?
2. Are my log sources **configured** properly to detect the threat?
3. Are my log sources even **capable** of detecting the threat?

The Real Risk



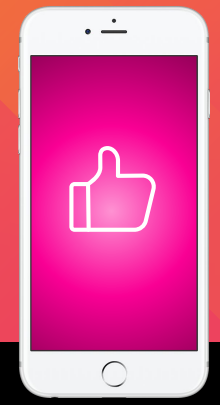
.conf19

splunk>

Thank You !

Go to the .conf19 mobile app to

RATE THIS SESSION



Appendix: Splunk Dashboard Source Code

```
<dashboard>  <label>Log Source Monitoring</label>  <row>  <panel>  <title>Total Active Percentage</title>  <single>  <search>  <query>| metasearch index=* sourcetype=* | stats dc(host) as active_host_count by sourcetype| lookup logsourcecounts.csv type as sourcetype| rename count as known_host_count| stats sum(active_host_count) as active_host_count sum(known_host_count) as known_host_count| eval total_perc=round((active_host_count/known_host_count)*100,0)| fields total_perc| rename total_perc as "Total Percentage Active"</query>  <earliest>-24h@h</earliest>  <latest>now</latest>  <sampleRatio>1</sampleRatio>  </search>  <option name="colorBy">value</option>  <option name="colorMode">block</option>  <option name="drilldown">none</option>  <option name="numberPrecision">0</option>  <option name="rangeColors">["0xd93f3c","0xf58f39","0xf7bc38","0x6db7c6","0x65a637"]</option>  <option name="rangeValues">[20,40,60,80]</option>  <option name="showSparkline">1</option>  <option name="showTrendIndicator">1</option>  <option name="trellis.enabled">0</option>  <option name="trellis.scales.shared">1</option>  <option name="trellis.size">medium</option>  <option name="trendColorInterpretation">standard</option>  <option name="trendDisplayMode">absolute</option>  <option name="unitPosition">after</option>  <option name="useColors">1</option>  <option name="useThousandSeparators">1</option>  </single>  </panel>  <panel>  <title>Total Count</title>  <single>  <search>  <query>| metasearch index=* sourcetype=* | stats dc(host) as active_host_count by sourcetype| lookup logsourcecounts.csv type as sourcetype| rename count as known_host_count| stats sum(active_host_count) as active_host_count sum(known_host_count) as known_host_count| eval total_count=active_host_count."/".known_host_count| fields total_count| rename total_count as "Total Count"</query>  <earliest>-24h@h</earliest>  <latest>now</latest>  <sampleRatio>1</sampleRatio>  </search>  <option name="colorBy">value</option>  <option name="colorMode">block</option>  <option name="drilldown">none</option>  <option name="numberPrecision">0</option>  <option name="rangeColors">["0x65a637","0x6db7c6","0xf7bc38","0xf58f39","0xd93f3c"]</option>  <option name="rangeValues">[0,30,70,100]</option>  <option name="showSparkline">1</option>  <option name="showTrendIndicator">1</option>  <option name="trellis.enabled">0</option>  <option name="trellis.scales.shared">1</option>  <option name="trellis.size">medium</option>  <option name="trendColorInterpretation">standard</option>  <option name="trendDisplayMode">absolute</option>  <option name="unitPosition">after</option>  <option name="useColors">1</option>  <option name="useThousandSeparators">1</option>  </single>  </panel>  </row>  <row>  <panel>  <title>Critical Log Source Reporting</title>  <single>  <search>  <query>| metasearch index=* sourcetype=* | stats dc(host) as active_host_count by sourcetype| lookup logsourcecounts.csv type as sourcetype| rename count as known_host_count| lookup critical_sourcetypes.csv sourcetype as sourcetype| where critical="y"| eval total_perc=round((active_host_count/known_host_count)*100,0)| fields sourcetype total_perc| rename total_perc as "Total Active Percentage"</query>  <earliest>-24h@h</earliest>  <latest>now</latest>  <sampleRatio>1</sampleRatio>  </search>  <option name="colorBy">value</option>  <option name="colorMode">block</option>  <option name="drilldown">none</option>  <option name="numberPrecision">0</option>  <option name="rangeColors">["0xd93f3c","0xf58f39","0xf7bc38","0x6db7c6","0x65a637"]</option>  <option name="rangeValues">[20,40,60,80]</option>  <option name="showSparkline">1</option>  <option name="showTrendIndicator">1</option>  <option name="trellis.enabled">1</option>  <option name="trellis.scales.shared">1</option>  <option name="trellis.size">medium</option>  <option name="trendColorInterpretation">standard</option>  <option name="trendDisplayMode">absolute</option>  <option name="unitPosition">after</option>  <option name="useColors">1</option>  <option name="useThousandSeparators">1</option>  </single>  </panel>  </row>  <row>  <panel>  <title>Critical Log Source Reporting Table</title>  <table>  <search>  <query>| metasearch index=* sourcetype=* | stats dc(host) as active_host_count by sourcetype| lookup logsourcecounts.csv type as sourcetype| rename count as known_host_count| lookup critical_sourcetypes.csv sourcetype as sourcetype| where critical="y"| fields - critical| eval total_perc=round((active_host_count/known_host_count)*100,0)| rename total_perc as "Total Active Percentage" active_host_count as "Active Host Count" known_host_count as "Known Host Count"| sort + "Total Active Percentage"</query>  <earliest>-24h@h</earliest>  <latest>now</latest>  <sampleRatio>1</sampleRatio>  </search>  <option name="count">100</option>  <option name="dataOverlayMode">none</option>  <option name="drilldown">none</option>  <option name="percentagesRow">false</option>  <option name="refresh.display">progressbar</option>  <option name="rowNumbers">false</option>  <option name="totalsRow">false</option>  <option name="wrap">true</option>  </table>  </panel>  </row>  <row>  <panel>  <title>Total Active Percentage by Sourcetype</title>  <single>  <search>  <query>| metasearch index=* sourcetype=* | stats dc(host) as active_host_count by sourcetype| lookup logsourcecounts.csv type as sourcetype| rename count as known_host_count| eval total_perc=round((active_host_count/known_host_count)*100,0)| fields sourcetype total_perc| rename total_perc as "Total Active Percentage"</query>  <earliest>-24h@h</earliest>  <latest>now</latest>  <sampleRatio>1</sampleRatio>  </search>  <option name="colorBy">value</option>  <option name="colorMode">block</option>  <option name="drilldown">none</option>  <option name="numberPrecision">0</option>  <option name="rangeColors">["0xd93f3c","0xf58f39","0xf7bc38","0x6db7c6","0x65a637"]</option>  <option name="rangeValues">[20,40,60,80]</option>  <option name="showSparkline">1</option>  <option name="showTrendIndicator">1</option>  <option name="trellis.enabled">1</option>  <option name="trellis.scales.shared">1</option>  <option name="trellis.size">medium</option>  <option name="trendColorInterpretation">standard</option>  <option name="trendDisplayMode">absolute</option>  <option name="unitPosition">after</option>  <option name="useColors">1</option>  <option name="useThousandSeparators">1</option>  </single>  </panel>  <panel>  <title>Largest Reporting Gap</title>  <table>  <search>  <query>| metasearch index=* sourcetype=* | stats dc(host) as active_host_count by sourcetype| lookup logsourcecounts.csv type as sourcetype| rename count as known_host_count| eval total_perc=round((active_host_count/known_host_count)*100,0)| where total_perc&amp;lt;40| rename total_perc as "Total Active Percentage" active_host_count as "Active Host Count" known_host_count as "Known Host Count"| sort + "Total Active Percentage"</query>  <earliest>-24h@h</earliest>  <latest>now</latest>  <sampleRatio>1</sampleRatio>  </search>  <option name="count">100</option>  <option name="dataOverlayMode">none</option>  <option name="drilldown">none</option>  <option name="percentagesRow">false</option>  <option name="rowNumbers">false</option>  <option name="totalsRow">false</option>  <option name="wrap">true</option>  </table>  </panel>  </row>  <row>  <panel>  <title>Reporting Log Source Table</title>  <table>  <search>  <query>| metasearch index=* sourcetype=* | stats dc(host) as active_host_count by sourcetype| lookup logsourcecounts.csv type as sourcetype| rename count as known_host_count| eval total_perc=round((active_host_count/known_host_count)*100,0)| rename total_perc as "Total Active Percentage" active_host_count as "Active Host Count" known_host_count as "Known Host Count"| sort + "Total Active Percentage"</query>  <earliest>-24h@h</earliest>  <latest>now</latest>  <sampleRatio>1</sampleRatio>  </search>  <option name="count">100</option>  <option name="dataOverlayMode">none</option>  <option name="drilldown">none</option>  <option name="percentagesRow">false</option>  <option name="refresh.display">progressbar</option>  <option name="rowNumbers">false</option>  <option name="totalsRow">false</option>  <option name="wrap">true</option>  </table>  </panel>  </row></dashboard>
```