

IETF  
Internet-Draft  
Intended status: Informational  
Expires: January 9, 2020

A. Taddei  
C. Wueest  
K. Roundy  
Symantec Corporation  
D. Lazanski  
Last Press Label  
July 08, 2019

**Capabilities and Limitations of an Endpoint-only Security Solution**  
**draft-taddei-smart-cless-introduction-01**

Abstract

In the context of existing, proposed and newly published protocols, this draft RFC is to establish the capabilities and limitations of endpoint-only security solutions and explore benefits and alternatives to mitigate those limits with the support of real case studies.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 9, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">3</a>
<a href="#">2.</a>	<a href="#">Abbreviations . . . . .</a>	<a href="#">4</a>
<a href="#">3.</a>	<a href="#">Definitions . . . . .</a>	<a href="#">5</a>
<a href="#">4.</a>	<a href="#">Disclaimer . . . . .</a>	<a href="#">6</a>
<a href="#">5.</a>	<a href="#">Endpoints: definitions, models and scope . . . . .</a>	<a href="#">6</a>
<a href="#">5.1.</a>	<a href="#">Internal representation of an endpoint . . . . .</a>	<a href="#">7</a>
<a href="#">5.2.</a>	<a href="#">Endpoints modeled in an end-to-end context . . . . .</a>	<a href="#">8</a>
<a href="#">6.</a>	<a href="#">Threat Landscape . . . . .</a>	<a href="#">9</a>
<a href="#">7.</a>	<a href="#">Endpoint Security Capabilities . . . . .</a>	<a href="#">11</a>
<a href="#">8.</a>	<a href="#">What would be a perfect endpoint security solution? . . . . .</a>	<a href="#">14</a>
<a href="#">9.</a>	<a href="#">The defence-in-depth principle . . . . .</a>	<a href="#">15</a>
<a href="#">10.</a>	<a href="#">Endpoint Security Limits . . . . .</a>	<a href="#">16</a>
10.1.	No possibility to put an endpoint security add-on on the UE . . . . .	<a href="#">17</a>
<a href="#">10.1.1.</a>	<a href="#">Not receiving any updates or functioning patches . . . . .</a>	<a href="#">18</a>
<a href="#">10.1.2.</a>	<a href="#">Mirai IoT bot . . . . .</a>	<a href="#">19</a>
<a href="#">10.2.</a>	<a href="#">Endpoints may not see the malware on the endpoint . . . . .</a>	<a href="#">20</a>
<a href="#">10.2.1.</a>	<a href="#">LoJax UEFI rootkit . . . . .</a>	<a href="#">20</a>
<a href="#">10.2.2.</a>	<a href="#">SGX Malware . . . . .</a>	<a href="#">21</a>
<a href="#">10.2.3.</a>	<a href="#">AMT Takeover . . . . .</a>	<a href="#">21</a>
<a href="#">10.2.4.</a>	<a href="#">AMT case study (anonymised) . . . . .</a>	<a href="#">22</a>
<a href="#">10.2.5.</a>	<a href="#">Users bypass the endpoint security . . . . .</a>	<a href="#">23</a>
<a href="#">10.3.</a>	<a href="#">Endpoints may miss information leakage attacks . . . . .</a>	<a href="#">23</a>
<a href="#">10.3.1.</a>	<a href="#">Meltdown/Specter . . . . .</a>	<a href="#">23</a>
<a href="#">10.3.2.</a>	<a href="#">Network daemon exploits . . . . .</a>	<a href="#">23</a>
<a href="#">10.3.3.</a>	<a href="#">SQL injection attacks . . . . .</a>	<a href="#">24</a>
<a href="#">10.3.4.</a>	<a href="#">Low and slow data exfiltration . . . . .</a>	<a href="#">24</a>
<a href="#">10.4.</a>	<a href="#">Suboptimality and gray areas . . . . .</a>	<a href="#">25</a>
<a href="#">10.4.1.</a>	<a href="#">Stolen credentials . . . . .</a>	<a href="#">25</a>
<a href="#">10.4.2.</a>	<a href="#">Zero Day Vulnerability . . . . .</a>	<a href="#">26</a>
<a href="#">10.4.3.</a>	<a href="#">Port scan over the network . . . . .</a>	<a href="#">26</a>
<a href="#">10.4.4.</a>	<a href="#">DDoS attacks . . . . .</a>	<a href="#">27</a>
<a href="#">11.</a>	<a href="#">Learnings from production data . . . . .</a>	<a href="#">28</a>
<a href="#">11.1.</a>	<a href="#">Endpoint only incidents . . . . .</a>	<a href="#">29</a>
11.2.	Security incidents detected primarily by network security products . . . . .	<a href="#">30</a>
<a href="#">11.2.1.</a>	<a href="#">Unauthorized external vulnerability scans . . . . .</a>	<a href="#">30</a>
<a href="#">11.2.2.</a>	<a href="#">Unauthorized internal vulnerability scans . . . . .</a>	<a href="#">31</a>
<a href="#">11.2.3.</a>	<a href="#">Malware downloads resulting in exposed endpoints . . . . .</a>	<a href="#">31</a>
<a href="#">11.2.4.</a>	<a href="#">Exploit kit infections . . . . .</a>	<a href="#">31</a>
<a href="#">11.2.5.</a>	<a href="#">Attacks against servers . . . . .</a>	<a href="#">32</a>
<a href="#">12.</a>	<a href="#">Regulatory Considerations . . . . .</a>	<a href="#">33</a>



<a href="#">12.1.</a>	<a href="#">IoT Security . . . . .</a>	<a href="#">33</a>
<a href="#">12.2.</a>	<a href="#">Network infrastructure . . . . .</a>	<a href="#">34</a>
<a href="#">12.3.</a>	<a href="#">Auditing and Assessment . . . . .</a>	<a href="#">34</a>
<a href="#">12.4.</a>	<a href="#">Privacy Considerations . . . . .</a>	<a href="#">35</a>
<a href="#">13.</a>	<a href="#">Human Rights Considerations . . . . .</a>	<a href="#">35</a>
<a href="#">14.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">35</a>
<a href="#">15.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">35</a>
<a href="#">16.</a>	<a href="#">Informative References . . . . .</a>	<a href="#">35</a>
<a href="#">Appendix A.</a>	<a href="#">Contributors . . . . .</a>	<a href="#">40</a>
	<a href="#">Authors' Addresses . . . . .</a>	<a href="#">41</a>

## **[1.](#) Introduction**

This Internet Draft aims to be a reference to the designers of protocols on the capabilities and limitations of security solutions on endpoint devices against malware and other attacks. As security is entering a new phase in the arms race between attackers and defenders, with many technical, economic and regulatory changes, and with a significant increase in major data breaches, it is a good moment to propose a systematic review and update on what is an old and constantly evolving problem: endpoint security.

With the above context in mind this document will focus on the capabilities and limitations of an endpoint-only security solution.

We want to explore a number of questions:

- o What endpoint models do we have?
- o What is the threat landscape under consideration?
- o Can we differentiate security and privacy threats?
- o What are common endpoint security capabilities?
- o What would be an ideal endpoint security solution?
- o What are the limits to endpoint security?
- o What is real production data telling us?
- o What can defence-in-depth help us with?
- o What are the economic considerations?
- o What are the regulatory considerations and constraints?
- o What are the human rights considerations?



Our goal with this review is to describe the benefits and limitations of endpoint security in the real world, rather than in the abstract. We aim to highlight security limitations that cannot be addressed by endpoint solutions and to suggest how these may be mitigated with the concept of a defence-in-depth approach, in order to increase the resilience against attacks and data breaches.

## **2. Abbreviations**

In this section we provide main abbreviations expansions

ABAC Attribute Based Access Control

AI Artificial Intelligence

AMT Active Management Technology

C&C Command and Control

CFI Control Flow Integrity

CFG Control Flow Guard

DDoS Distributed Denial of Service

DEP Data Execution Prevention

DGA Domain Generating Algorithms

DLP Data Loss Prevention

DMARC Domain-based Message Authentication, Reporting and Conformance

DoS Denial of Service

EE Execution Environment

EDR Endpoint Detection and Response

EPP Endpoint Protection Platform

FP False Positive

HIPS Host Intrusion Prevention System

ICD Integrated Cyber Defence

ICMP Internet Control Message Protocol



IDS Intrusion Detection System

IoT Internet of Things

IPS Intrusion Prevention System

ML Machine Learning

MSS Managed Security Services

MSSP Managed Security Services Provider

NIST National Institute of Standards and Technology

NX No Execute Bit

P2P Peer to Peer

RAP Reuse Attack Protector

RBAC Role Based Access Control

RDP Remote Desktop Protocol

ROP Return Oriented Programming

SANS System Administration, Networking, and Security

SGX Software Guard eXtensions

SSH Secure SHell

UE User Equipment

UEFI Unified Extensible Firmware Interface

UX User Experience

VM Virtual Machine

XSS Cross Site Scripting

### **3. Definitions**

In this section we provide definitions that are marked

- o (L) Local to this document





- o (G REFERENCE) Global and then will be preceded by a reference

DoS (L) Literally a Denial of Service. Not to be confused with a Network DoS or DDoS.

Endpoint security capabilities (L) How to protect the endpoint with three different aspects of protection:

- o Prevention - The attack doesn't succeed by intrinsic or explicit security capabilities.
- o Detection - The attack is happening or has happened and is recorded and/or signalled to another component for action.
- o Mitigation - Once detected, the attack can be halted or its effects can at least be reduced or reversed.

System (L) A system is a heterogeneous set of any IT capabilities including hardware, software, endpoints (including IoT), networks, data centers and platforms with no assumptions on deployment form factor (physical, virtual, microservices), deployment scenario, geographic distribution, or dispersion.

User Equipment (G ITU-T H.360) Equipment under the control of an End User

#### **4. Disclaimer**

This document is a first draft and is incomplete on purpose. Indeed there are several areas where there are different ways to develop this draft and the authors are seeking for feedback and extended collaboration. This is to be noted too, that this is the first draft RFC for the authors and contributors, so, coaching and help will be appreciated. Overall, 'a bon entendeur, salut'.

Comments are solicited and should be addressed to the authors.

#### **5. Endpoints: definitions, models and scope**

Endpoints are the origin and destination for a communication between parties. This encompasses User Equipment (UE) and the Host at the other end of the communication. Whilst it is recognized that these two ends of the communication may represent a vast amount of diverse endpoints, this document will set here a requirement for a uniform way to describe the endpoints in order to work from an equally uniform representation of what is called the Attack Surface. In the same spirit as the IETF TEEP Working Group generalized its work, see [[TEEP](#)], this document will rely on another document identified as



[I-D.[draft-mcfadden-smart-endpoint-taxonomy-for-cless-00](#)] in order to represent the taxonomy of endpoints.

For example:

- o The following would be considered as UEs: a smartphone, a smart device, any IoT device, a laptop, a desktop, a workstation, etc.
- o Hosts represent too, physical servers, virtual servers/machines, etc.

We require a framework in order to define and model the endpoint itself and the position of the endpoint in the network. In this initial analysis we focus on endpoints that are User Equipment (UE) rather than on hosts. In the future, with the help of [I-D.[draft-mcfadden-smart-endpoint-taxonomy-for-cless-00](#)] we hope to balance and unify the model.

In addition, we need two models for the endpoint, internally and in an end-to-end context within the network. With this approach we expect both models to help us cover all the attack surface and the threat landscape and therefore help us list the capabilities and limitations for endpoint security.

Indeed, this will help us understand point attacks versus composite attacks within context, and, accordingly, understand holistically the capabilities and the limitations of endpoint security. For example to differentiate when only an application on the end point is affected.

### **5.1. Internal representation of an endpoint**

This section interfaces here with [I-D.[draft-mcfadden-smart-endpoint-taxonomy-for-cless-00](#)] which starts from the below internal representation of an endpoint which could be generalized by the simple diagram below:

```
+-----+
| Application           |
+-----+
| OS / Execution Environment |
+-----+
| Hardware              |
+-----+
```



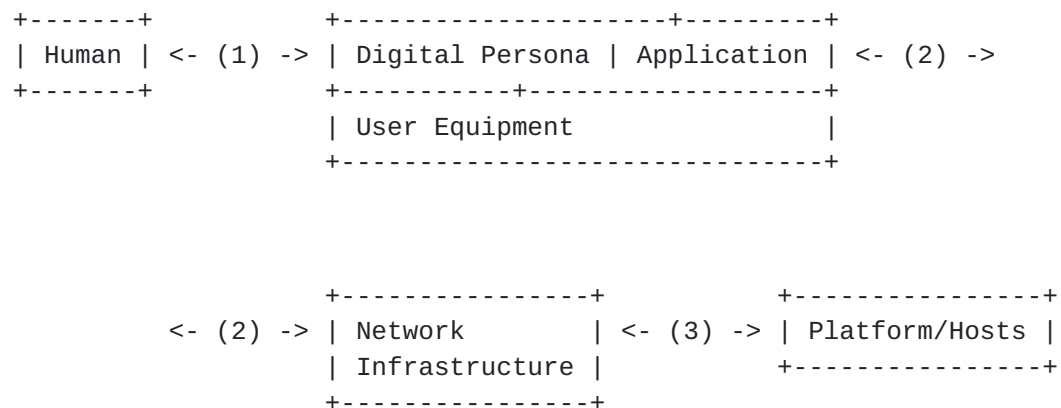
Today there are many combinations of Hardware, OS/EE pairing and Application layers, offering the user a vast set of features with a wide spectrum of capabilities.

Furthermore we can consider that an application running on a UE or a host is an endpoint too, so we have multiple ways to read the above diagram.

In essence we want to consider here endpoints including those which have a variance in electrical power, computational power, memory, disk, network interfaces, size, ownership, connectics, etc. and therefore why we rely on [\[I-D.draft-mcfadden-smart-endpoint-taxonomy-for-cless-00\]](#).

## 5.2. Endpoints modeled in an end-to-end context

A representation of endpoints in an end-to-end context could look like the following diagram:



1. Humans have a user experience (UX) with the UE, starting with an explicit or implicit Digital Persona, engaging with an application
2. The application will have sessions through a large Network Infrastructure where we do not assume anything of the infrastructure (could be landlines, mobile networks, satellites, etc.) and those sessions reach
3. a Platform consisting of many Hosts either physical or virtual and it ensures a large part of the end-to-end user experience.



In this end-to-end model we see that many other systems may have interactions with the UE: the human, the UX, the digital persona, the sessions, the intermediate network infrastructure, and the hosts and application at the destination.

If we now look at security aspects of the above models, the threat landscape is very large and the attack surface will cover all the components and interactions at any level.

## 6. Threat Landscape

(Editor's note: this section will require a significant amount of future development.)

Given the vast number of combinations that the above generic modeling offers us, defining a threat landscape should be done carefully and will require a systematic methodology.

Therefore this entire section will be developed through future iterations of the document, in this initial version we will start structuring an approach and then adjust this based on feedback.

There is no doubt that we want to cover typical known attacks such as:

- o Malware (Trojans, viruses, backdoors, bots, etc.)
- o Adware and spyware
- o Exploits
- o Phishing
- o Script based attacks
- o Ransomware, local Denial of Service (DoS) attacks
- o Denial of Service (DoS) attacks
- o Malicious removable storage devices (USB)
- o In memory attacks
- o Rootkits and firmware attacks
- o Scams and online fraud
- o System abuse (staging/proxying)





- o etc.

To illustrate the difficulty to define a good threat landscape, when it comes to cryptojacking and coinmining that were on the rise, in which category do they fall: malware? DoS? system abuse? or a category on its own?

This is why we wanted to conduct a thorough gap analysis using existing definitions and frameworks, but we couldn't find an existing comprehensive and recognized taxonomy dedicated to the threat landscape on endpoints. We found however different models in this field, and have considered two. We are open to further suggestions.

Indeed both of the analysed frameworks contain threat landscape descriptions:

- o MITRE Common Attack Pattern Enumeration Classification (CAPEC). See [[CAPEC](#)].
- o MITRE ATT&CK. See [[ATTACK](#)].

These offer us interesting ways to assess the threat landscape:

- o CAPEC offers a hierarchical view of attack patterns by domains which can match some aspects of both of our above models, but we will need to identify those attacks that fit exactly in our scope.
- o ATT&CK offers a very straightforward categorized knowledge base of attacks, but it concentrates on the enterprise attack chain, so we will need to do some work to extract what we need.

We recognise however that these frameworks do not address all of the threats that can affect the security of a system, for example they do not cover; routing hijacking, flooding, selective blocking, unauthorised modification of data sent to an endpoint, etc. Further work to define categories of threats is therefore required.

As a further example, phishing should be included as an attack, but whilst this is indeed an attack that will materialize on a device through an application (email, webmail, etc.), the real target of this attack is not the device, but the human behind the digital persona.

Having a methodology of assessment is necessary here, because it will help decide what is in scope vs. out of scope.

We are aware that once a method and the categories are fully defined in this section, it will force a review of all the following sections



in the document. Whilst remapping will be necessary, it should not drastically change the draft.

## **7. Endpoint Security Capabilities**

In this section we try to define some endpoint security capabilities (Editor's note: this section will require future development.)

In this version of the document we will start by developing a framework to categorize and position endpoint security capabilities with the goal of defining what an ideal endpoint security capability would look like.

By endpoint security capabilities we mean how to protect the endpoint against attacks. Protection has many meanings, we want to distinguish three different aspects of protection:

- o Prevention - The attack doesn't succeed by intrinsic or explicit security capabilities.
- o Detection - The attack is happening or has happened and is recorded and/or signalled to another component for action.
- o Mitigation - Once detected, the attack can be halted or its effects can at least be reduced or reversed.

For example, prevention methods include keeping the software updated and patching vulnerabilities, implementing measures to authenticate the provenance of incoming data to stop the delivery of malicious content, or choosing strong passwords. Detection methods include inspecting logs or network traffic. Mitigation could include deploying backups to recover from an attack with minimal disruption.

Our intention however is not just to consider each endpoint security capability separately, but also the overall endpoint security holistically with all its interdependencies. Indeed, we defined a simple endpoint, but each layer may or may not have a certain spectrum of intrinsic capabilities and there may be multiple ways to provide add-on and third-party endpoint security capabilities, allowing complex interactions between all of these components.

We define two different aspects of endpoint security capabilities and their subdivisions as:

- o (A) Intrinsic security capability can be built-into each of the endpoint model layers

- \* (1) Hardware



- \* (2) OS/EE
- \* (3) Application
- o (B) Add-on security capability can be
  - \* (4) a component of the hardware
  - \* (5) a component of the OS/EE
  - \* (6) an application by itself

In (A) we relate to a 'security by design' intention of the developers and they will intrinsically offer a security model and security capabilities as part of their design. A typical example of this is the authorization model.

In (B) a 3rd party is offering an additional security component which was not necessarily considered when the Hardware, OS/EE or Application were designed.

In the future we will review all the main categories of security capabilities that are known to date and assess security capability enablers like Artificial Intelligence (AI) and Machine Learning (ML). For each category we will try to give a review on how effective the capability is in securing the system.

With regard to (6), there are many available options for add-on security capabilities offered by third-parties as applications on a commercial or open-source basis. Gartner (see [[GARTNERREPORT](#)]) highlights the evolution of endpoint security towards two directions as shown in [[EPPEDR](#)], [[EPPSECURITY](#)], [[EPPGUIDE](#)].

- o Endpoint Protection Platform (EPP) as an integrated security solution designed to detect and block threats at the device level.
- o Endpoint Detection and Response (EDR) as a combination of next generation tools to provide anomaly detection and alerting, forensic analysis and endpoint remediation capabilities.

Among the security capabilities that we list, the endpoint can perform the following:

- o Intrinsic
  - \* Software updates / patching
  - \* Access Control (RBAC, ABAC, etc.)



- \* Authentication
- \* Authorization
- \* Detailed event logging
- o Execution protection
  - \* Exploit mitigation (file/memory)
  - \* Tamper protection
  - \* Whitelisting filter by signatures, signed code or other means
  - \* System hardening and lockdown (HIPS, trusted boot, etc.)
- o Malware protection
  - \* Scanning - on access/on write/scheduled/quick scan (file/memory)
  - \* Reputation-based blocking on files or by ML
  - \* Behavior-based detection - (heuristic based/ML)
  - \* Rootkit and firmware detection
  - \* Threat intelligence based detection (cloud-based/on premise)
  - \* Static detection - generic, by emulation, by ML, by signature
- o Attack/Exploit/Application Protection
  - \* Application protection (browser, messaging clients, social media, etc.)
    - + Disinformation Protection (anti-phishing, fake news, anti-spam, etc.)
    - + Detection of unintended link location (URL blocklist, etc.)
    - + Memory exploit mitigation, e.g. browsers
  - \* Network Protection (local firewall, IDS, IPS and local proxy) inbound and outbound
  - \* Detection of network manipulation (ARP, DNS, etc.)





- \* Data Loss Prevention and exfiltration detection (incl. covert channels)

## **8. What would be a perfect endpoint security solution?**

With all the above knowledge, let's consider what we could expect from a perfect endpoint security 'system'. It would:

- o find instantly accurate reputation for any file before it gets executed and block it if needed.
- o monitor any behavior on the endpoint, including inbound and outbound network traffic, learn and identify normal behavior and detect and block malicious actions, even if the attack is misusing legitimate clean system tools or hiding with a rootkit.
- o patch instantly across all devices/systems/OSes, including virtual patching, meaning you can patch or shield an application even before an official patch is released.
- o exploit protection methods for all processes where applicable, e.g. no execute bit (NX), data execution prevention (DEP), address space layout randomization (ASLR), Control Flow Integrity Guard (CFI/CFG), stack canaries, shadow stack, reuse attack protection (RAP), etc. all of which are methods, which make it very difficult to successfully run any exploit, even for zero day vulnerabilities.
- o detect attempts to re-route data to addresses other than those which the user intended, e.g. detect incorrectly served DNS entries, TLS connections to sites with invalid certificates, data that is being proxied without explicit user consent, etc.
- o have an emulator/sandbox/micro virtualization to execute code and analyse the outcome and perform a roll back of all actions if needed, e.g. for ransomware.
- o allow the endpoint to communicate with the other endpoints in the local network and globally, to learn from 'the crowd' and dynamically update rules based on its findings.
- o be in constant sync with all other endpoints deployed on a network and other security solutions, run on any OS, with no delay (including offline modes and on legacy systems).
- o run from the OS/EE when possible.



- o run as one of the first process on the OS/EE and protect itself from any form of unwanted tampering.
- o offers a reliable logging that can't be tampered with, even in the event of system compromise.
- o receive updates instantly from a trusted central entity.

## **9. The defence-in-depth principle**

In this section we give a high level view of what we mean by 'defence-in-depth'.

Whilst endpoint security systems have good capabilities, sometimes it is debatable and perhaps suboptimal to let the endpoint run the capability alone or at all. It is generally considered good security practice to adopt a defence-in-depth approach (see [[USCERT](#)]). The Open Web Application Security Project group (OWASP) describes the concept as follows: "The principle of defense-in-depth is that layered security mechanisms increase security of the system as a whole. If an attack causes one security mechanism to fail, other mechanisms may still provide the necessary security to protect the system." (see [[OWASP](#)])

Indeed there are many other constituencies as per our end-to-end model that can participate in the defence process: The network, the infrastructure itself, the platform, the human, the user experience and in a hybrid of an on premise and cloud approach, an Integrated Cyber Defence (ICD) of the entire chain.

The simple idea behind the concept is that "every little helps". If the endpoint is not 100% secure itself, the detection chance can increase with additional security capabilities from other entities. We acknowledge that there are some case where adding an additional component to the system may degrade the overall security level by introducing new weaknesses.

There are various reference article in the industry highlighting limitations of endpoint only solutions. For example this quote here, which talks about multi-tier solutions: "There are limitations with any endpoint protection solution, however, that can limit protection to only the client layer. There is also a need for security above the client layer, as endpoint protection products cannot intercept traffic. Vendors will often sell a multi-tiered solution that enables a network appliance to assist the endpoint protection client by intercepting traffic between the attacker and the infected client. Vendors will also sell solutions that monitor and intercept traffic on internal or external network segments to protect the enterprise



from these threats. A prime example of the limitations of endpoint protection software is infection via a phishing attack." [[ADAPTURE](#)].

Some sources point out that even the best solution might not get deployed in the optimal way in a real world scenario as the environment can be very complex: "While endpoint security has improved significantly with the introduction of application whitelisting and other technologies, our systems and devices are simply too diverse and too interconnected to ensure that host security can be deployed 100% ubiquitously and 100% effectively." [[NETTODAY](#)]

On these grounds it is considered a good idea to follow a layered approach when it comes to security. "In today's complex threat environment, companies need to adopt a comprehensive, layered approach to security, which is a challenging task in such as rapidly evolving, crowded market." [[HSTODAY](#)]

It is important to comprehend the capabilities of endpoint security solutions in this overall picture of the connected environment, which includes other systems, networks and various protocols that are used to interact with these entities. Understanding possible shortcomings from single layered solutions can help counterbalance such weaknesses in the architectural concept or the protocol design.

In order to quantify any potential benefits or limitations of the various layered scenarios in regards to security a solid data set is needed. This section requires statistics about proportions of attacks that go undetected in various cases. We propose analysing data for the following four cases:

- o There is no security solution
- o Security is only on the endpoint
- o Security is only on the network
- o Security is on both the endpoint and the network

However reconciling various statistics requires a lot of caution and time, a methodology and consistent classification to avoid any misinterpretation.

## **[10.](#) Endpoint Security Limits**

The previous section defines an ideal endpoint security 'system', however, from the real world, the expectation of what we can get from



an endpoint security solution will look more along the following lines:

- o may not be able to run at full capacity due to computational power limits, battery life, performance, or policies (such as BYOD restrictions in enterprise networks), etc.
- o may not be able to run at full capacity as it slows down performance too much.
- o will miss some of the malware or attacks, regardless of detection method used, like signatures, heuristics, machine learning (ML), artificial intelligence (AI), etc.
- o have some level of False Positives (FP).
- o not monitoring or logging all activities on the system, e.g. due to constraints of disk space or when a clean windows tool is being triggered to do something malicious but the activity is not logged. Such activity can be logged, but a decision needs to be made if it's clean or not.
- o have its own vulnerabilities or simple instabilities that could be used to compromise the system.
- o be tampered with by the user, e.g. disabled or reconfigured.
- o be tampered with by the attacker, e.g. exceptions added or log files wiped.

In the section below we review a number of these limitations through real examples, step by step. Some limitations are absolute, and some limitations result in a grey area or suboptimality for the solution.

#### **10.1. No possibility to put an endpoint security add-on on the UE**

UEs will vary a lot; by 2022, an estimated 29 billion devices will be connected, with 18 billion of them related to IoT [[ERICSSON](#)]. Many IoT products lack the capacity to install any endpoint security capabilities, are unable to update the software, and it is not possible to force the UE provider to improve or even offer an intrinsic security capability.

We acknowledge that the numbers do vary significantly depending on the source, for example:

- o [[STATISTA1](#)] is showing the current trajectory of IoT devices from 25B to date to 40+B in 2022 and 75B in 2025.





- o [\[ERICSSON\]](#) is more conservative and might requires an update, but it was reaching 29B devices in 2022, with a nice breakdown between device types and connectivity.
- o [\[STATISTA2\]](#) is showing a breakdown by verticals and is even more conservative than both of the above.
- o [\[ENISA\]](#) it refers to a [\[GARTNERIOT\]](#) report from 2017 which sets a trajectory to 20B devices by 2020.

In IoT we find UEs such as medical devices which are limited by regulation, welding robots that can't be slowed down, smart light bulbs which are limited by the processing power, etc. There are many factors influencing whether endpoint security can be added to a UE:

- o The UE is simply not powerful enough or the performance hit is too high.
- o Adding your own security will breach the warranty or will invalidate a certification or a regulation (breach of validity).
- o The UE needs to run in real-time and any delay introduced by a security process might break the process.
- o Some UEs are simply locked by design and the manufacturer does not provide a security solution (e.g. smart TV, fitness tracker or personal artificial assistants) see [\[CANDID1\]](#), [\[CANDID2\]](#).

In the future, a possible research problem would be to find hard data on the exact proportion of IoT devices that are unable to run any endpoint security add-on or that have no intrinsic security built-in.

The other hidden dimension here is the economical aspect. Many manufacturer are reluctant to invest in IoT device security, because it can significantly increases the cost of their solution and there is the perception that they will lose market shares, as customers are not prepared to pay the extra cost for added security.

#### **10.1.1. Not receiving any updates or functioning patches**

The endpoint security system may lack a built-in capability to be patched or it may be connected to a network that prevents the process of downloading updates automatically. For example stand-alone medical systems or industrial systems in isolated network segments often do not have a communication channel to the Internet.

Even if security updates are received, they typically will only be periodically updated; hence there will be a window of opportunity for



an attacker, between the time the attack is first used, and the time the attack is discovered/patched and the patch is deployed.

In addition updates and patches may themselves be malicious by mistake, or on purpose if not properly authenticated, or if the source of the updates has malicious intent. This could be part of a software update supply chain attack or an elaborate attacker breaking the update process, as for example seen with the Flamer group (see [\[FLAMER\]](#)).

A recent survey found that fewer than 10% of consumer IoT companies follow vulnerability disclosure guidelines at all, which is regarded as a basic first step in patching vulnerabilities (see [\[IOTPATCHING\]](#)). This indicates that many IoT devices do not have a defined update process or may not even create patches for most of the vulnerabilities.

Furthermore some endpoints system may reach the end of their support period and therefore no longer receive any updates for the OS/EE or the security solution due to missing licenses. However the systems may remain in use and become increasingly vulnerable as time goes on and new attacks are discovered.

#### **[10.1.2.](#) Mirai IoT bot**



+-----+-----+	
Description	A Mirai bot infecting various IoT devices through
	weak passwords over Telnet port TCP 23 and by using
	various vulnerabilities, for example the SonicWall
	GMS XML-RPC Remote Code Execution Vulnerability
	(CVE-2018-9866) on TCP port 21009. Once a device is
	compromised it will scan for further victims and
	then start a DoS attack.
+-----+-----+	
Simplified	Compromised device scans network for multiple open
attack	ports, attempts infection through weak password and
process	exploits, downloads more payload, starts DoS
	attack.
UE	No security tool present on majority of IoT
	devices, hence no detection possible. If a
	rudimentary security solution with limited
	capabilities such as outgoing firewall is present
	on the IoT device e.g. router, then it might be
	able to detect the outbound DoS attack and slow it
	down.
References	[ <a href="#">MIRAI1</a> ][MIRAI2]
+-----+-----+	

## [10.2.](#) Endpoints may not see the malware on the endpoint

### [10.2.1.](#) LoJax UEFI rootkit



Description	A device compromised with the LoJax UEFI rootkit, which is active before the OS/EE is started, hence before the endpoint security is active. It can pass back a clean 'image' when the security solution tries to scan the UEFI. Infection can either happen offline with physical access or through a dropper malware from the OS/EE.
UE	A perfect endpoint security could potentially detect the installation process if it is done from the OS/EE and not with physical modification or in the factory. Once the device is compromised the endpoint security solution can neither detect nor remove the rootkit. The endpoint solution may detect any of the exhibited behaviour, for example if the rootkit drops another malware onto the OS/EE at a later stage.
Reference	[ <a href="#">LOJAX</a> ]

#### [10.2.2.](#) SGX Malware

Description	Malware can hide in the Intel Software Guard eXtensions (SGX) enclave chip feature. This is a hardware-isolated section of the CPU's processing memory. Code running inside the SGX can use return-oriented programming (ROP) to perform malicious actions.
UE	Since the SGX feature is by design out of reach for the OS/EE, an endpoint security solution can neither detect nor remove any injected malware. A perfect endpoint security solution could potentially detect the installation process if it is done from the OS/EE and not with physical modification or in the factory.
References	[ <a href="#">SGX1</a> ] [ <a href="#">SGX2</a> ]

#### [10.2.3.](#) AMT Takeover





Description	A targeted attack group can remotely execute code on a system through the Intel AMT (Active Management Technology) vulnerability (CVE-2017-5689) over TCP ports 16992/16993. This provides full access to the computer, including remote keyboard and monitor access. The attacker can install malware, modify the system or steal information.
UE	The AMT is accessible even if the PC is turned off. Therefore any endpoint security software installed on the OS, would not be able to see this traffic and therefore also not able to detect it.
References	[ <a href="#">AMT1</a> ], [ <a href="#">AMT2</a> ]

#### **10.2.4. AMT case study (anonymised)**

An enterprise has a data center containing very sensitive data. Their workstations use a certain Intel chipset which integrates the AMT feature for remote computer maintenance. AMT is an interface for hardware management of the workstations, including transmission of screen content and keyboard and mouse input for remote maintenance. Communication with the management workstation is implemented by AMT through the network interface card (NIC) on the motherboard. The network packets generated in this way are invisible both to the main processor and thus to the OS running on the workstation. In autumn of 2015, it became known that some AMT-enabled computers had a flaw that allowed AMT's remote maintenance component to be activated and configured by attackers. This also worked when the workstations were switched off. The leakage of data through this vulnerability is elusive and difficult to detect. The identified threat situation led the organization to a new requirement implementing a method that can reliably detect this and similar vulnerabilities. In particular, the detection of rootkits and manipulated firmware, and this includes also (UEFI) BIOS - has also been a focus of their attention.

The method used as a solution, compares the desired data packets generated by a client operating system - the user, with the data packets received on the switch port. If more data has been received on the switch port than was been sent by the operating system - the user, there is a strong possibility that something bad is happening - like for example an infection via modified firmware or by rootkit.



#### **10.2.5. Users bypass the endpoint security**

Description	Endpoint security systems should not interfere with the normal operation of the endpoint to the extent that users become frustrated and want to disable them or configure them to disable a significant fraction of important security capabilities.
UE	Add-on endpoint security is now bypassed or disabled by the user. Unless the endpoint is under monitored management or can prevent a user from modifying the configuration, then this is shutting down a significant fraction of the security capabilities.
References	<a href="#">[NINESIGNS]</a>

#### **10.3. Endpoints may miss information leakage attacks**

Another aspect that endpoint security has issues in detecting are information disclosure or leakage attacks, especially on shared virtual/physical systems.

##### **10.3.1. Meltdown/Specter**

The Meltdown/Specter vulnerabilities and all its variants may allow reading of physical memory belonging to another virtual machine (VM) on the same physical system. This could reveal passwords, credentials, certificates etc. The trick is that an attacker can spin up his own VM on the same physical hardware. As this VM is controlled by the attacker, they will ensure that there is no endpoint security that detects the Meltdown exploit code when run. It is very difficult for the attacked VM to detect the memory read-outs. For know CPU vulnerabilities there are software patches available than can be applied. If it is an external service provider, it might not be in the power of the user to patch the physical system or to determine if this has been done by the provider.

##### **10.3.2. Network daemon exploits**

Other attack types, which leak memory data from a vulnerable web server, are quite difficult to detect for an endpoint security. For example the Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This could lead to credentials or keys being



exposed. An endpoint solution needs to either patch the vulnerable application or monitor it for any signs of exploitation or data leakage and prevent the data from being exfiltrated.

#### **10.3.3. SQL injection attacks**

A SQL injection attack is an example of an attack that exploits the backend logic of an application. Typically this is a web application with access to a database. By encoding specific command characters into the query string, additional SQL commands can be triggered. A successful attack can lead to the content of the whole database being exposed to the attacker. There are other similar attacks that can be grouped together for the purpose of this task, such as command injection or cross site scripting (XSS). Although they are different attacks, they all at their core fail at input filtering and validation, leading to unwanted actions being performed.

Applications that are vulnerable to SQL injections are very common and are not restricted to web applications. An endpoint solution needs to monitor all data entered into possible vulnerable applications. This should include data received from the network. A generic pattern matching for standard SQL injection attack strings can be applied to potentially block some of the attacks. In order to block all types of SQL injection attacks the endpoint solution should have some knowledge about the logic of the monitored application, which helps to determine how normal requests differ from attacks. Applications can be analysed at source code level for potential weaknesses, but dynamically patching is very difficult. See [[SQL](#)]

#### **10.3.4. Low and slow data exfiltration**

An endpoint security solution can detect low and slow data exfiltration, for example when interesting data sources are tracked and access to them is monitored. If the data source is not on the endpoint itself, e.g. a database in the network, then the received data needs to be tagged and its further use needs to be tracked. To make detection difficult, an attacker could decide to use an exfiltration process that sends only 10 bytes every Sunday to a legitimate cloud service. If that is not in the normal behavior pattern, then this anomaly could be detected by the endpoint. If the process that sends the data or the destination IP address have a bad reputation, then they could be stopped. Though it is very difficult to reliably block such an attack and most solutions have a specific threshold that needs to be exceeded before it is detected as an anomaly.



## **10.4. Suboptimality and gray areas**

### **10.4.1. Stolen credentials**

Stolen credentials and misuse of system tools such as RDP, Telnet or SSH are a valid scenario during attacks. An attacker can use stolen credentials to remotely log into a system and access data or execute commands in this context like the legitimate user might do. An endpoint security solution can restrict access from specific IP addresses, but this is difficult in a dynamic environment and when an attacker might have already compromised a trusted device and misuse it as a stepping stone for lateral movement. The endpoint could perform additional checks of the source device, such as verifying installed applications and certain conditions. Again this will not work in all scenarios, e.g. a hijacked valid device during lateral movement.

This means that the system will not be able to simply block the connection if the authentication with the stolen credentials succeeds. A multi factor authentication (MFA) could limit the use of stolen credentials, but depending on the system used and the determination of the attacker they might be able to bypass this hurdle as well e.g. cloning a SIM card to read text message codes.

As a next step, a solution on the endpoint can monitor the behavior of the logged in user and determine if it represents expected normal behavior. Unfortunately, there is the chance for false positives that might block legitimate actions, hence the rules are usually not applied too tightly. The system can monitor for suspicious behavior, similar to malware detection, where every action is carefully analyzed and all activity is tracked. For example if the SSH user is adding all files to archives with passwords and then deletes the original files in the file explorer, then this could result in a ransomware case scenario. If only a few files are processed per hour, then this activity will be very difficult for the endpoint to distinguish from normal activity, in order to flag it as malicious.

The problem of attackers blending in with normal activity is one of the biggest challenges with so called living off the land attack methods. The attacker chooses to keep their profile low by not installing any additional binary files on the system, but instead misuses legitimate system tools to carry out their malicious intent. This means that there is no malware file that could be identified and the detection relies solely on other methods such as behaviour based monitoring [[LOTLSYMC](#)].

If information is shared across multiple endpoints, then each one could learn from the others and see how many connections came in from





that source, what files were involved and what behavior the clients exhibited. This crowd wisdom approach would allow blocking rules to be applied after the first incident across multiple endpoints.

#### **10.4.2. Zero Day Vulnerability**

Description	An attacker exploits a zero day vulnerability or any recent vulnerability.
UE	In theory this scenario could be handled by the endpoint security: a) Once the intrinsic security system has been patched, exploitation of the vulnerability can be prevented. b) The add-on security with enhanced capabilities or updated methods can detect and mitigate the vulnerability. It does not necessarily require the official patch.
Challenge	In practice many systems remain vulnerable to a vulnerability months or even years after a security fix has been released. Moreover there is a big gap between when a vulnerability is disclosed and when a security fix is available. Also there is a big gap between when a security fix is available and when the security fix is actually applied. A recent study over three years, examined the patching time of 12 client-side and 112 server-side applications in enterprise hosts and servers. It took over 6 months on average to patch 90% of the population across all vulnerabilities. <a href="#">[NDSSPATCH]</a> . We note too: "The patching of servers is overall much worse than the patching of client applications. On average a server application remains vulnerable for 7.5 months."
References	<a href="#">[ZERODAY1]</a> <a href="#">[ZERODAY2]</a>

#### **10.4.3. Port scan over the network**

An infected machine, let's say a Mirai bot on a router, is scanning a class B network for IP addresses with TCP port 80 open. The malware can slow it down to 1 IP address per 5 seconds (or any other threshold) and it can go in randomized order (like for example the nmap tool does) in order to make it difficult to find a sequential pattern. To increase detection difficulties, legitimate requests to existing web servers can be added in at random intervals.



An endpoint solution might be able to detect this behaviour, depending on the threshold, but it will be difficult. At some point the pattern will be similar to browsing the web, so either the endpoint blocks the bot scanning and also the user from surfing, or it allows both.

To make it even harder, the attacker can use a botnet that communicates over peer-to-peer (P2P) or a central command and control server (C&C) and then distribute the scan load over multiple hosts. This means each endpoint only scans a subset, let's say 100 IP addresses, but all 1,000 bots scan a total of 100,000 IP addresses.

This attack is difficult to detect by a reasonable threshold on each endpoint individually. If the endpoints talk to each other and exchange information, then a collective decision can be made on the bigger picture of the bot traffic.

Another option for the endpoint solution is to block the bot malware from operating on the computer, for example by detecting the installation, analyzing the behavior of the process or by preventing the binary from accessing the network. This includes blocking any form of communication for the process to its C&C server, regardless of if it is using a P2P network or misusing legitimate system tools or browsers to communicate with the Internet. Blocking indirect communication over system tools as part of living off the land tactics, can be very challenging.

See [[BOT](#)]

#### **10.4.4. DDoS attacks**

For this example let us consider a botnet of 100,000 compromised computers and each one sends a burst of traffic to a remote target, for one second each, alternating in groups. This will generate some waves of pulse attack traffic. Similar comments can be made about overall pulsed DDoS attacks [[PDDoS](#)].

A solution on the endpoint can attempt to detect the outgoing traffic. If the DoS attack is volume based and the time span of each pulse is large enough or the repeating frequency for each bot is high, then detection with thresholds on the endpoint is feasible. It is different, if it is an application layer DoS attack, where the logic of the receiving application is targeted, for example with too many search queries in HTTP GET requests. This would flood the backend server with intensive search requests, which can result in the web site no longer being responsive. Such attacks can succeed with a low amount of requests being sent, especially if its distributed over a botnet. This makes it very difficult for a single



endpoint to detect such an ongoing attack, without knowledge from other endpoints or the network.

Another option for the endpoint solution is to block the bot malware from operating on the computer, for example by detection the installation, analyzing the behavior of the process or by preventing the binary from accessing the network. This includes blocking any form of communication for the process to its C&C server, regardless of if it is using a P2P network or misusing legitimate system tools or browsers to communicate with the Internet. Blocking indirect communication over system tools as part of living off the land tactics, can be very challenging.

## **11. Learnings from production data**

From the above limited considerations we can now check what we see from real production data using

- o the method described in [[MONEYBALL](#)]
- o the anonymised production data of Symantec MSS production for the past 3 months

The core idea is to consider, based on all the imperfections we started to list above including the 'grey areas', that cybersecurity analysts are often presented with suspicious machine activity that does not conclusively indicate a compromise, resulting in undetected incidents or costly investigations into the most appropriate remedial actions.

As Managed Security Services Providers (MSSP's) are confronted with these data quality issues, but also possess a wealth of cross-product security data that enables innovative solutions, we decided to use the Symantec MSS service for the past 3 months. The Symantec MSS service monitors over 100 security products from a wide variety of security vendors for hundreds of enterprise class customers from all verticals.

We selected the subset of customers using the service that deploy both network and endpoint security products to determine which types of security incidents were most likely to be detected by endpoint products vs. network products. In doing so, we were particularly interested in identifying which categories of incidents are detected by endpoint products and not network products, and vice versa. Thus, we examined prevalent categories of incidents for which the only actionable security alerts were predominantly produced by one type of security product and not the other. To do so, we extracted all security incidents detected by Symantec MSS on behalf of hundreds of



customers that deploy both network and endpoint security products, over a three-month period from December 2018 through the end of February 2019. We acknowledge that some attacks might have been blocked by the first product and therefore have never been seen by the next security solution, which influences the final numbers.

With this in mind, we could identify incidents based on:

Severity	4 - Emergency, 3 - Critical, 2 - Warning, 1 -	
	Informational	
Incident	Malicious Code, Deception Activity, Improper Usage,	
Category	Investigation, etc.	
Incident	Trojan Horse Infection, Suspicious DGA Activity,	
Type	Suspicious Traffic, Suspicious URL Activity,	
	Backdoor infection, etc.	
# network	Amount of network only security incidents	
incidents		
# all	What is the total amount of incidents on all	
incidents	security solutions	
Percentage	Percentage of network security only incidents	

We ended up with

- o Hundreds of thousands of security incidents
- o which we could categorize in 275 incident types by category and severity (triplets Severity-Category-Type)
- o out of which we searched how many incidents of each type were detected by a network security product and missed by deployed endpoint security products at least 75% of the time or vice versa

### **11.1. Endpoint only incidents**

The categories of incidents that are detected primarily by endpoint security products are fairly intuitive. They consist primarily of detections of file-based threats and detection of malicious behaviors through monitoring of system and network behavior at the process level. The most prevalent of these behavioral detections include detections of suspicious URLs based on heuristics and blacklists of IP addresses or domain names. Since most of these alerts are not





corroborated by network products, it seems probable that the blacklists associated with network products tend to be more focused on attacks while host-based intrusion prevention system alerts focus more on malware command and control traffic. Most other behavioral detections at the endpoint provide alerts based on system behavior that is deemed dangerous and symptomatic of malicious intent by a malicious or infected process. The highest severity incidents detected on endpoints are instances of post-compromise outbound network behavior that are symptomatic of command and control communications traffic, but these did not show up as being primarily detected by endpoint products as they are frequently corroborated by network-based alerts.

### **11.2. Security incidents detected primarily by network security products**

Perhaps less intuitive are the results of examining categories of security incidents that are detected primarily by network security products and only rarely corroborated by endpoint security products. Below we provide details regarding incident categories for which a network security product produced a detection and for which there were no actionable endpoint alerts for at least 75% of the incidents in the category.

In our study we found 32 incident type, category, and severity triplets of this type. The following categories critical incident types were reported by MSS customers, and we discuss each in turn in decreasing order of prevalence:

#### **11.2.1. Unauthorized external vulnerability scans**

Perhaps unsurprisingly, unauthorized external attempts to scan corporate resources for vulnerabilities and other purposes are detected in large volumes by a broad variety of network-focused security products. 79% of incidents of this type were detected by network security products with critical-severity alerts, these security incident detections are not accompanied by any actionable endpoint alerts, despite the fact that endpoint security products are deployed by these enterprises. This category of threats encompasses a broad variety of attacks, the most prevalent of which are the following: Horizontal scans, SQL injection attacks, password disclosure vulnerabilities, directory traversal attacks, and blacklist hits. Of these categories of detections, horizontal scans stand out as the category of detection that endpoint-security products are least likely to detect on their own.



#### **11.2.2. Unauthorized internal vulnerability scans**

Unauthorized internal vulnerability scans, though less frequent, are more alarming, as they are likely to represent possible post-compromise activity. We note that the Managed Security Service works with its customers to maintain lists of devices that are authorized to perform internal vulnerability scans, and their activity is reported separately at a lower levels of incident severity. 89% of detected unauthorized internal vulnerability scans are detected by network products without any corroborating actionable alerts from endpoint security products. As compared to unauthorized external scan incidents, internal hosts that perform vulnerability scans are far more active and the fraction of alerts that detect horizontal scans is higher, representing half of the total alerts generated. Alerts focused on Network-Behavior Anomaly Detection also appear for internal hosts.

#### **11.2.3. Malware downloads resulting in exposed endpoints**

This category of threats is generally detected by network security appliances. Despite these enterprises being purchasers of endpoint security products, 76% of the incidents detected by the network security products do not show a corresponding alert by an endpoint security product. A broad variety of network appliances contributed to the detection of a diverse collection of malware samples.

#### **11.2.4. Exploit kit infections**

This category of infections represents instances in which the customer's machines are exposed to exploit kits. These threats were detected by network appliances that extract suspicious URLs from network traffic taps and use a combination of sandbox technology and blacklists to identify websites that deploy a variety of exploit kits that were not being caught by endpoint security products. In this three month time period, the most prevalent categories of exploit kits detected involved redirections to the Magnitude exploit kit and exploit kits associated with phishing scams and attempts to expose users to fake Anti-Virus warnings and tools. A breakdown of the results is included below:



Severity	3 - Critical
Incident Category	Malicious Code
Incident Type	Exploit Kit Infection
# network incidents	26
# all incidents	26
Percentage	100%

The network security product that detected these incidents produced the following alerts:

- o Advanced Malware Payloads
- o Exploit.Kit.FakeAV
- o Exploit.Kit.Magnitude
- o Exploit.Kit.MagnitudeRedirect
- o Exploit.Kit.PhishScams
- o HTMLMagnitudeLandingPage

#### **11.2.5. Attacks against servers**

In addition to detecting the aforementioned critical security incident categories, network security devices frequently detect a broad variety of attacks against servers that usually lack corroboration at the endpoint. Most server attacks are not matched by endpoint protection alerts: 62% are unmatched for critical incidents, and 88% are unmatched as lower severity incidents. This category of incidents is the most prevalent category of incidents detected primarily by network products, but they are usually rated lower in severity than the aforementioned classes of alerts as they are very commonplace. Even when these alerts are corroborated by endpoint protection alerts, the endpoint alerts are often low in severity, as in the case of file-based threats that appear to have been blocked or successfully cleaned up by an Anti-Virus solution. The challenge in taking action against server attacks is that it can be difficult to assess which of these attacks were successful in causing actual damage, and for this reason, for the fraction of server attacks that demonstrate corroborating endpoint security



alerts, even if of low severity, should be examined. It is interesting to note the cooperative role played by both network and endpoint security devices in these instances.

## **12. Regulatory Considerations**

This section will briefly look at the regulatory landscape and develop a specific view on the impact on endpoints with the goal to see what we might be able to learn.

Legal requirements, compliance, regulatory frameworks and mandatory reporting are no longer separate from any security evaluation, process or requirement within an organisation, enterprise system or intranet. It is essential to look at the technical and regulatory approaches together. This section will look at two examples of legal requirements and guidance:

(1) IoT security (2) Network infrastructure

This section is by no means complete, but it does a discussion on this aspect of endpoint and ecosystem regulation.

### **12.1. IoT Security**

IoT security regulation is emerging in the form of voluntary frameworks and self-assessments that relate to endpoint security issues.. These frameworks focus first on the end point, or mobile device, in the IoT environment and then on the holistic ecosystem itself.

In 2017 the National Institute of Standards and Technology released its draft IoT Cybersecurity Framework based on consultations and interviews with all stakeholders over several years previously [[NISTIOTP](#)]. Some of the themes which emerged was the need for IoT governance, assessment frameworks, review of all aspects of the IoT ecosystem and a process for coordinated vulnerability disclosure inside an organisation. As evidenced by the 2018 Endpoint Protection and Response Survey by SANS, only 47% of organisations know that their endpoints have been breached and a further 20% are unsure [[EPRSANS](#)]. So a systemic approach from NIST was welcomed and the NIST framework became the gold standard for national IoT security frameworks.

Other IoT security frameworks include the Singapore IoT Cyber Security Guide from January 2019 and the UK's Secure by Design or The Government's Code of Practice for Consumer Internet of Things (IoT) Security for manufacturers, with guidance for consumers on smart devices at home [[IMDAIOTG](#)], [[SBDGOVUK](#)]. Once again both look at





securing the IoT device or endpoint, but also security for the entire value chain of the IoT system. The Singapore framework makes the point about the entire system clear, "Similar to any system, an IoT system is as secure as its weakest link. It is thus important to ensure that proper security considerations and measures are put in place for both the implementation and operational stages of the deployment of any IoT system." [[IMDAIOTG](#)] Finally, the IoT Security Foundation, the GSMA and the Internet Society have all released their own frameworks for IoT security. All have similar characteristics which focus on the entire value chain and ecosystem, but also on vulnerability disclosure and checklist assessments. What makes each of these approaches slightly different is the differing perspectives of the organization advocating it. The GSMA is the mobile trade association and so it focuses on mobile devices while the Internet Society focuses on the Internet ecosystem and a multistakeholder approach. Systematically underpinning all the frameworks is the holistic approach with voluntary best practices and implementation based on the needs of the user or organisation adopting the framework [[IOTSFCE](#)], [[GSMAIOT](#)], [[ISTRUST](#)].

### **12.2. Network infrastructure**

In Europe, the Network and Information Security Directive, which was passed in July 2016, require implementation by each European member state with a threefold aim. First, to put into place a national strategy for network and infrastructure security including best practices, guidelines, training and stakeholder consultations. Second, to coordinate national CSIRTs with CERT-EU and third to provide incident control and response systems for critical infrastructure and digital services [[EURLEX](#)]. This Directive demonstrates the importance given across the EU to network resilience and incident reporting. While securing the endpoint is acknowledged, the focus is on ensuring the security of European interoperable networks. In short, the importance of the security of the network including incident response shows that it isn't only the endpoints that should be the focus of the regulation and legal frameworks.

### **12.3. Auditing and Assessment**

This section will talk about other risk assessment and auditing regulatory requirements beyond the NIS directive.

One example of risk assessment as a regulatory requirement is the New York State law 23 NYCRR 500 of the Regulations of the Superintendent of Financial Services (Cybersecurity Requirements for Financial Services Companies). Among the requirements, audit, risk assessment and risk reporting are included like,



(2) include audit trails designed to detect and respond to Cybersecurity Events that have a reasonable likelihood of materially harming any material part of the normal operations of the Covered Entity. [NYCYBER]

#### **12.4. Privacy Considerations**

We may consider a specific focus on privacy in the future.

#### **13. Human Rights Considerations**

This section may develop a specific view of requirements, limits and constraints coming from Human Rights perspective on endpoint security.

#### **14. Security Considerations**

This document is about Security Considerations

#### **15. IANA Considerations**

This document has no actions for IANA

#### **16. Informative References**

[ADAPTURE]

Cullen, T., "Limits of endpoint only", July 2017, <<https://www.adapture.com/blog/evaluating-leading-endpoint-security-vendors/>>.

[AMT1]

Khandelwal, S., "Explained - How Intel AMT Vulnerability Allows to Hack Computers Remotely", May 2017, <<https://thehackernews.com/2017/05/intel-amt-vulnerability.html>>.

[AMT2]

Symantec, ., "Web Attack Intel AMT Privilege Escalation CVE-2017-5689", 2017, <[https://www.symantec.com/security\\_response/attacksignatures/detail.jsp?asid=29888](https://www.symantec.com/security_response/attacksignatures/detail.jsp?asid=29888)>.

[ATTACK]

"MITRE ATT&CK", n.d., <<https://attack.mitre.org>>.

[BOT]

Marinho, R., "Exploring a P2P transient botnet - From Discovery to Enumeration", July 2017, <<https://morphuslabs.com/exploring-a-p2p-transient-botnet-from-discovery-to-enumeration-e72870354950>>.



- [CANDID1] Wueest, C., "How my TV got infected with ransomware and what you can learn about it", November 2015, <<https://www.symantec.com/connect/blogs/how-my-tv-got-infected-ransomware-and-what-you-can-learn-it>>.
- [CANDID2] Dickson, B., "Millions of smart TVs are vulnerable to hackers", February 2014, <<https://www.dailydot.com/debug/protect-smart-tv/>>.
- [CAPEC] "MITRE CAPEC", n.d., <<https://capec.mitre.org/data/definitions/3000.html>>.
- [ENISA] ENISA, ., "Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures", November 2017, <<https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot>>.
- [EPPEDR] Redscan, ., "EPP and EDR - What's the difference?", June 2018, <<https://www.redscan.com/news/epp-vs-edr-whats-the-difference/>>.
- [EPPGUIDE] "IT Pro's Guide to Endpoint Protection", n.d., <<https://www.barkly.com/it-pros-guide-to-endpoint-protection>>.
- [EPPSECURITY] Hunt, J., "Advantages and Disadvantages of Three Top Endpoint Security Vendors", n.d., <<https://www.adapture.com/blog/evaluating-leading-endpoint-security-vendors/>>.
- [EPRSANS] Neely, L., "Endpoint Protection and Response A SANS Survey", June 2018, <<https://www.sans.org/reading-room/whitepapers/clients/paper/38460>>.
- [EPTAXONOMY] MacFadden, M., "Endpoint Taxonomy for CLESS", July 2019, <<https://www.ietf.org/id/draft-mcfadden-smart-endpoint-taxonomy-for-cless-00.txt>>.
- [ERICSSON] Ericsson, ., "Internet of Things forecast", n.d., <<https://www.ericsson.com/en/mobility-report/internet-of-things-forecast>>.



- [EURLEX] EUP, ., "Directive (EU) 2016/1148", July 2016,  
<[https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=urisrv:OJ.L\\_.2016.194.01.0001.01.ENG&toc=OJ:L:2016:194:TOC](https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=urisrv:OJ.L_.2016.194.01.0001.01.ENG&toc=OJ:L:2016:194:TOC)>.
- [FLAMER] Symantec, ., "W32.Flamer Microsoft Windows Update Man-in-the-Middle", June 2012,  
<<https://www.symantec.com/connect/blogs/w32flamer-microsoft-windows-update-man-middle>>.
- [GARTNERIOT] Van der Meulen, R., "Gartner Says 8.4 Billion Connected Things Will be in Use in 2017, Up 31 percent from 2016", February 2017, <<https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016>>.
- [GARTNERREPORT] Crotty, J., "New Gartner Report Redefines Endpoints Protection for 2018", January 2018,  
<<https://www.crowdstrike.com/blog/new-gartner-report-redefines-endpoint-protection-for-2018/>>.
- [GSMAIOT] GSMA, ., "GSMA IoT Security Guidelines and Assessment", n.d., <<https://www.gsma.com/iot/iot-security/iot-security-guidelines/>>.
- [HSTODAY] Hstoday, ., "Layered Approach Critical to Effective Endpoint Protection", October 2016,  
<<https://www.hstoday.us/channels/federal-state-local/layered-approach-critical-to-effective-endpoint-protection/>>.
- [I-D.[draft-mcfadden-smart-endpoint-taxonomy-for-cless-00](#)] McFadden, M., "Endpoint Taxonomy for CLESS", [draft-mcfadden-smart-endpoint-taxonomy-for-cless-00](#) (work in progress), July 2019.
- [IMDAIOTG] IMDA, ., "IMDA IoT Cyber Security Guide", January 2019,  
<<https://www.imda.gov.sg/-/media/imda/files/regulation-licensing-and-consultations/consultations/open-for-public-comments/consultation-for-iot-cyber-security-guide/imda-iot-cyber-security-guide.pdf>>.





## [IOTPATCHING]

Rogers, D., "Handling vulnerabilities as an IoT vendor", December 2018, <<https://www.iotsecurityfoundation.org/less-than-10-of-consumer-iot-companies-follow-vulnerability-disclosure-guidelines/>>.

[IOTSFCF] IoTSF, ., "IoT Security Compliance Framework", December 2018, <<https://www.iotsecurityfoundation.org/best-practice-guidelines/>>.

[ISTRUST] ISOC, ., "Internet of Things (IoT) Trust Framework v2.5", May 2018, <<https://www.internetsociety.org/resources/doc/2018/iot-trust-framework-v2-5/>>.

[LOJAX] ESET, ., "LoJax First UEFI rootkit found in the wild, courtesy of the Sednit group", September 2018, <<https://www.welivesecurity.com/2018/09/27/lojax-first-uefi-rootkit-found-wild-courtesy-sednit-group/>>.

## [LOTLSYMC]

Wueest, C., "Living off the land and fileless attack techniques", July 2017, <<https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-living-off-the-land-and-fileless-attack-techniques-en.pdf>>.

[MIRAI1] Symantec, ., "Mirai, what you need to know about the botnet behind recent major DDoS attacks", October 2016, <<https://www.symantec.com/connect/blogs/mirai-what-you-need-know-about-botnet-behind-recent-major-ddos-attacks>>.

[MIRAI2] Krebsonsecurity, ., "19 Mirai Botnet Authors Avoid Jail Time", September 2018, <<https://krebsonsecurity.com/tag/mirai-botnet/>>.

## [MONEYBALL]

Roundy, K., "Predicting Cyber Threats with Virtual Security Products. ACSAC", 2017, <<https://www.cc.gatech.edu/~dchau/papers/17-acsac-moneyball.pdf>>.

## [NDSSPATCH]

Caballero, J., "Mind Your Own Business A Longitudinal Study of Threats and Vulnerabilities in Enterprises", February 2019, <[https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019\\_03B-1-2\\_Kotzias\\_paper.pdf](https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019_03B-1-2_Kotzias_paper.pdf)>.



## [NETTODAY]

Dix, J., "Layered Security Defenses What layer is most critical network or endpoint", July 2011, <<https://www.networkworld.com/article/2220204/tech-debates/layered-security-defenses--what-layer-is-most-critical--network-or-endpoint-.html>>.

## [NINESIGNS]

Smith, K., "9 signs your endpoint security isn't working", May 2017, <<https://securelement.com/9-signs-your-endpoint-security-isnt-working/>>.

## [NISTIOTP]

NIST, ., "NIST Cybersecurity for IoT Program", November 2016, <<https://www.nist.gov/programs-projects/nist-cybersecurity-iot-program>>.

## [NYCYBER]

NYCRR, ., "See 3 NYCRR 500 of the Regulations of the Superintendent of Financial Services (Cybersecurity Requirements for Financial Services Companies)", n.d., <<https://www.dfs.ny.gov/docs/legal/regulations/adoptions/dfsrf500txt.pdf>>.

## [OWASP]

OWASP, ., "Defense in depth definition", August 2015, <[https://www.owasp.org/index.php/Defense\\_in\\_depth](https://www.owasp.org/index.php/Defense_in_depth)>.

## [PDDoS]

Seals, T., "Pulse-Wave DDoS Attacks Mark a New Tactics in Q2", October 2017, <<https://www.infosecurity-magazine.com/news/pulsewave-ddos-attacks-mark-q2/>>.

## [SBDGOVUK]

UK, GOV., "Secure by Design", February 2019, <<https://www.gov.uk/government/collections/secure-by-design>>.

## [SGX1]

Claburn, T., "Intel SGX safe room easily trashed by white-hat hacking marauders Enclave malware demoed", February 2019, <[https://www.theregister.co.uk/2019/02/12/intel\\_sgx\\_hacked/](https://www.theregister.co.uk/2019/02/12/intel_sgx_hacked/)>.

## [SGX2]

Cimpanu, C., "Researchers hide malware in Intel SGX enclaves", February 2019, <<https://www.zdnet.com/article/researchers-hide-malware-in-intel-sgx-enclaves/>>.

## [SQL]

Cobb, M., "SQL injection detection tools and prevention strategies", November 2009, <<https://www.computerweekly.com/tip/SQL-injection-detection-tools-and-prevention-strategies>>.



## [STATISTA1]

Statista, ., "Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions)", n.d., <<https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>>.

## [STATISTA2]

Statista, ., "Size of Internet of Things market worldwide in 2014 and 2020 by industry (in billion U.S dollars)", n.d., <<https://blogs-images.forbes.com/louiscolumbus/files/2017/12/size-of-IoT-Market-globally-2014-to-2020.jpg>>.

## [TEEP]

Cam-Winget, N., "Trust Execution Environment Protocol", March 2018, <<https://datatracker.ietf.org/wg/teep/about>>.

## [USCERT]

Michael, C., "Principles of defense-in-depth", September 2005, <<https://www.us-cert.gov/bsi/articles/knowledge/principles/defense-in-depth>>.

## [ZERODAY1]

McHugh, J., "Windows of Vulnerability A Case Study Analysis", 2000, <[http://www.cs.colostate.edu/~cs635/Windows\\_of\\_Vulnerability.pdf](http://www.cs.colostate.edu/~cs635/Windows_of_Vulnerability.pdf)>.

## [ZERODAY2]

Plattner, B., "Large-Scale Vulnerability Analysis", September 2006, <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.173.3056&rep=rep1&type=pdf>>.

## **Appendix A. Contributors**

- o Arnaud Taddei  
Symantec  
arnaud\_taddei@symantec.com
- o Bret Jordan  
Symantec  
bret\_jordan@symantec.com
- o Candid Wueest  
Symantec  
candid\_wueest@symantec.com
- o Chris Larsen  
Symantec  
chris\_larsen@symantec.com



- o Andre Engel  
Symantec  
andre\_ngel@symantec.com
- o Kevin Roundy  
Symantec  
kevin\_roundy@symantec.com
- o Yuqiong Sun  
Symantec  
Yuqiong\_Sun@symantec.com
- o David Wells  
Symantec  
David\_Wells@symantec.com
- o Dominique Lazanski  
Last Press Label  
dml@lastpresslabel.com

#### Authors' Addresses

Arnaud Taddei  
Symantec Corporation  
350 Ellis Street  
Mountain View CA 94043  
USA

Email: arnaud\_taddei@symantec.com

Candid Wueest  
Symantec Corporation  
350 Ellis Street  
Mountain View CA 94043  
USA

Email: candid\_wueest@symantec.com

Kevin A. Roundy  
Symantec Corporation  
350 Ellis Street  
Mountain View CA 94043  
USA

Email: kevin\_roundy@symantec.com





Dominique Lazanski  
Last Press Label  
Flat 1, 109A Columbia Road  
London E2 7RL  
UK

Email: [dml@lastpresslabel.com](mailto:dml@lastpresslabel.com)